Instrukcja tworzenia CRUDa JavaFX

PAWEŁ JADANOWSKI

Pobieramy

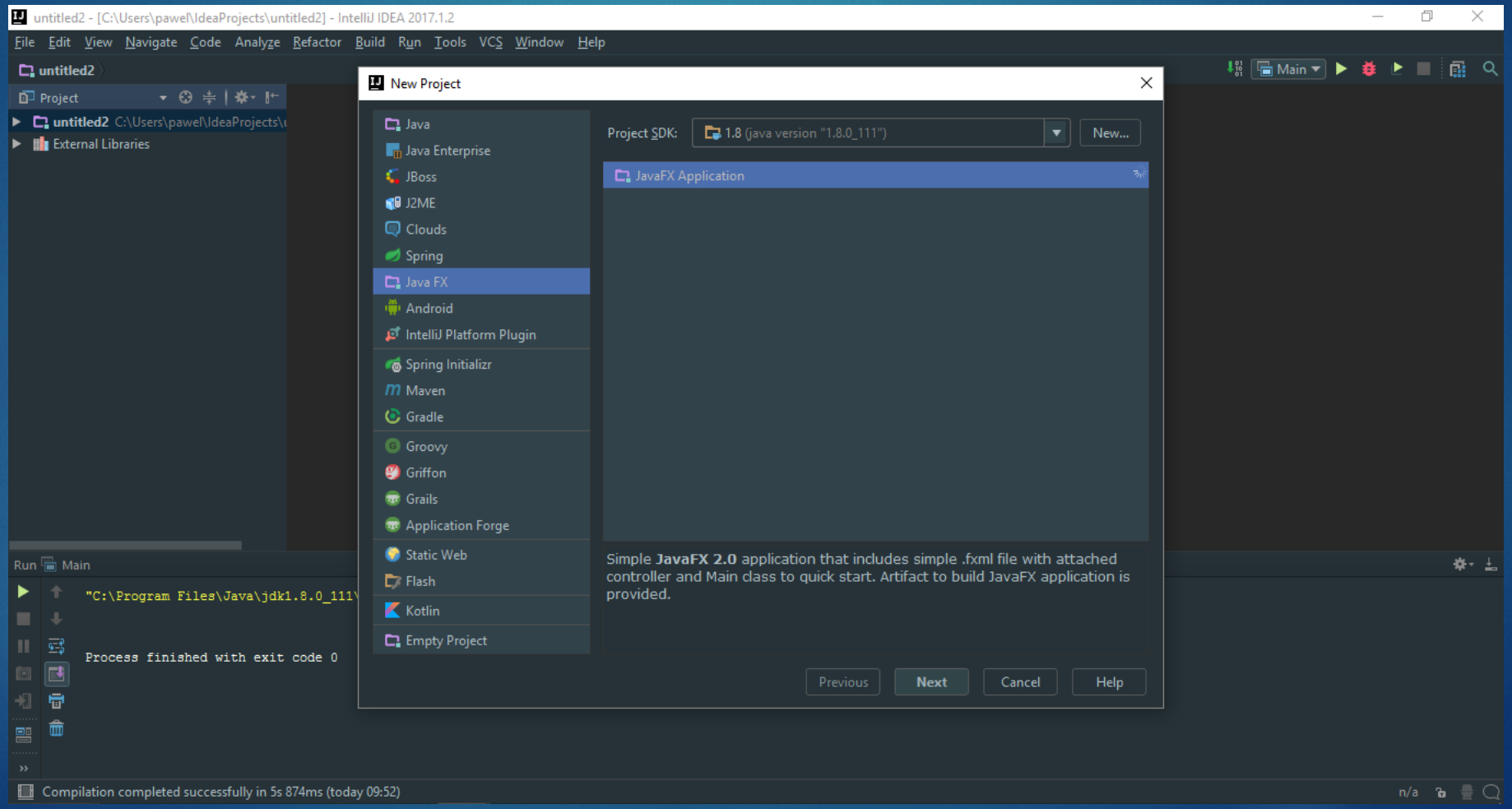
IntelliJ IDEA Community Edition
https://www.jetbrains.com/idea/download/download-thanks.html?code=IIC

Scene Buildera
http://gluonhq.com/products/scene-builder/

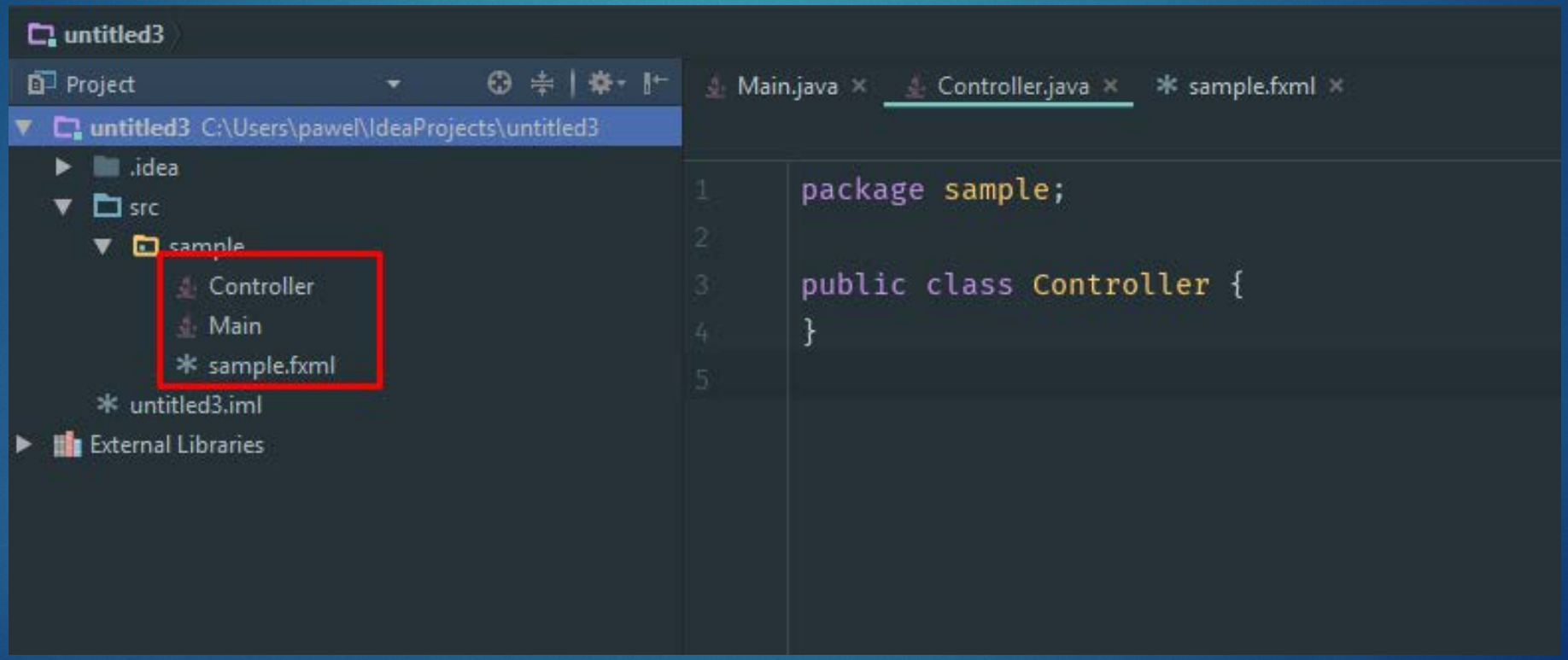
JDBC SQLite Driver
https://bitbucket.org/xerial/sqlite-jdbc/downloads/

Tworzenie nowego projektu JavaFX

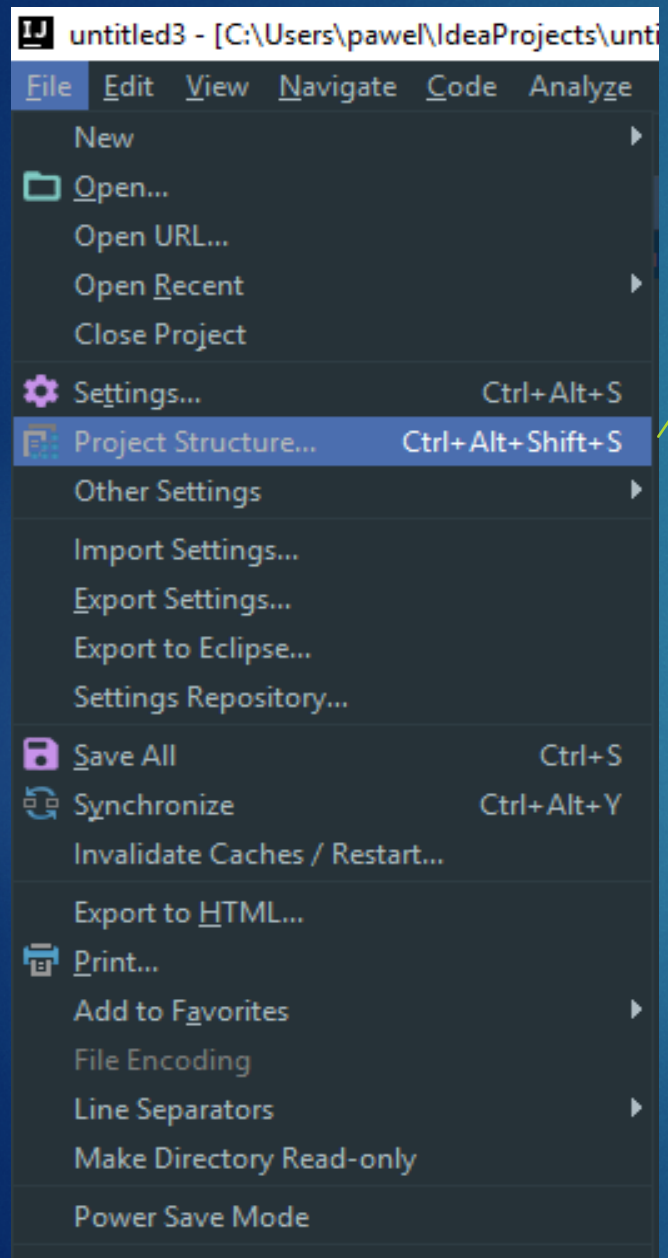
W IntelliJ Idea wybieramy
File → New → Project

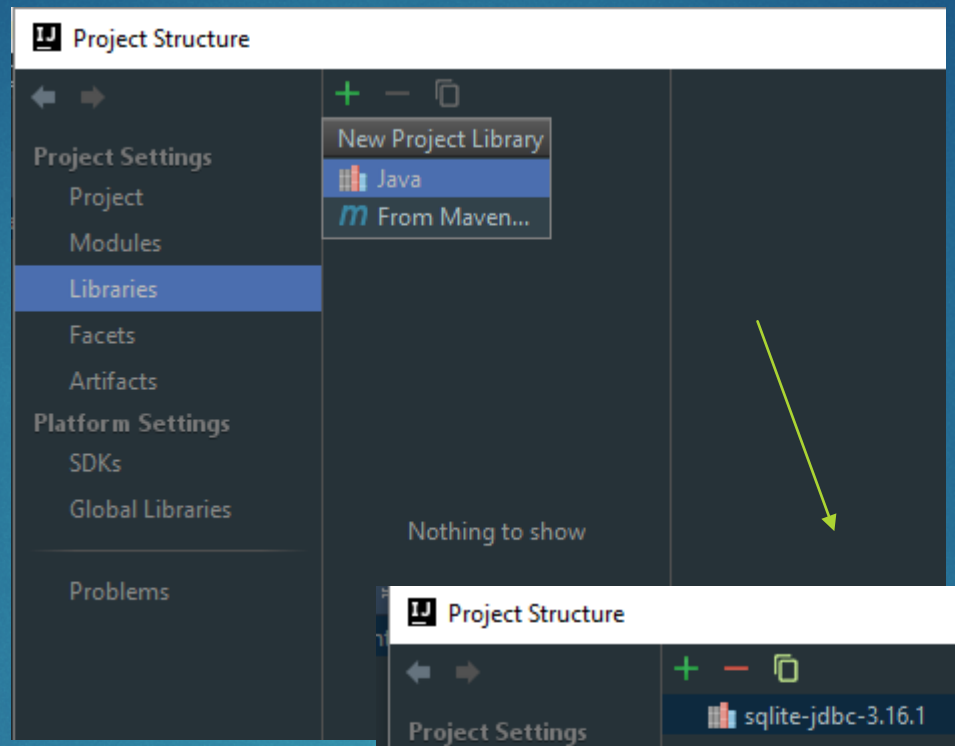


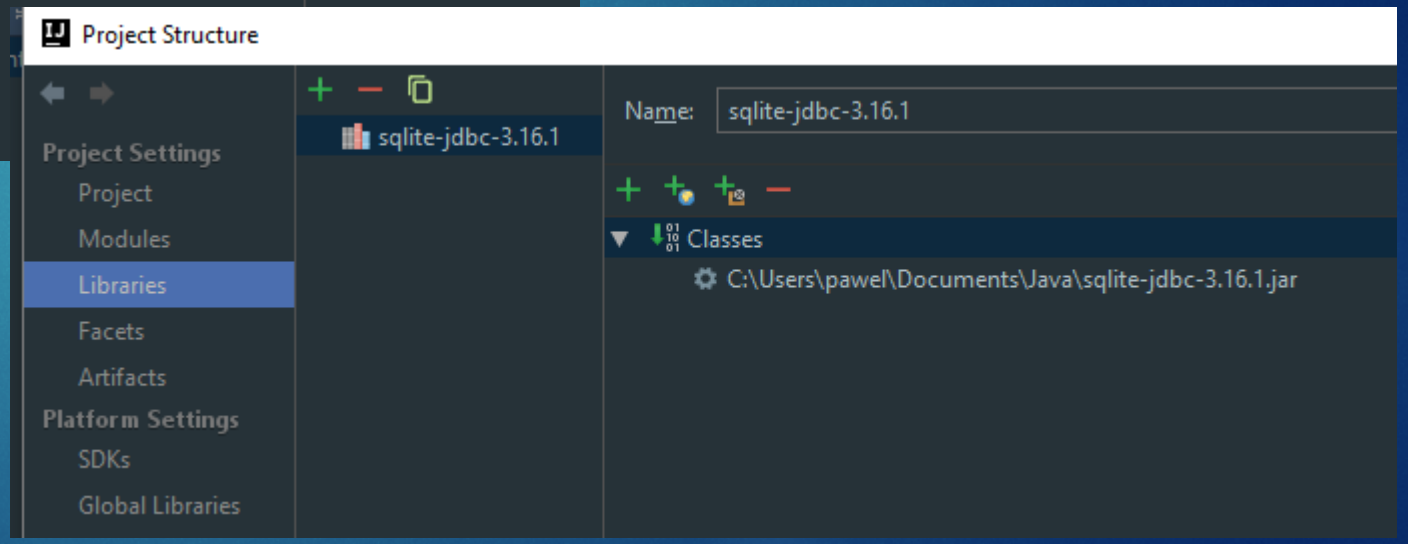
Automatycznie zostają wygenerowane 3 pliki: Controller, Main, sample.fmxl



W celu połączenia do bazy danych potrzebujemy podpiąć sterownik







Tworzymy klasę z połączeniem do bazy

dbConnection.java

```java
package sample;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class dbConnection {

    private static final String SQLITECONN = "jdbc:sqlite:school.sqlite";

    public static Connection getConnection() throws SQLException {

        try {
            Class.forName("org.sqlite.JDBC");
            return DriverManager.getConnection(SQLITECONN);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
        return null;
    }
}
```

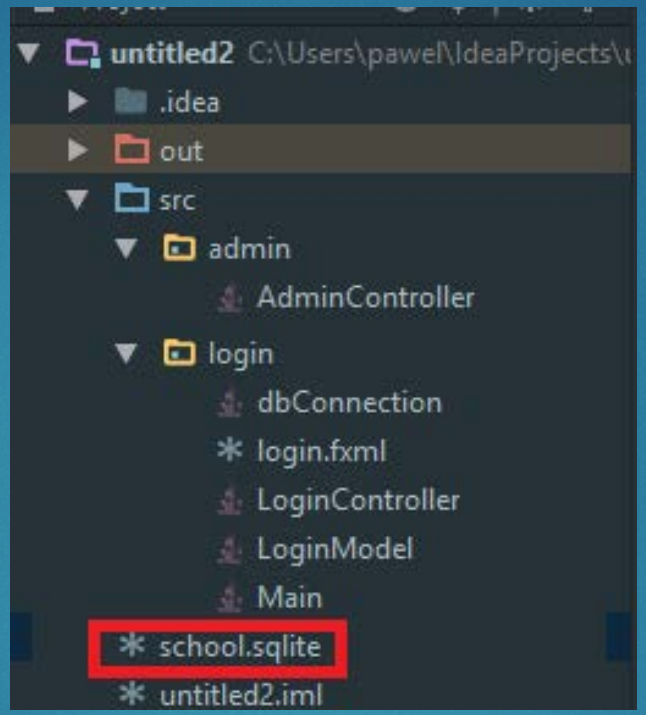
Podpinamy Scene buildera do programu

Po kliknięciu prawym na pliku *.fxml wybieramy
„Open in SceneBuilder”
Przy pierwszej próbie będziemy musieli podpiąć Scene Builder’a

Znajduje się on odpowiednio

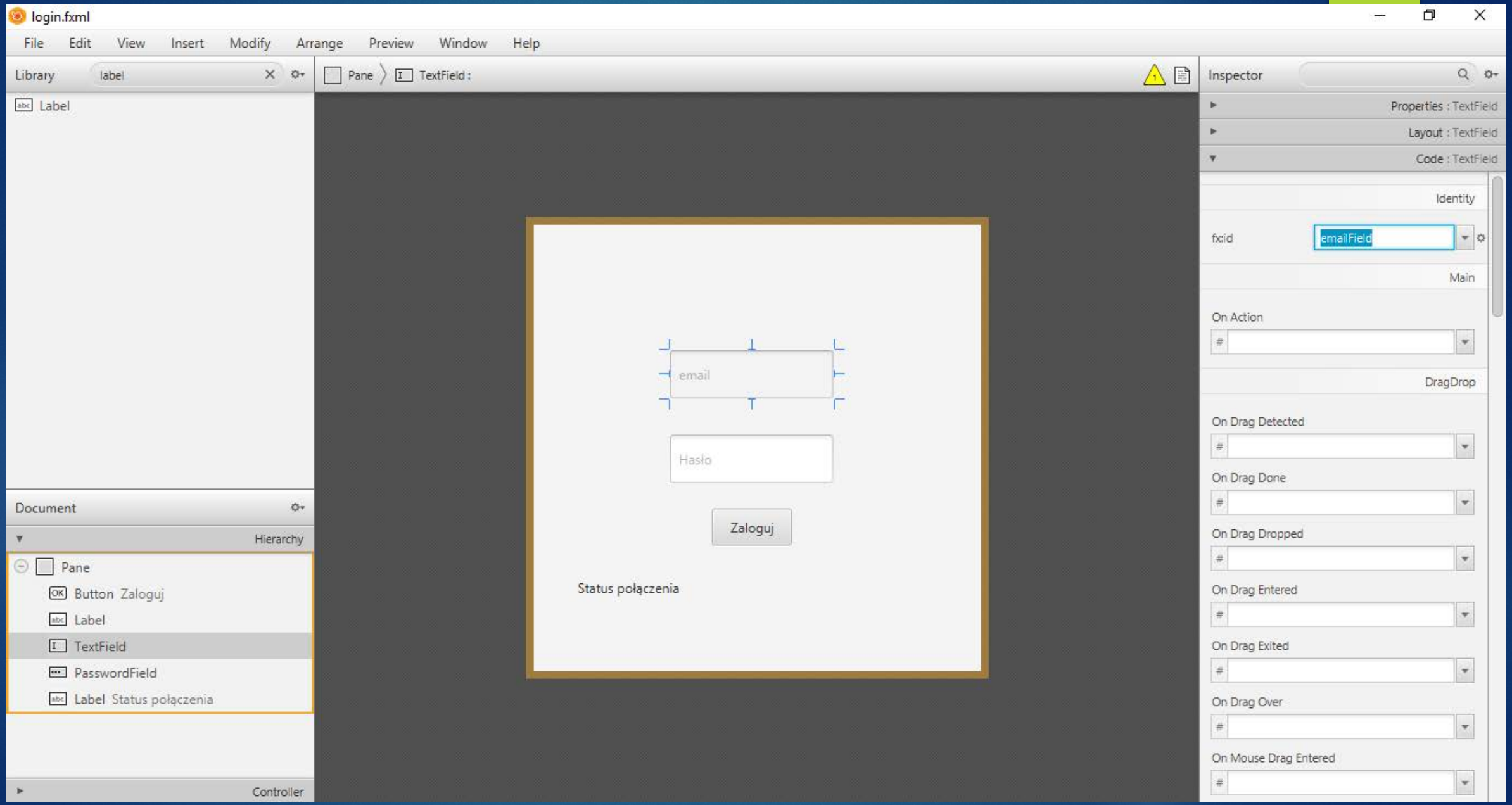
C:\Users\Pawel\AppData\Local\SceneBuilder\SceneBuilder.exe

Ostatecznie zmieniając nazwy na bardziej wymowne niż „sample” nasza struktura projektu wygląda tak

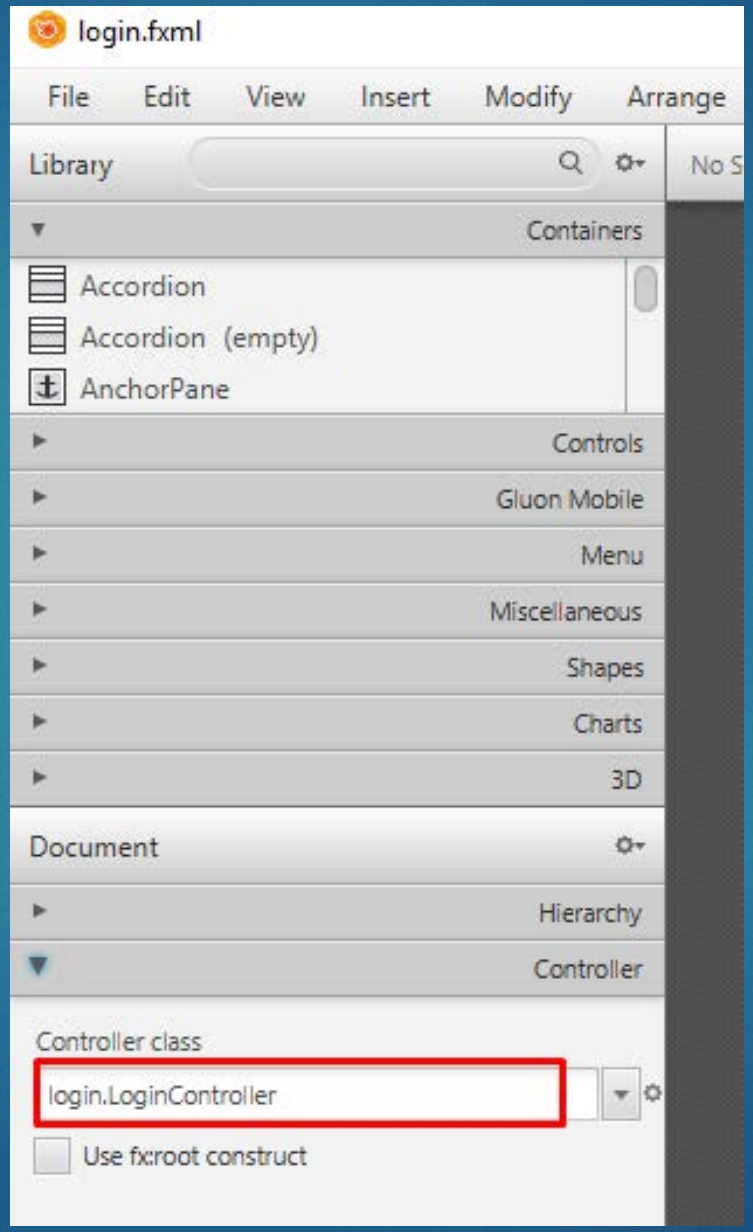


Warto zwrócić uwagę na to że plik z bazą danych został również dorzucony do projektu

Następnie tworzymy layout oraz każdemu elementowi przypisujemy unikalne fx:id



Podpinamy kontroler do pliku fmxl



Następnie tworzymy klasę LoginModel

Tworzymy klasę LoginModel i piszemy poniższy kod

File - C:\Users\pawel\IdeaProjects\untitled2\src\sample\LoginModel.java

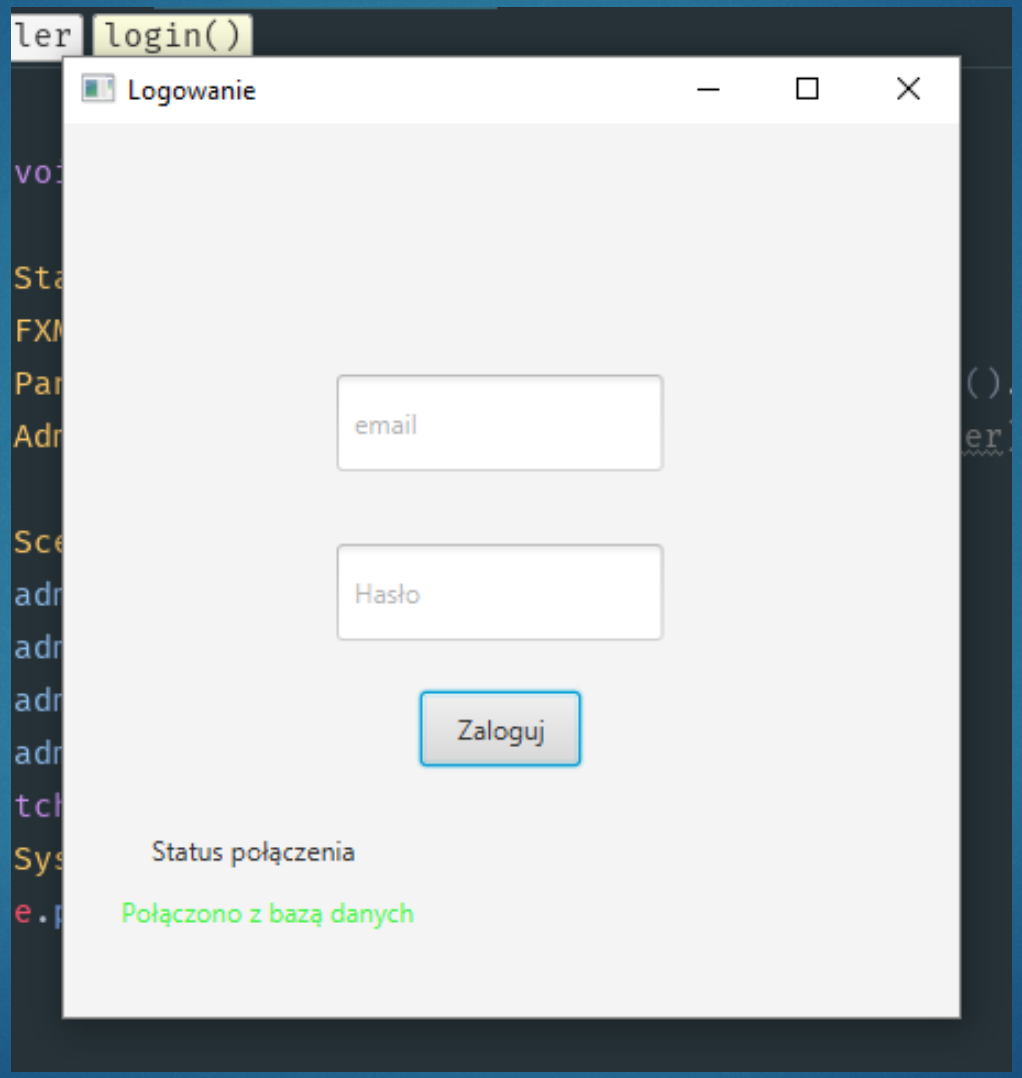
```
1 package sample;
2
3 import java.sql.Connection;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7
8 public class LoginModel {
9
10     Connection connection;
11
12     public LoginModel(){
13         try {
14             this.connection = dbConnection.getConnection();
15         } catch(SQLException e){
16             e.printStackTrace();
17         }
18
19         if (this.connection == null){
20             System.exit(1);
21         }
22     }
23
24     public boolean isDatabaseConnected(){
25         return this.connection != null;
26     }
27
28     public boolean checkLoginData(String user, String pass) throws Exception
29     {
30         PreparedStatement pr = null;
31         ResultSet result = null;
32
33         String sql = "SELECT * FROM students where email=? and password = ? ;";
34
35         try {
36             pr = this.connection.prepareStatement(sql);
37             pr.setString(1, user);
38             pr.setString(2, pass);
39
40             result = pr.executeQuery();
41
42             if (result.next()){
43                 return true;
44             } else  return false;
45         }
46         catch (SQLException e){
47             return false;
48         }
49         finally {
50             pr.close();
51             result.close();
52         }
53     }
54
55
56 }
57
```

oraz tworzymy klasę LoginController

```java
package login;

import admin.AdminController;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.stage.Stage;

import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;

public class LoginController implements Initializable {


    LoginModel loginModel = new LoginModel();
    @FXML
    private Label dbStatus, loginStatusLbl;
    @FXML
    private TextField emailField;
    @FXML
    private PasswordField passwordField;
    @FXML
    private Button loginButton;

    public void initialize(URL url, ResourceBundle rb) {
        if (this.loginModel.isDatabaseConnected()) {
            this.dbStatus.setText("Połączono z bazą danych");
            this.dbStatus.setTextFill(Color.web("#42f445"));

        } else {
            this.dbStatus.setText("Brak połączenia z bazą danych");
            this.dbStatus.setTextFill(Color.web("#f44141"));
        }

    }


    @FXML
    public void login(ActionEvent event) {
        try {
            if (this.loginModel.checkLoginData(
                    this.emailField.getText(),
                    this.emailField.getText()
            )) {
                Stage stage = (Stage) this.loginButton.getScene().getWindow();
                stage.close();

                adminLogin();

            } else {
                this.loginStatusLbl.setText("Błędne dane");
            }

        } catch (Exception e) {
            e.printStackTrace();
        }


    }
```

```
66
67      private void adminLogin() {
68          try{
69              Stage adminStage = new Stage();
70              FXMLLoader adminLoader = new FXMLLoader();
71              Pane adminroot = (Pane)adminLoader.load(getClass().getResource("/
    admin/adminFXML.fxml").openStream());
72              AdminController adminController = (AdminController)adminLoader.
    getController();
73
74              Scene scene = new Scene(adminroot);
75              adminStage.setScene(scene);
76              adminStage.setTitle("Panel Admina");
77              adminStage.setResizable(false);
78              adminStage.show();
79          } catch (IOException e) {
80              System.out.println("AdminLogin() ex");
81              e.printStackTrace();
82          }
83
84      }
85
86  }
87
```

Pozostało już uruchomić program



W pakiecie admin powinniśmy mieć teraz następujące klasy

- AdminController
- adminFXML.fxml
- Oraz w zależności od rodzaju przechowywanych tabel w bazie klasę np. StudentData dla tabeli ze studentami

Zajmijmy się najpierw wypełnieniem klasy StudentData
Tworzymy w niej pola odpowiadające rodzajom kolumn w danej tabeli. Przykładowy plik

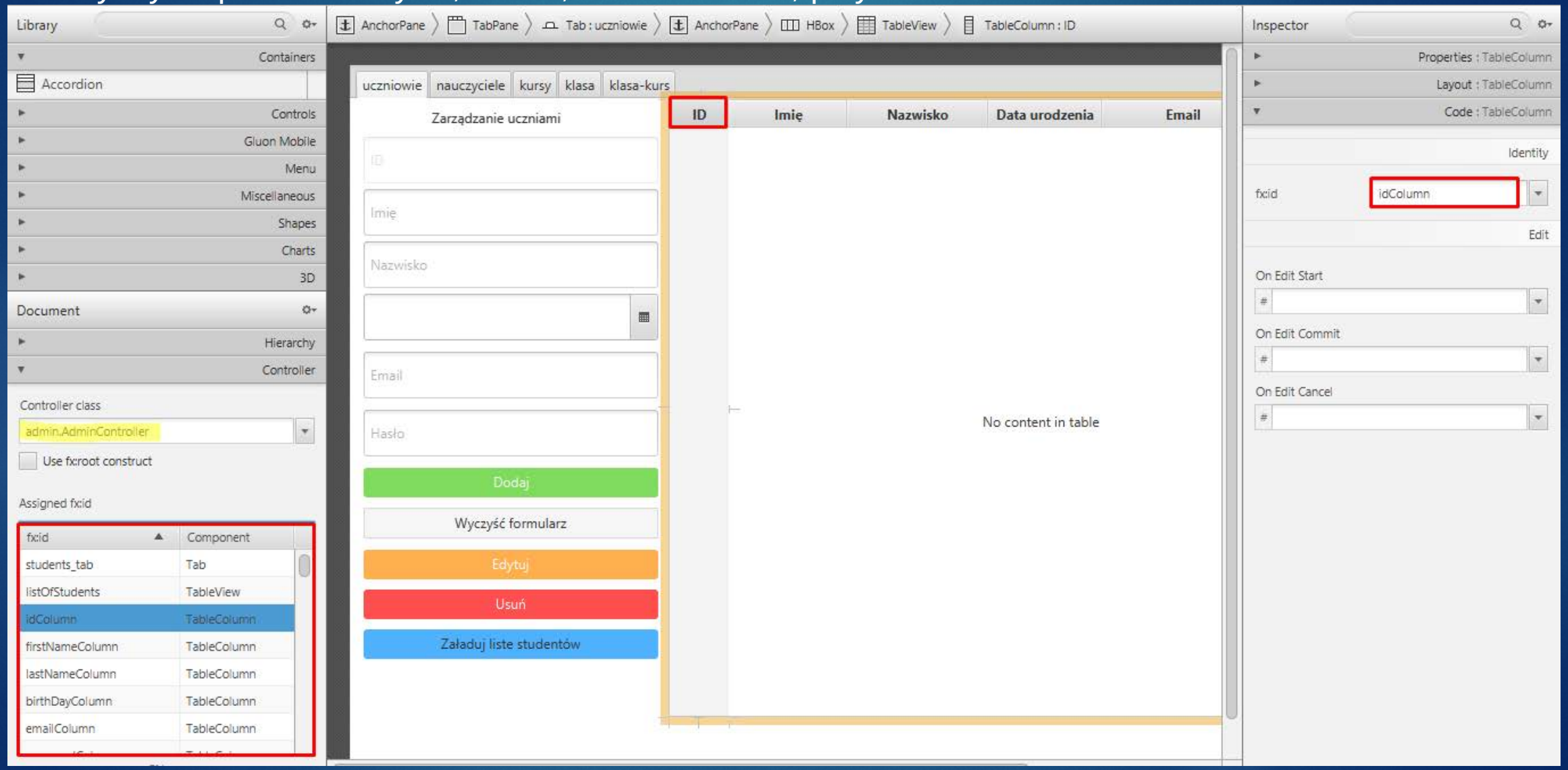
```java
package admin;


import javafx.beans.property.SimpleIntegerProperty;
import javafx.beans.property.SimpleStringProperty;

public class StudentData {

    private final SimpleIntegerProperty ID;
    private final SimpleStringProperty firstName;
    private final SimpleStringProperty lastName;
    private final SimpleStringProperty birthDay;
    private final SimpleStringProperty email;
    private final SimpleStringProperty password;

    public StudentData(int id, String firstname, String lastname, String dob, String email, String pass) {
        this.ID = new SimpleIntegerProperty(id);
        this.firstName = new SimpleStringProperty(firstname);
        this.lastName = new SimpleStringProperty(lastname);
        this.birthDay = new SimpleStringProperty(dob);
        this.email = new SimpleStringProperty(email);
        this.password = new SimpleStringProperty(pass);
    }

    public int getID() {
        return ID.get();
    }

    public SimpleIntegerProperty IDProperty() {
        return ID;
    }

    public void setID(int ID) {
        this.ID.set(ID);
    }

    public String getFirstName() {
        return firstName.get();
    }

    public SimpleStringProperty firstNameProperty() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName.set(firstName);
    }

    public String getLastName() {
        return lastName.get();
    }

    public SimpleStringProperty lastNameProperty() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName.set(lastName);
    }

    public String getBirthDay() {
        return birthDay.get();
    }

```

```
    public SimpleStringProperty birthDayProperty() {
        return birthDay;
    }

    public void setBirthDay(String birthDay) {
        this.birthDay.set(birthDay);
    }

    public String getEmail() {
        return email.get();
    }

    public SimpleStringProperty emailProperty() {
        return email;
    }

    public void setEmail(String email) {
        this.email.set(email);
    }

    public String getPassword() {
        return password.get();
    }

    public SimpleStringProperty passwordProperty() {
        return password;
    }

    public void setPassword(String password) {
        this.password.set(password);
    }
}

```

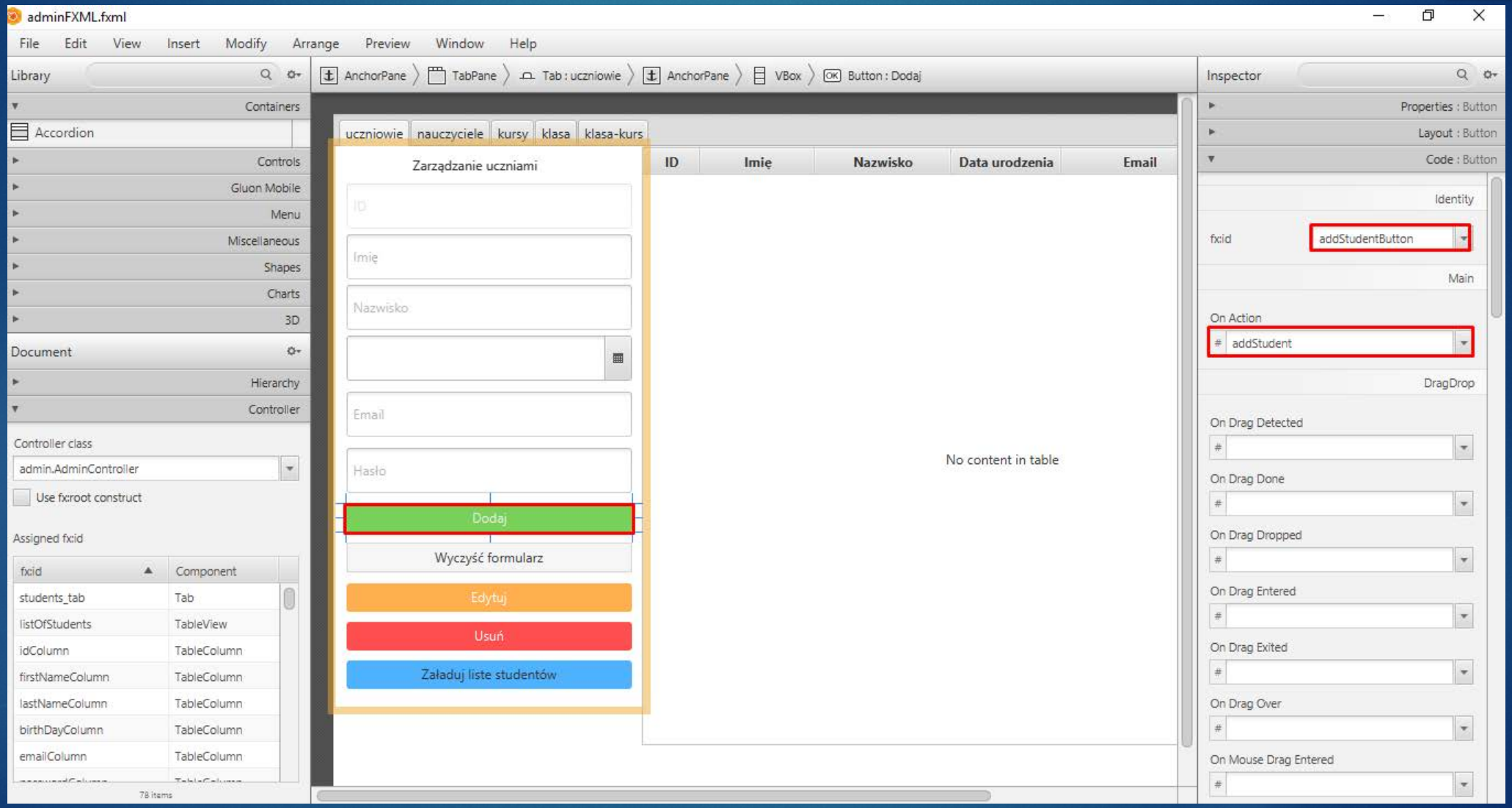
Następnie edytujemy plik adminFXML w Scene Builderze i tworzymy layout

Ważną rzeczą jest to aby do każdego elementu przypisać unikalne fx:id

Dotyczy to pól tekstowych, tabeli, kolumn tabeli, przycisków etc



Pozostaje nam przypisać jeszcze do odpowiednich przycisków odpowiednie funkcje oraz napisać je w kontrolerze



Przykładowy AdminController =>

```java
package admin;

import javafx.collections.FXCollections;
import javafx.collections.ListChangeListener;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.*;
import javafx.scene.control.cell.PropertyValueFactory;
import login.dbConnection;

import java.net.URL;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.time.format.DateTimeParseException;
import java.util.Optional;
import java.util.ResourceBundle;

public class AdminController implements Initializable {

    // dla studentow
    @FXML
    private TextField idField, firstNameField, lastNameField, emailField;
    @FXML
    private DatePicker birthDayField;
    @FXML
    private PasswordField passwordField;
    @FXML
    private TableView<StudentData> listOfStudents;

    @FXML
    private TableColumn<StudentData, String> idColumn, firstNameColumn, lastNameColumn;
    @FXML
    private TableColumn<StudentData, String> emailColumn, birthDayColumn, passwordColumn;

    @FXML
    private Button listAllStudentsBtn;

    @FXML
    Tab students_tab;

    @FXML
    private ObservableList<StudentData> studentList;


    private dbConnection db;


    @Override
    public void initialize(URL location, ResourceBundle resources) {
        this.db = new dbConnection();

        ObservableList<StudentData> selectedRow = listOfStudents.getSelectionModel().getSelectedItems();
        selectedRow.addListener(new ListChangeListener() {
            @Override
            public void onChanged(Change c) throws NullPointerException {
                try {
                    StudentData selectedR = listOfStudents.getSelectionModel().getSelectedItem();

                    // zanim ustawi poprawne ID znajdz w tabeli
                    String ssid = findProperID(); // napisz funkcję która pobiera id studenta np z ukrytego pola
                    idField.setText(ssid);

                    firstNameField.setText(selectedR.getFirstName());
                    lastNameField.setText(selectedR.getLastName());
```

```java

                    // ustawianie pola daty
                    DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd.MM.yyyy");
                    try {
                        birthDayField.setValue(LocalDate.parse(selectedR.getBirthDay(), formatter));
                    } catch (DateTimeParseException | NullPointerException e) {
                        birthDayField.setValue(null);
                    }

                    passwordField.setText(selectedR.getPassword());
                    emailField.setText(selectedR.getEmail());

                } catch (NullPointerException e) {
                    idField.setText(null);
                    firstNameField.setText(null);
                    lastNameField.setText(null);
                    birthDayField.setValue(null);
                    passwordField.setText(null);
                    emailField.setText(null);
                }

            }
        });


    }

    private int idCounter = 1;

    @FXML
    public void loadStudentData(ActionEvent actionEvent) {
        String query = "SELECT * from students;";
        try {

            Connection conn = dbConnection.getConnection();
            this.studentList = FXCollections.observableArrayList();

            ResultSet rs = conn.createStatement().executeQuery(query);

            while (rs.next()) {
                this.studentList.add(new StudentData(
                        idCounter++,
                        rs.getString(2),
                        rs.getString(3),
                        rs.getString(4),
                        rs.getString(5),
                        rs.getString(6)
                ));
            }
        } catch (SQLException er) {
            er.printStackTrace();
        }

        this.idColumn.setCellValueFactory(new PropertyValueFactory<StudentData, String>("ID"));
        this.firstNameColumn.setCellValueFactory(new PropertyValueFactory<StudentData, String>("firstName"));
        this.lastNameColumn.setCellValueFactory(new PropertyValueFactory<StudentData, String>("lastName"));
        this.birthDayColumn.setCellValueFactory(new PropertyValueFactory<StudentData, String>("birthDay"));
        this.emailColumn.setCellValueFactory(new PropertyValueFactory<StudentData, String>("email"));
        this.passwordColumn.setCellValueFactory(new PropertyValueFactory<StudentData, String>("password"));

        this.listOfStudents.setItems(null);
        this.listOfStudents.setItems(this.studentList);

        idCounter = 1;
    }
```

```java

    @FXML
    public void addStudent(ActionEvent actionEvent) {
        String insertQuery = "insert into students(firstname, lastname, birthDay, email, password) " +
                "values (?,?,?,?,?)";
        try {
            Connection conn = dbConnection.getConnection();
            PreparedStatement stm = conn.prepareStatement(insertQuery);

            // walidacja danych
            // funkcja containsDigit oraz isEmail do napisania własnoręcznie
            if (containsDigit(this.firstNameField.getText())){
                Alert alert = new Alert(Alert.AlertType.ERROR);
                alert.setTitle("Niewłaściwe dane");
                alert.setHeaderText("Dane nie mogą zawierać cyfr");
                alert.showAndWait();
            }

            if (!isEmail(this.emailField.getText())){
                Alert alert = new Alert(Alert.AlertType.ERROR);
                alert.setTitle("Niewłaściwe dane");
                alert.setHeaderText("Email się nie zgadza");
                alert.showAndWait();
            }

            // można też pobrać id studenta z ukrytego pola
//            stm.setString(1, String.valueOf(this.idField.getText()));
            stm.setString(1, String.valueOf(AdminController.this.firstNameField.getText()));
            stm.setString(2, String.valueOf(AdminController.this.lastNameField.getText()));
            stm.setString(3, String.valueOf(AdminController.this.birthDayField.getEditor().getText()));
            stm.setString(4, String.valueOf(AdminController.this.emailField.getText()));
            stm.setString(5, String.valueOf(AdminController.this.passwordField.getText()));

            stm.execute();
            conn.close();

        } catch (SQLException | NullPointerException e) {
            e.printStackTrace();
        }
    }

    @FXML
    public void deleteStudent(ActionEvent event) {

        StudentData getSelectedRow = listOfStudents.getSelectionModel().getSelectedItem();
        try {
            Connection conn = dbConnection.getConnection();

            if (!getSelectedRow.toString().equals("")) {

                String fn = getSelectedRow.getFirstName();
                String ln = getSelectedRow.getLastName();
                String sID = null;

                PreparedStatement ps = conn.prepareStatement("SELECT student_id FROM students where firstName = ? AND lastName = ? ;");

                ps.setString(1, fn);
                ps.setString(2, ln);

                ResultSet res = ps.executeQuery();

                while (res.next()) {
                    sID = res.getString(1);
                }
                System.out.println(sID);

```

```java
                String deleteQuery = "delete from students where student_id=" + idField.getText() + ";";

                Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
                alert.setTitle("Potwierdzenie usunięcia");
                alert.setHeaderText("Chcesz usunąć studenta?");

                alert.setContentText("" + getSelectedRow.getFirstName() + " " + getSelectedRow.getLastName());

                Optional<ButtonType> result = alert.showAndWait();
                if (result.get() == ButtonType.OK) {
                    PreparedStatement stm = conn.prepareStatement(deleteQuery);
                    stm.execute();

                    listAllStudentsBtn.fire();

                } else {
                    // ... user chose CANCEL or closed the dialog
                }

            }


            conn.close();

        } catch (SQLException e) {
            e.printStackTrace();
        }
    }


    @FXML
    public void editStudent(ActionEvent event) {

        String updateQuery = "update students set firstname=? , lastname=? , birthDay=? , email=?, password=?" +
                "where student_id = ?";

        try {
            Connection conn = dbConnection.getConnection();
            PreparedStatement stm = conn.prepareStatement(updateQuery);

            stm.setString(1, String.valueOf(AdminController.this.firstNameField.getText()));
            stm.setString(2, String.valueOf(AdminController.this.lastNameField.getText()));
            stm.setString(3, String.valueOf(AdminController.this.birthDayField.getEditor().getText()));
            stm.setString(4, String.valueOf(AdminController.this.emailField.getText()));
            stm.setString(5, String.valueOf(AdminController.this.passwordField.getText()));
            stm.setString(6, String.valueOf(AdminController.this.idField.getText()));

            stm.execute();


            conn.close();

        } catch (SQLException | NullPointerException e) {
            e.printStackTrace();
        }

    }


}
```

Oczywiście tak zrealizowany program ma parę „niedociągnięć” ale poprawę funkcjonalności zostawię już osobistym gustom ;)

Ważne, że tak zrealizowane zadanie „u mnie działa” ;)

Powodzenia