Uniwersytet Marii Curie-Skłodowskiej

System logowania studentów UMCS

Projekt na Inżynierię Oprogramowania

Kowalczyk Łukasz – Grupa Środowa na 10:05
2019-05-30

Przypadki Użycia

Rejestracja

Przypadek Użycia	Użytkownik zakład konto.
Scenariusz	Prawidłowe dodanie konta użytkownika do systemu.
Warunki wstępne	Użytkownik nie posiada konta w Systemie.
Niezmienniki	Użytkownik chce założyć konto w systemie.
Opis	Użytkownik wchodzi na okno zawierające formularz rejestracji, wprowadza dane i wysyła je do systemu. System waliduje dane.
	Scenariusz powodzenia:
	Wprowadzone dane są prawidłowe. System wyświetla komunikat powodzenia i użytkownik zostaje przeniesiony do okna logowania.
	Scenariusz niepowodzenia (złe dane):
	Użytkownik dostaje komunikat o nieprawidłowych danych. Powrót do formularza.
Warunki końcowe	Użytkownik zostaje zarejestrowany, a jego dane zostają zapisane w bazie danych systemu.

Logowanie do systemu

Przypadek użycia	Użytkownik loguje się do systemu.
Scenariusz	Zalogowanie użytkownika do systemu.
Warunki wstępne	Użytkownik nie jest zalogowany do systemu i posiada konto.
Niezmienniki	Użytkownik chce się zalogować do systemu.
Opis	Użytkownik wchodzi na okno logowania zawierającą formularz logowania, wprowadza swój login i hasło, a następnie zatwierdza dane, wysyłając je do systemu.
	Główny scenariusz powodzenia:

	Wprowadzone dane są prawidłowe. Użytkownik zostaje zalogowany do systemu i przeniesiony do panelu użytkownika.
	Scenariusz niepowodzenia:
	Użytkownik dostaje komunikat o błędnych danych i zostaje przeniesiony do okna logowania.
Warunki końcowe	Użytkownik jest zalogowany do systemu

Wyświetl oceny

Przypadek użycia	Użytkownik wyświetla oceny.
Scenariusz	Prawidłowe wyświetlenie ocen.
Warunki wstępne	Użytkownik jest zalogowany do systemu jako student.
Niezmienniki	Użytkownik chce wyświetlić oceny.
Opis	Użytkownik wybiera opcje wyświetl oceny. Następnie wybiera kurs, na który jest zapisany. System wyświetla wszystkie oceny przypisane do jego kursu.
Warunki końcowe	Oceny zostają wyświetlone.

Wstaw Oceny

Przypadek użycia	Użytkownik dodaje oceny do systemu.
Scenariusz	Prawidłowe dodanie ocen do systemu.
Warunki wstępne	Użytkownik jest zalogowany jako prowadzący.
Niezmienniki	Użytkownik chce dodać oceny do systemu.
Opis	Użytkownik wybiera opcje wstaw oceny. Następnie wybiera kurs i studenta należącego do danego kursu. Wpisuje ocenę i zatwierdza. Wprowadzona ocena zostanie dodana do bazy danych.
Warunki końcowe	Oceny zostają dodane do bazy danych w systemie.

Utwórz kurs

Przypadek użycia	Użytkownik tworzy nowy kurs.
Scenariusz	Prawidłowe utworzenie kursu.
Warunki wstępne	Użytkownik jest zalogowany jako prowadzący.
Niezmienniki	Użytkownik chce utworzyć nowy kurs.
Opis	Użytkownik wybiera opcję utworzenia kursu, podaje nazwę nowego kursu i zatwierdza. Dane zostają dodane do bazy danych w systemie.
	Główny scenariusz powodzenia:
	Użytkownik dostaje komunikat o powodzeniu. Kurs zostaje dodany do bazy danych w systemie. Okno wpisywania kursów zostaje wyczyszczone.
	Scenariusz niepowodzenia (duplikat):
	Zatwierdzenie nie prowadzi do żadnych zmian.
Warunki końcowe	Kurs zostaje dodany do bazy danych w systemie.

Zapisz się na kurs

Przypadek użycia	Użytkownik zapisuje się na kurs.
Scenariusz	Prawidłowe zapisanie użytkownika do kursu.
Warunki wstępne	Użytkownik jest zalogowany jako student.
Niezmienniki	Użytkownik chce zapisać się na kurs.
Opis	Użytkownik wybiera opcję kurs. Wybiera kurs z listy dostępnych kursów i zatwierdza wybór. Użytkownik zostaje zapisany na kurs i dane zostają wprowadzone do bazy danych systemu.
Warunki końcowe	Użytkownik zostaje zapisany na kurs i dane zostają wysłane do bazy danych systemu.

Wstaw materiały

Przypadek użycia	Użytkownik dodaje nowe materiały do kursu.
Scenariusz	Prawidłowe dodanie materiałów do kursu.
Warunki wstępne	Użytkownik jest zalogowany jako prowadzący i jest zapisany na kurs.
Niezmienniki	Użytkownik chce wstawić materiały.
Opis	Użytkownik wybiera opcje kursu. System wyświetla wszystkie kursy przypisane do użytkownika. Użytkownik wybiera kurs i opcję dodaj materiały. Użytkownik zostaje przeniesiony do okna wyboru pliku, wybiera pliki do wstawienia i zatwierdza wybór. System wysyła plik do bazy danych systemu.
Warunki końcowe	Materiały zostają dodane do bazy danych systemu.

Pobierz materiały

Przypadek użycia	Użytkownik pobiera materiały.
Scenariusz	Prawidłowe pobranie materiałów z kursu.
Warunki wstępne	Użytkownik jest zalogowany jako student i jest zapisany na kurs.
Niezmienniki	Użytkownik chce pobrać materiały.
Opis	Użytkownik wybiera opcje kursu. System wyświetla kursy, na które jest zapisany. Użytkownik wybiera kurs i opcję pobierz materiały, następnie zostaje przeniesiony do wyboru folderu do zapisu plików i zatwierdza wybór. Pliki zostają pobrane z bazy danych systemu i skopiowane do wybranego folderu na komputerze użytkownika.
Warunki końcowe	Materiały zostają zapisane na komputerze użytkownika.

Diagram przypadków użycia

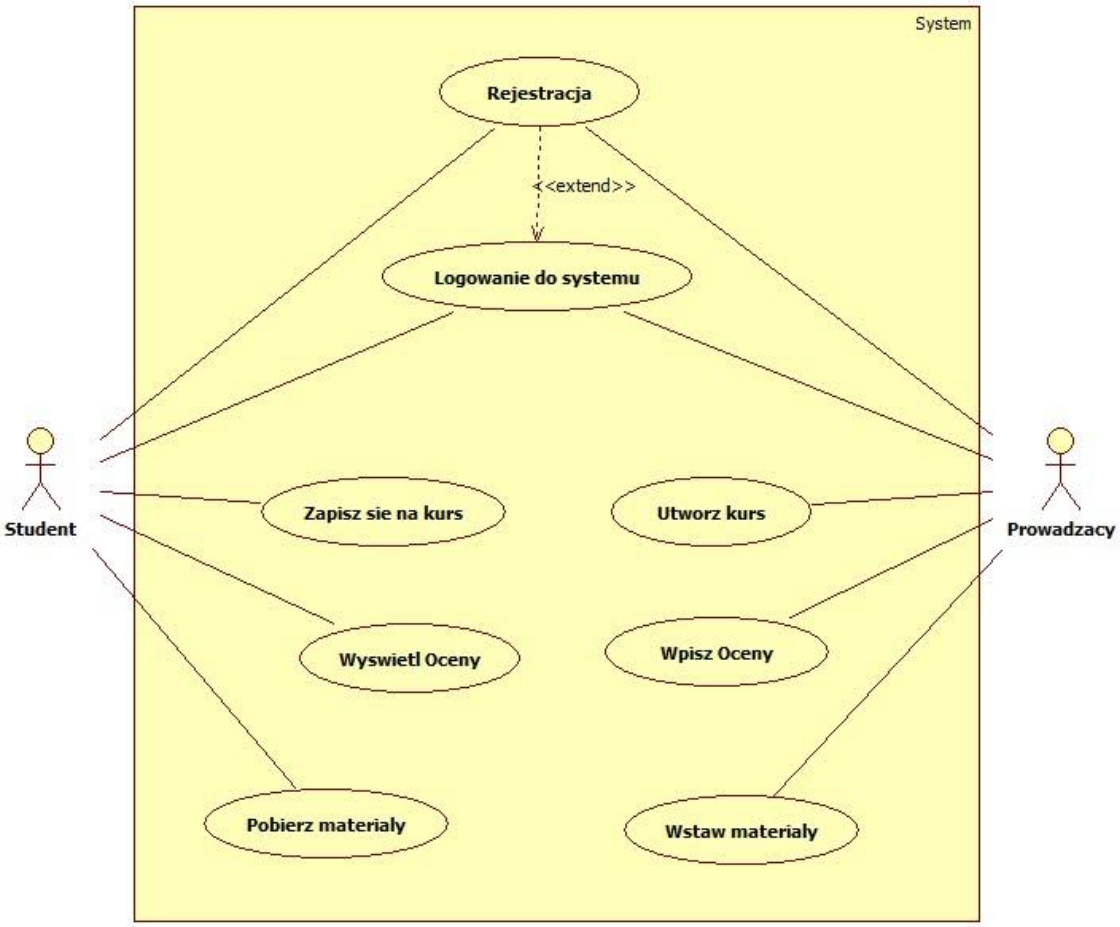
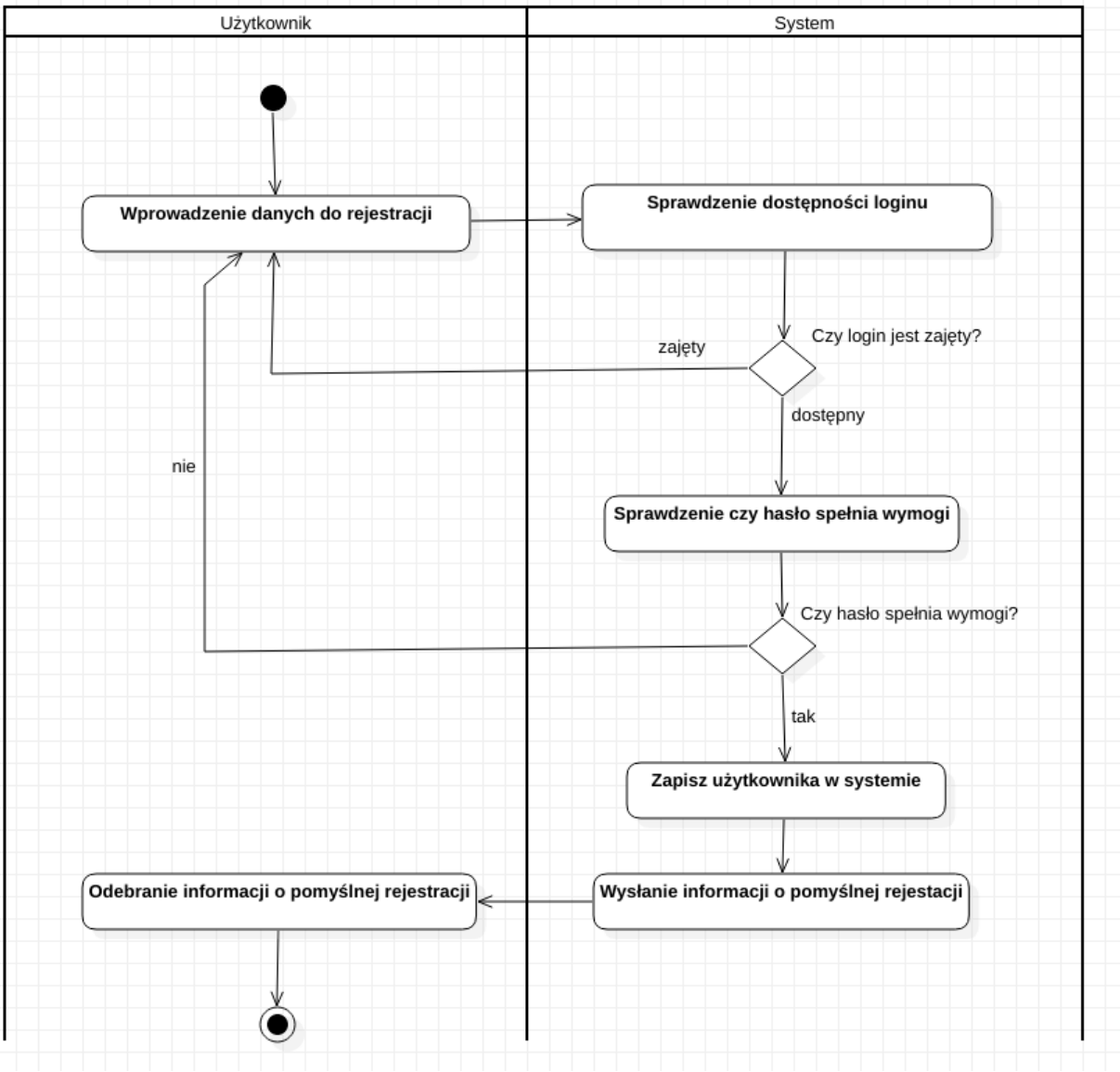
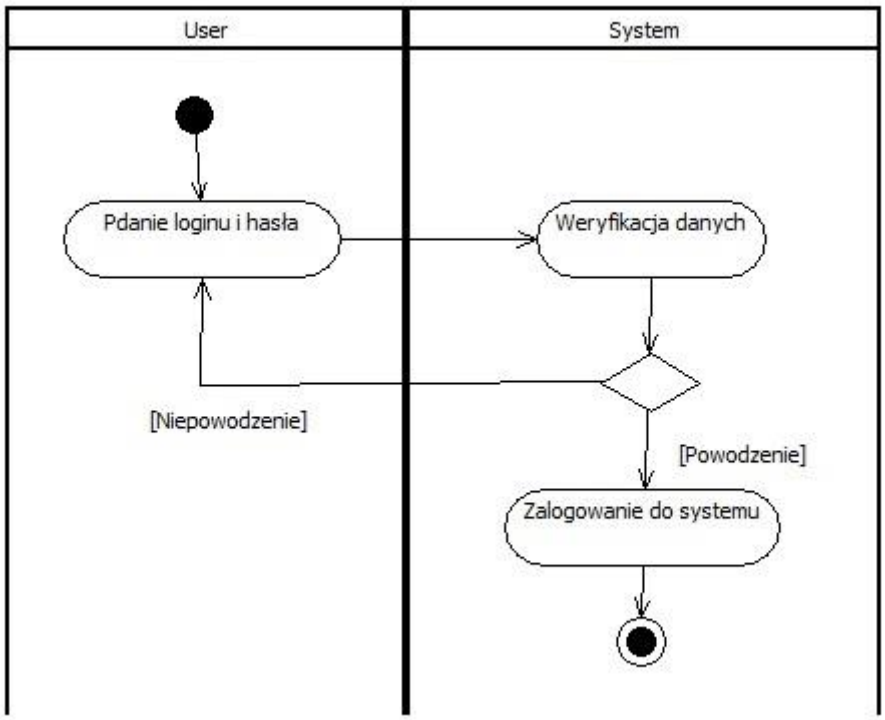

System
Rejestracja
<<extend>>
Logowanie do systemu
Student
Zapisz sie na kurs
Utworz kurs
Prowadzacy
Wyswietl Oceny
Wpisz Oceny
Pobierz materialy
Wstaw materialy

Diagram Aktywności

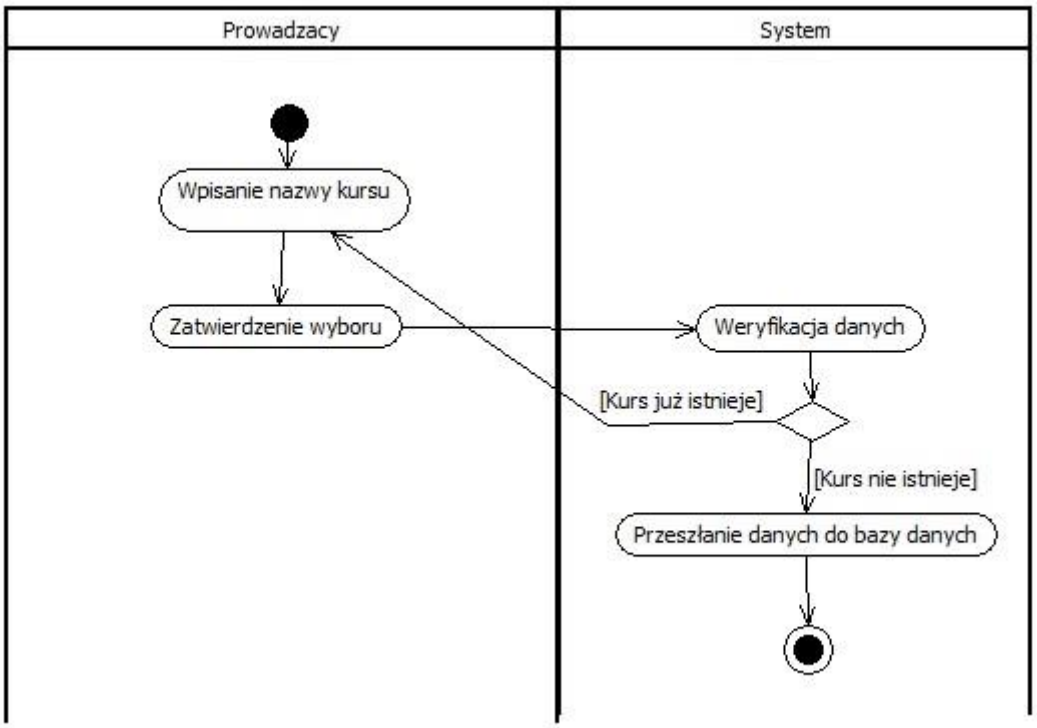
Rejestracja



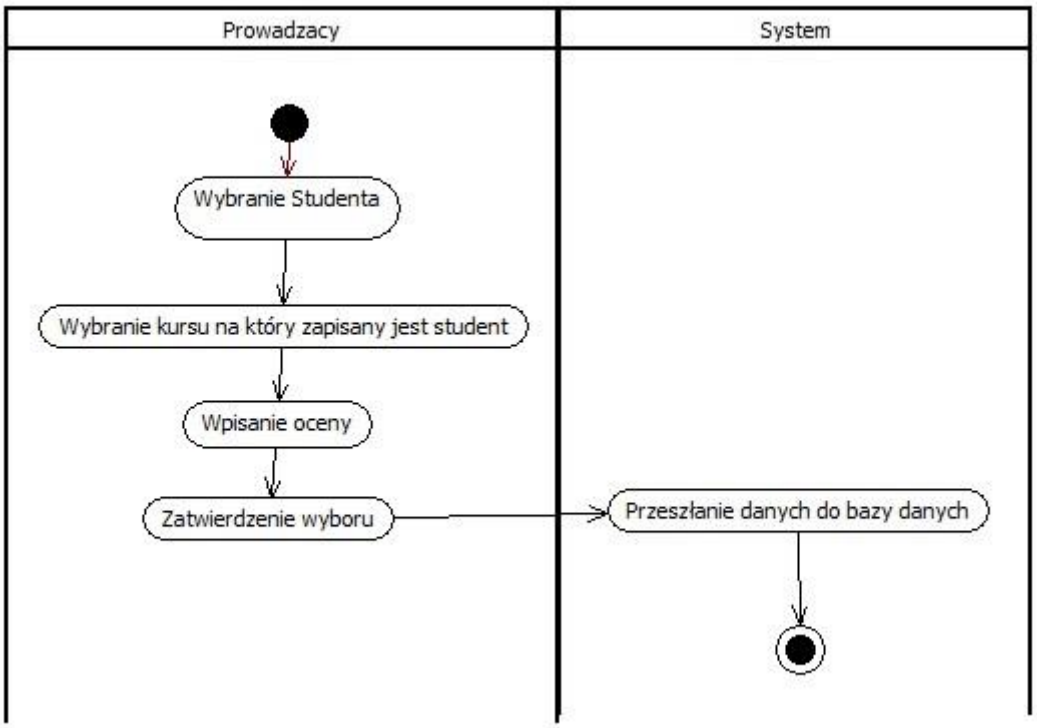
Logowanie



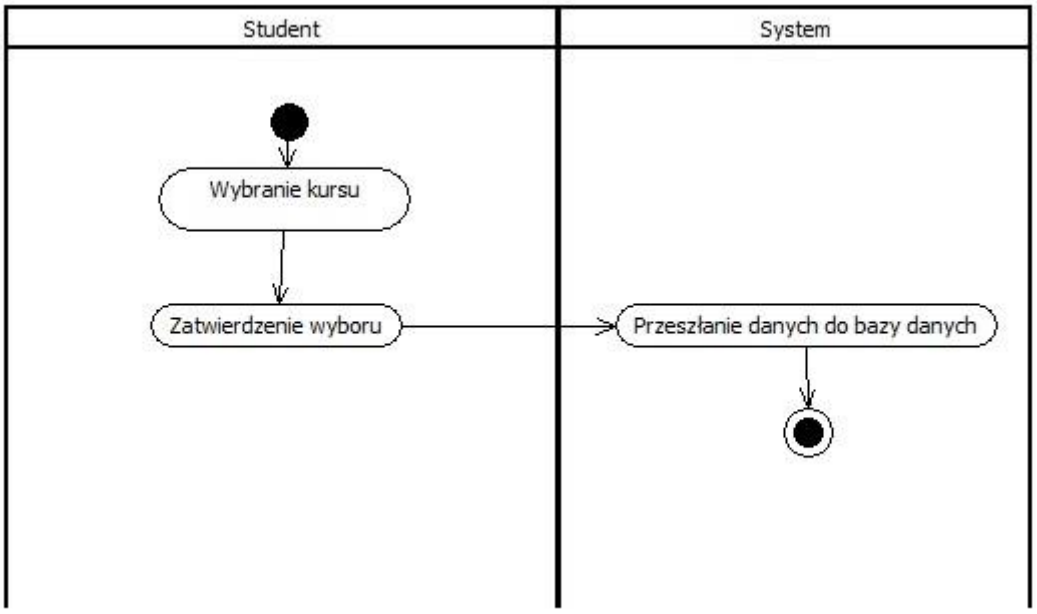
Utwórz kurs



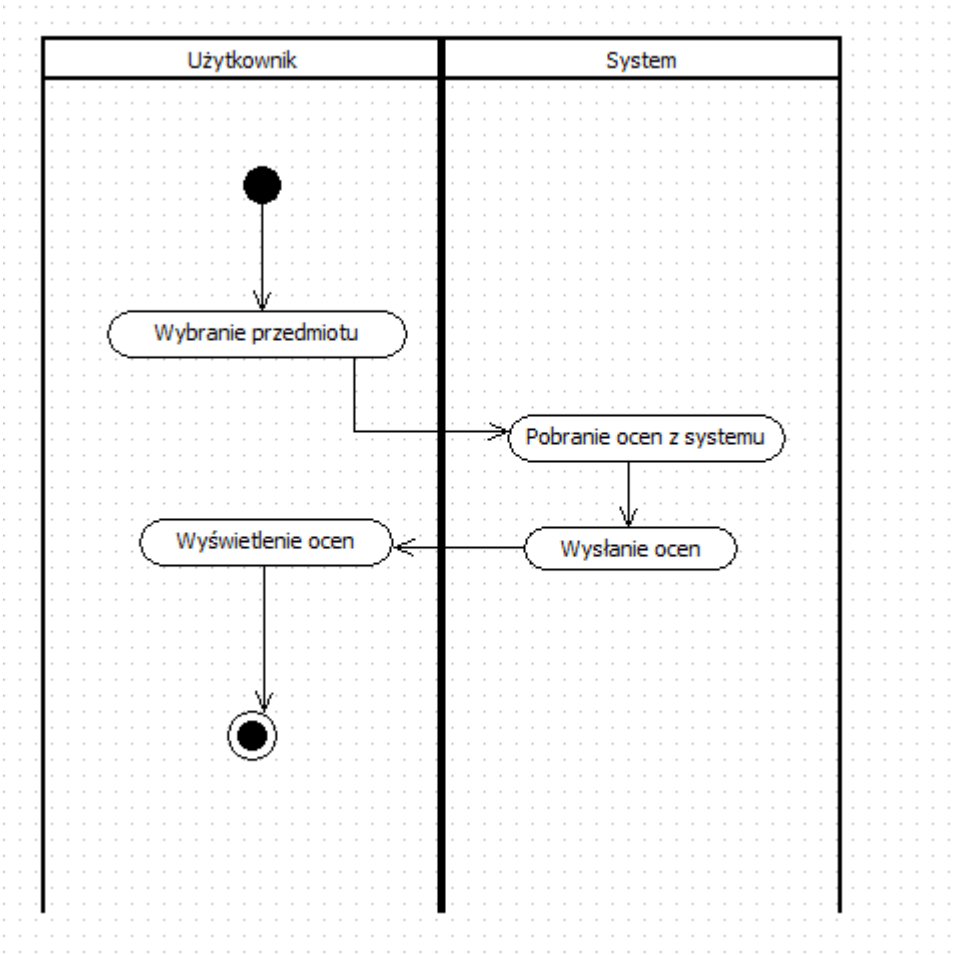
Wstaw oceny



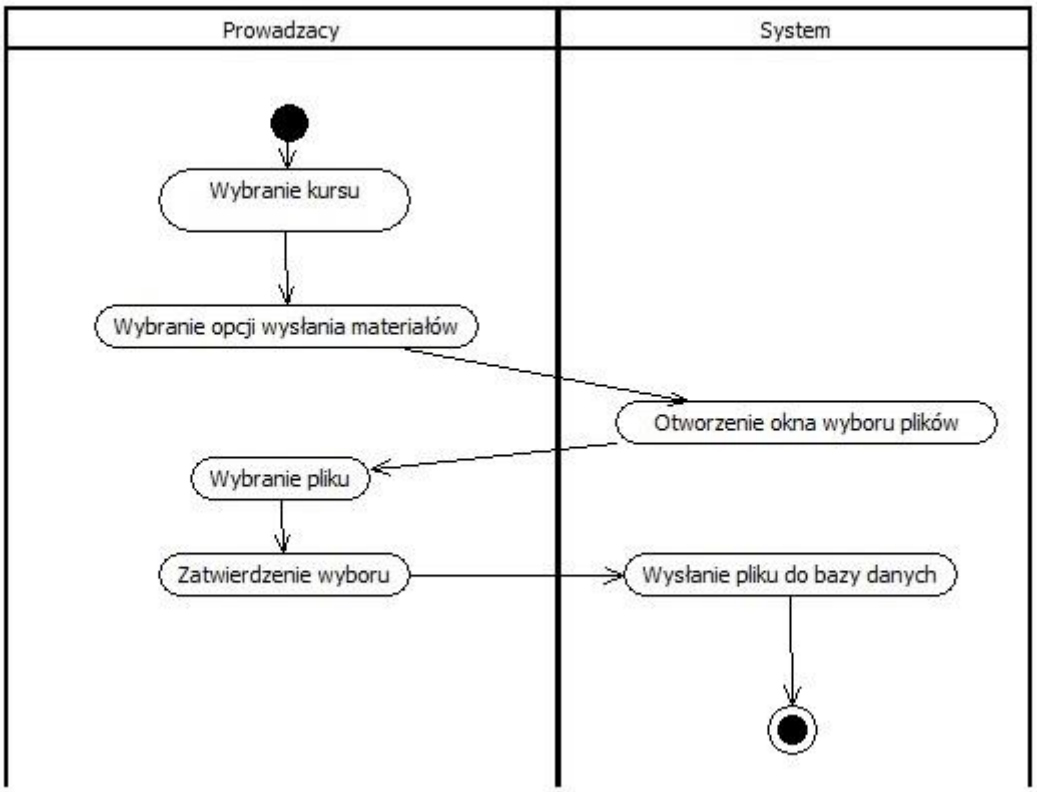
Zapisz się na Kurs



Wyświetl Oceny



Wstaw Materiały



Pobierz Materiały

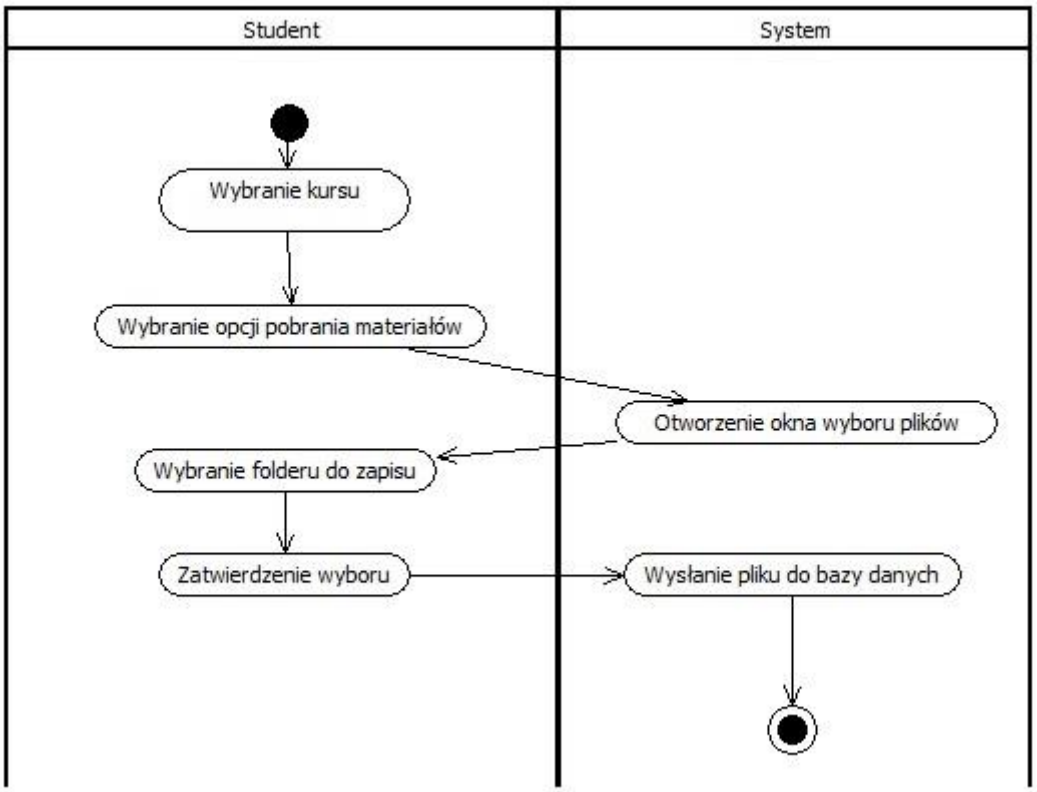


Diagram Klas

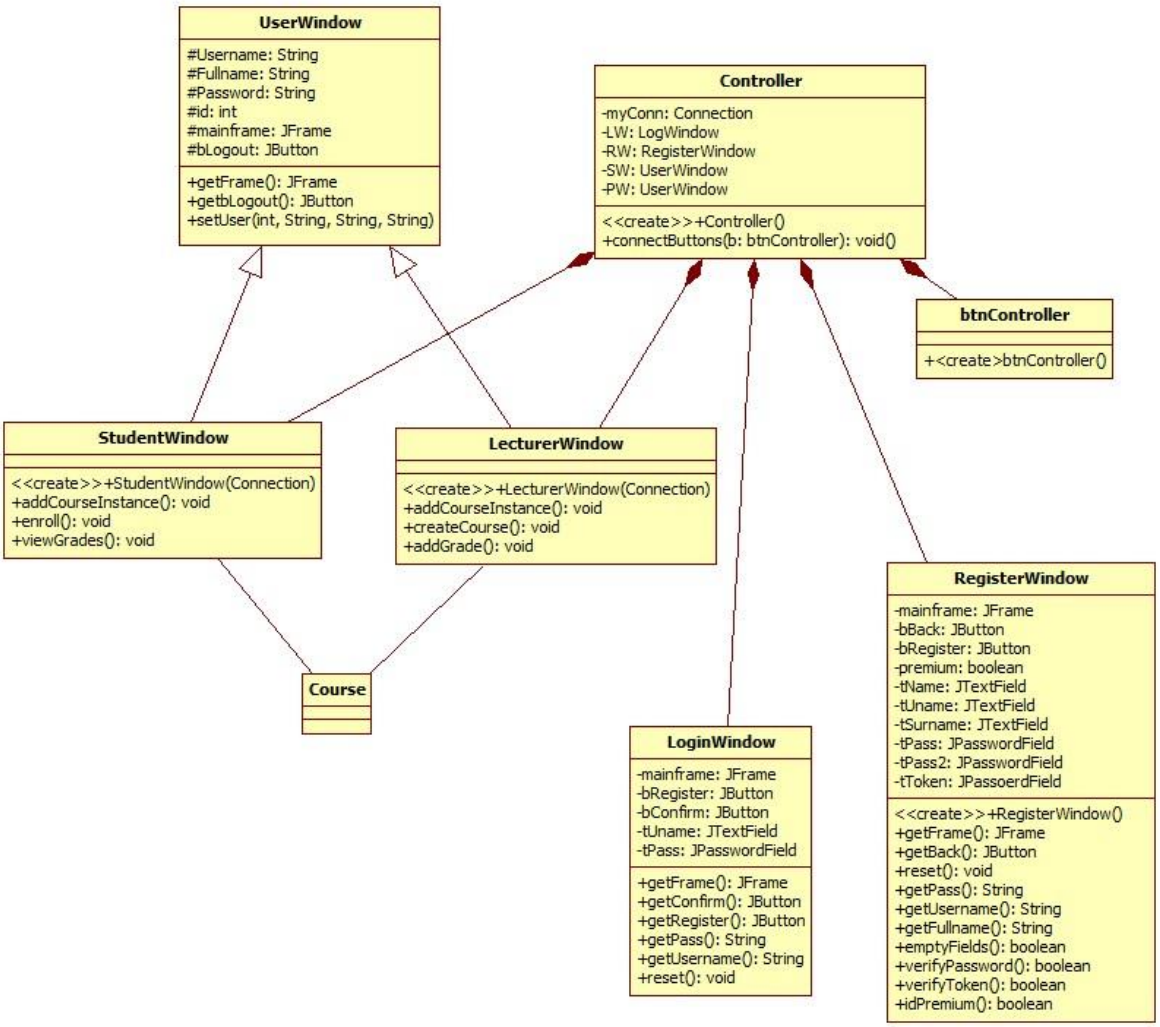
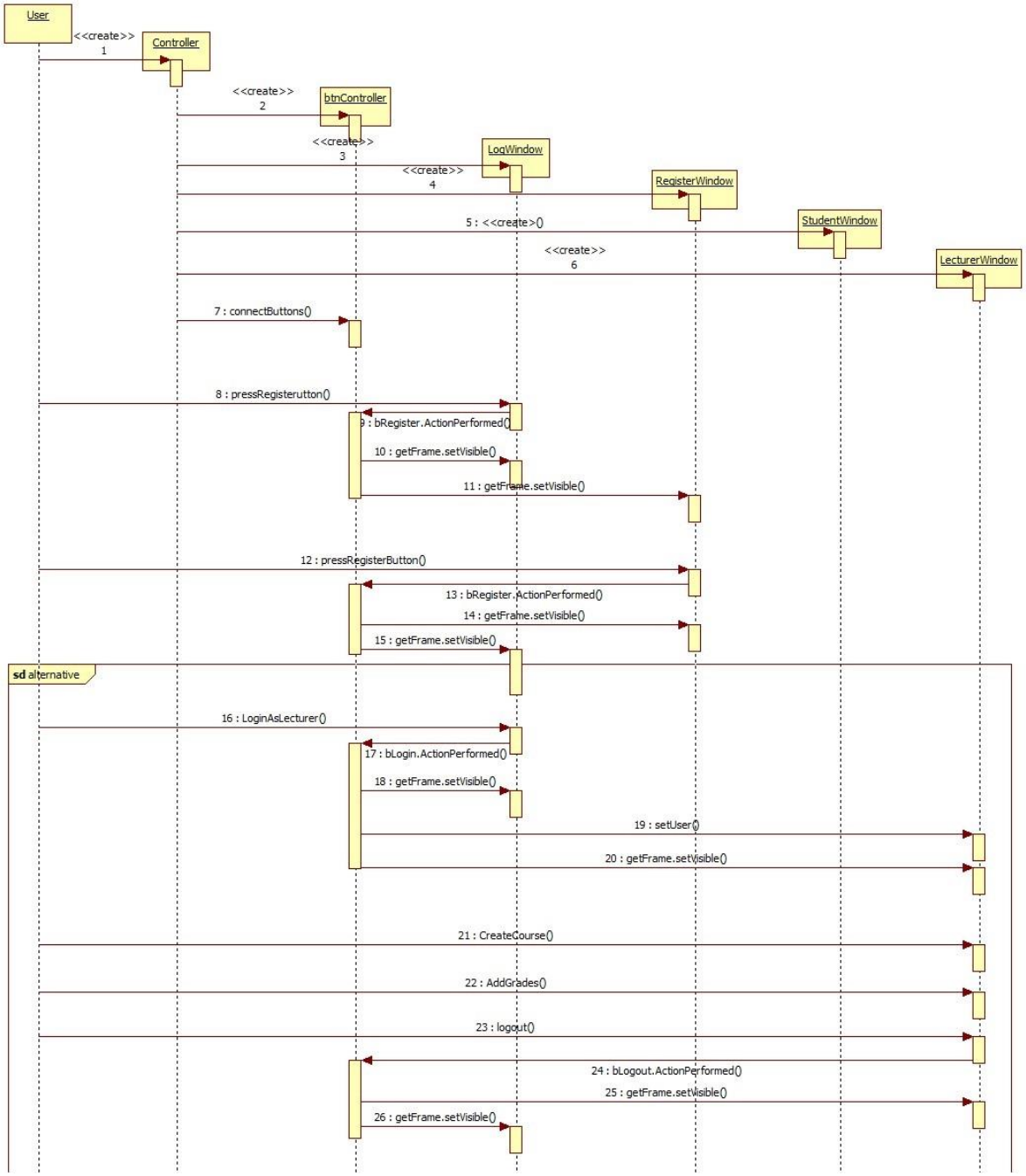
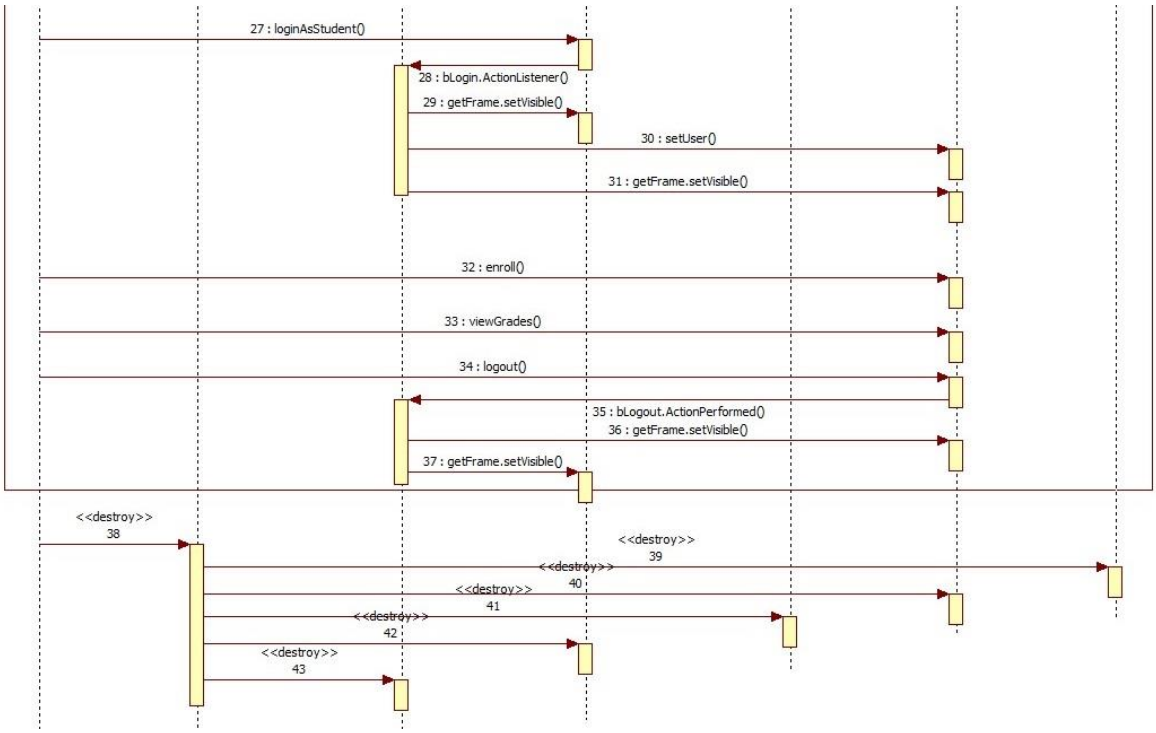


Diagram Sekwencji




27 : loginAsStudent()
28 : bLogin.ActionListener()
29 : getFrame.setVisible()
30 : setUser()
31 : getFrame.setVisible()
32 : enroll()
33 : viewGrades()
34 : logout()
35 : bLogout.ActionPerformed()
36 : getFrame.setVisible()
37 : getFrame.setVisible()
<<destroy>>
38
<<destroy>>
39
<<destroy>>
40
<<destroy>>
41
<<destroy>>
42
<<destroy>>
43

Kod programu

Controller.java

```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class Controller {

    private Connection myConn;
    private LogWindow LW;
    private RegisterWindow RW;
    private UserWindow SW,PW;

    private class btnController implements ActionListener
    {
        @Override
        public void actionPerformed(ActionEvent ae) {
            Object src = ae.getSource();
            if(src==LW.bConfirm)
            {
                try {
                    String query = "select * from Users where Username=?
and Password=?";
                    PreparedStatement mySt =
myConn.prepareStatement(query);
                    mySt.setString(1,LW.getUsername());
                    mySt.setString(2,LW.getPass());
                    ResultSet rs=mySt.executeQuery();
                    if(rs.next())
                    {
                        if(rs.getString("Type").equals("Student")) {

SW.setUser(rs.getInt("id"),rs.getString("Username"),rs.getString("Fullname"
),rs.getString("Password"));
                            SW.getFrame().setVisible(true);
                        }
                        else {

PW.setUser(rs.getInt("id"),rs.getString("Username"),rs.getString("Fullname"
),rs.getString("Password"));
                            PW.getFrame().setVisible(true);
```

```
                        }
                        LW.getFrame().setVisible(false);
                    }
                    else
                    {
                        JOptionPane.showMessageDialog(null, "ZĹ‚y login lub
hasĹ‚o", "InfoBox: " + "BĹ‚Ä…d", JOptionPane.INFORMATION_MESSAGE);
                    }
                }
                catch (Exception e)
                {
                    e.printStackTrace();
                }
            }
            if(src==LW.bRegister)
            {
                LW.getFrame().setVisible(false);
                RW.getFrame().setVisible(true);
            }
            if(src==RW.bRegister)
            {
                if(!RW.emptyFields() && RW.verifyPassword() &&
RW.verifyToken()) {

                    String query = "select * from Users where Username=?";
                    String query2 = "insert into Users "
                            + "(Username, Fullname, Password, Type)"
                            + "VALUES (?,?,?,?)";
                    try {
                        PreparedStatement ps =
myConn.prepareStatement(query);
                        PreparedStatement ps2 =
myConn.prepareStatement(query2);
                        ps.setString(1,RW.getUserame());
                        ResultSet rs = ps.executeQuery();
                        if(rs.next())
                        {
                            JOptionPane.showMessageDialog(null, "Podany
login jest juĹ‚ zajÄ™ty", "InfoBox: " + "BĹ‚ad",
JOptionPane.INFORMATION_MESSAGE);
                        }
                        else
                        {
                            String type;
                            if(RW.isPremium())
                                type="Lecturer";
                            else
                                type="Student";
                            ps2.setString(1,RW.getUserame());
```

```
                            ps2.setString(2,RW.getFullname());
                            ps2.setString(3,RW.getPass());
                            ps2.setString(4,type);
                            ps2.executeUpdate();
                            RW.reset();
                            RW.getFrame().setVisible(false);
                            LW.getFrame().setVisible(true);
                        }
                    }
                    catch(Exception e)
                    {
                        e.printStackTrace();
                    }
                }
                else
                {
                    JOptionPane.showMessageDialog(null, "Niepoprawne dane",
"InfoBox: " + "Błąd", JOptionPane.INFORMATION_MESSAGE);
                }
            }
            if(src==RW.bBack)
            {
                RW.reset();
                LW.reset();
                RW.getFrame().setVisible(false);
                LW.getFrame().setVisible(true);
            }
            if(src==PW.bLogout)
            {
                LW.reset();
                PW.getFrame().setVisible(false);
                LW.getFrame().setVisible(true);

            }
            if(src==SW.bLogout)
            {
                LW.reset();
                SW.getFrame().setVisible(false);
                LW.getFrame().setVisible(true);
            }
        }
    }

    public Controller()
    {
        try {
            myConn =
DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/UMCS?useUnicode=tr
```

```
ue&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTim
ezone=UTC", "root", "*888rootSQL");
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
        btnController controller = new btnController();
        LW = new LogWindow();
        RW = new RegisterWindow();
        PW = new LecturerWindow(myConn);
        SW = new StudentWindow(myConn);
        connectButtons(controller);
    }

    public void connectButtons(btnController b)
    {
        LW.bRegister.addActionListener(b);
        LW.bConfirm.addActionListener(b);
        RW.bRegister.addActionListener(b);
        RW.bBack.addActionListener(b);
        PW.getBLogout().addActionListener(b);
        SW.getBLogout().addActionListener(b);
    }
}
```

LogWindow.java

```
import javax.swing.*;

public class LogWindow {

    private JFrame mainframe;
    public JButton bConfirm,bRegister;
    private JTextField tUname;
    private JPasswordField tPass;

    public LogWindow()
    {
        mainframe = new JFrame("Logowanie");
        mainframe.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        mainframe.setLayout(null);
        mainframe.setResizable(false);
        mainframe.setLocationRelativeTo(null);
        int w=400,h=200;
        mainframe.setSize(w,h);

        JPanel mainpanel = new JPanel();
        mainpanel.setBounds(0,0,w,h);
```

```
        mainpanel.setLayout(null);

        bConfirm = new JButton("Zaloguj");
        bConfirm.setBounds(90,130,100,30);

        bRegister = new JButton("Rejestracja");
        bRegister.setBounds(210,130,100,30);

        JLabel lUname = new JLabel("Login:");
        lUname.setBounds(70,30,60,30);

        JLabel lPass = new JLabel("HasŁ,o:");
        lPass.setBounds(70,70,60,30);

        tUname = new JTextField();
        tUname.setBounds(140,30,190,30);

        tPass = new JPasswordField();
        tPass.setBounds(140,70,190,30);

        mainpanel.add(bConfirm);
        mainpanel.add(bRegister);
        mainpanel.add(lUname);
        mainpanel.add(lPass);
        mainpanel.add(tUname);
        mainpanel.add(tPass);

        mainframe.add(mainpanel);

        mainframe.setVisible(true);
    }
    public JFrame getFrame(){return mainframe;}

    public String getUsername(){ return tUname.getText();}

    public String getPass(){ return new String(tPass.getPassword());}

    public void reset()
    {
        tUname.setText("");
        tPass.setText("");
    }
}
```

RegisterWindow.java

```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class RegisterWindow {

    private JFrame mainframe;
    private JTextField tName,tSurname,tUname;
    private JPasswordField tPass,tPass2,tToken;
    public JButton bRegister,bBack;
    private boolean premium;
    public RegisterWindow()
    {
        mainframe = new JFrame("Rejstracja");
        mainframe.setLocationRelativeTo(null);
        mainframe.setLayout(null);
        mainframe.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        int w=400,h=400;
        mainframe.setSize(w,h);
        premium=false;

        JPanel mainpanel = new JPanel();
        mainpanel.setBounds(0,0,w,h);
        mainpanel.setLayout(null);

        JLabel lName = new JLabel("Imie:");
        lName.setBounds(50,80,100,30);
        mainpanel.add(lName);

        JLabel lSurname = new JLabel("Nazwisko:");
        lSurname.setBounds(50,120,100,30);
        mainpanel.add(lSurname);

        JLabel lUname = new JLabel("Login:");
        lUname.setBounds(50,160,100,30);
        mainpanel.add(lUname);

        JLabel lPass = new JLabel("HasŁ,o:");
        lPass.setBounds(50,200,100,30);
        mainpanel.add(lPass);

        JLabel lPass2 = new JLabel("Potwierdzenie:");
        lPass2.setBounds(50,240,100,30);
        mainpanel.add(lPass2);

        JLabel lToken = new JLabel("Token:");
        lToken.setBounds(50,280,100,30);
        mainpanel.add(lToken);
```

```
tName = new JTextField();
tName.setBounds(150,80,200,30);
mainpanel.add(tName);

tSurname = new JTextField();
tSurname.setBounds(150,120,200,30);
mainpanel.add(tSurname);

tUname = new JTextField();
tUname.setBounds(150,160,200,30);
mainpanel.add(tUname);

tPass = new JPasswordField();
tPass.setBounds(150,200,200,30);
mainpanel.add(tPass);

tPass2 = new JPasswordField();
tPass2.setBounds(150,240,200,30);
mainpanel.add(tPass2);

tToken = new JPasswordField();
tToken.setBounds(150,280,200,30);
tToken.setEnabled(false);
mainpanel.add(tToken);

bRegister = new JButton("Zarejestruj");
bRegister.setBounds(90,330,100,30);
mainpanel.add(bRegister);

bBack = new JButton("Cofnij");
bBack.setBounds(210,330,100,30);
mainpanel.add(bBack);

ButtonGroup UserType = new ButtonGroup();

JRadioButton rbStudent = new JRadioButton("Student",true);
rbStudent.setBounds(90,30,100,30);
rbStudent.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent actionEvent) {
        tToken.setText("");
        tToken.setEnabled(false);
        premium=true;
    }
});
UserType.add(rbStudent);
mainpanel.add(rbStudent);
```

```
        JRadioButton rbLecturer = new JRadioButton("ProwadzÄ…cy",false);
        rbLecturer.setBounds(210,30,100,30);
        rbLecturer.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent actionEvent) {
                tToken.setEnabled(true);
                premium=true;
            }
        });
        UserType.add(rbLecturer);
        mainpanel.add(rbLecturer);

        mainframe.add(mainpanel);

        mainframe.setVisible(false);
    }

    public JFrame getFrame()
    {
        return mainframe;
    }

    public String getUserame()
    {
        return tUname.getText();
    }

    public String getPass()
    {
        return new String(tPass.getPassword());
    }

    public boolean isPremium()
    {
        return premium;
    }

    public boolean verifyToken()
    {
        if(isPremium())
            return new String(tToken.getPassword()).equals("P2PP2W");
        return true;
    }

    public boolean verifyPassword()
    {
        return new String(tPass.getPassword()).equals(new
String(tPass2.getPassword()));
    }
```

```
    public String getFullname()
    {
        return tName.getText()+" "+tSurname.getText();
    }

    public boolean emptyFields()
    {
        if(tUname.getText().isEmpty())
            return true;
        if(tName.getText().isEmpty())
            return true;
        if(tSurname.getText().isEmpty())
            return true;
        if(new String(tPass.getPassword()).isEmpty())
            return true;
        return false;
    }

    public void reset()
    {
        tUname.setText("");
        tSurname.setText("");
        tToken.setText("");
        tPass.setText("");
        tPass2.setText("");
        tName.setText("");
    }
}
```

UserWindow.java

```
import javax.swing.*;
import java.awt.*;
import java.sql.Connection;

public class UserWindow {

    protected int id;
    protected String Username, Name, Pass;
    protected Connection myConn;
    protected JFrame mainframe;
    protected JButton bLogout,bCourse,bGrades;
    protected JLabel lName;
    public UserWindow(Connection myConn)
    {
        this.id=-1;
        this.Name="";
        this.Username="";
```

```
        this.Pass="";
        this.myConn = myConn;
        int w=300,h=400;
        mainframe = new JFrame("Panel Usera");
        mainframe.setSize(w,h);
        mainframe.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        mainframe.setResizable(false);
        mainframe.setLocationRelativeTo(null);
        mainframe.setLayout(null);

        JPanel mainpanel = new JPanel();
        mainpanel.setBounds(0,0,w,h);
        mainpanel.setLayout(null);

        lName = new JLabel("User");
        lName.setBounds(50,50,200,50);
        lName.setFont(new Font("Verdana",Font.BOLD,25));
        lName.setHorizontalAlignment(SwingConstants.CENTER);
        mainpanel.add(lName);

        bCourse = new JButton("Kursy");
        bCourse.setBounds(50,150,200,50);
        mainpanel.add(bCourse);

        bGrades = new JButton("Oceny");
        bGrades.setBounds(50,220,200,50);
        mainpanel.add(bGrades);

        bLogout = new JButton("Wyloguj");
        bLogout.setBounds(50,290,200,50);
        mainpanel.add(bLogout);

        mainframe.add(mainpanel);
        mainframe.setVisible(false);
    }

    public JFrame getFrame() { return mainframe; }

    public JButton getBLogout() { return bLogout; }

    public void setUser(int id, String Username, String Name, String Pass)
    {
        this.id=id;
        this.Name=Name;
        this.Username=Username;
        this.Pass=Pass;
        lName.setText(Name);
    }
}
```

StudentWindow.java

```
import javax.imageio.ImageIO;
import javax.swing.*;
import javax.swing.filechooser.FileSystemView;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.InputStream;
import java.sql.*;

public class StudentWindow extends UserWindow{

    private int counter;
    public StudentWindow(Connection myConn)
    {
        super(myConn);
        mainframe.setTitle("Panel Studenta");

        bCourse.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent ae) {
                enroll();
            }
        });

        bGrades.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent actionEvent) {
                displayGrades();
            }
        });
    }

    private void addCourseInstance(String name,JPanel p, GridBagConstraints
gbc)
    {
        JPanel pi = new JPanel();
        pi.setLayout(null);
        JLabel l = new JLabel(name);
        l.setBounds(20, 10, 200, 30);
        JButton b = new JButton("Pliki");
        b.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent actionEvent) {
                try
                {
                    String query = "select M.File "
```

```
                            +"from Materials M, Courses C "
                            +"where M.Course_id = C.id and C.Name = ?";
                    PreparedStatement ps = myConn.prepareStatement(query);
                    ps.setString(1,name);
                    ResultSet rs = ps.executeQuery();
                    int i=0;
                    String path="";
                    int returnValue = 1;
                    while(rs.next())
                    {
                        if(i==0)
                        {
                            JFileChooser jfc = new
JFileChooser(FileSystemView.getFileSystemView().getHomeDirectory());

jfc.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
                            returnValue = jfc.showOpenDialog(null);
                            if(returnValue==JFileChooser.APPROVE_OPTION)
                            {

path=jfc.getSelectedFile().getAbsolutePath();
                            }
                        }
                        if(returnValue==JFileChooser.APPROVE_OPTION)
                        {
                            Blob b = rs.getBlob("File");
                            InputStream in = b.getBinaryStream();
                            BufferedImage im = ImageIO.read(in);
                            File f = new
File(path+"/Material"+Integer.toString(i+1)+".png");
                            ImageIO.write(im,"png",f);
                        }
                        i+=1;
                    }
                }
                catch (Exception e)
                {
                    e.printStackTrace();
                }
            }
        });
        b.setBounds(280, 10, 80, 30);
        pi.add(b);
        pi.add(l);
        gbc.gridx = 0;
        gbc.gridy = counter;
        gbc.ipadx = 380;
        gbc.ipady = 50;
        gbc.fill = GridBagConstraints.HORIZONTAL;
```

```
        p.add(pi, gbc);
        counter+=1;
    }

    private void enroll()
    {
        JFrame courseframe = new JFrame("Kursy");
        courseframe.setLocationRelativeTo(null);

courseframe.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
        courseframe.setResizable(false);
        courseframe.setLayout(null);
        courseframe.setSize(500,600);

        JPanel p = new JPanel();
        GridBagConstraints gbc = new GridBagConstraints();
        GridBagLayout gbl = new GridBagLayout();
        p.setLayout(gbl);
        counter=0;
        try {
            String query = "Select C.Name "
                    + "from Courses C,Registry R, Users U "
                    + "where U.id = ? and U.id = R.Student_id and
R.Course_id = C.id";
            PreparedStatement ps1 = myConn.prepareStatement(query);
            ps1.setInt(1,id);
            ResultSet rs = ps1.executeQuery();
            while(rs.next()) {
                addCourseInstance(rs.getString("Name"),p,gbc);
            }
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
        JScrollPane scroll = new
JScrollPane(p,JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED,JScrollPane.HORIZONT
AL_SCROLLBAR_AS_NEEDED);
        scroll.getVerticalScrollBar().setPreferredSize(new
Dimension(15,0));
        scroll.setBounds(50,100,400,400);

        JLabel lCour = new JLabel("Twoje Kursy");
        lCour.setBounds(50,30,400,50);
        lCour.setFont(new Font("Verdana",Font.BOLD,30));
        lCour.setHorizontalAlignment(SwingConstants.CENTER);
        courseframe.add(lCour);

        JComboBox courseBox = new JComboBox();
```

```
        courseBox.setBounds(50,520,300,30);
        courseframe.add(courseBox);

        try{
            String query = "Select Name from Courses "
                    + "where Name not in (Select C.Name "
                    + "from Courses C,Registry R, Users U "
                    + "where U.id = ? and U.id = R.Student_id and
R.Course_id = C.id "
                    + "group by C.Name)";
            PreparedStatement pr = myConn.prepareStatement(query);
            pr.setInt(1,id);

            ResultSet rs = pr.executeQuery();
            int i=0;
            while(rs.next())
            {
                i+=1;
                courseBox.addItem(rs.getString("Name"));
            }
            if(i==0)
            {
                Statement pr2 = myConn.createStatement();
                ResultSet rs2 = pr2.executeQuery("Select Name from
Courses");
                while(rs2.next())
                {
                    courseBox.addItem(rs2.getString("Name"));
                }
            }
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }

        JButton bEnroll = new JButton("Zapisz");
        bEnroll.setBounds(370,520,80,30);
        bEnroll.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent actionEvent) {
                try
                {
                    String result = (String) courseBox.getSelectedItem();
                    String query = "insert into Registry "
                            + "(Course_id,Student_id ) "
                            + "select C.id, ? "
                            + "from Courses C "
                            + "where C.Name = ?";
```

```
                    System.out.println(result);
                    PreparedStatement ps = myConn.prepareStatement(query);
                    ps.setInt(1,id);
                    ps.setString(2,result);
                    ps.executeUpdate();
                    courseBox.removeItem(result);
                    addCourseInstance(result,p,gbc);
                    courseframe.setVisible(false);
                    courseframe.setVisible(true);
                }
                catch (Exception e)
                {
                    e.printStackTrace();
                }
            }
        });
        courseframe.add(bEnroll);
        courseframe.add(scroll);
        courseframe.setVisible(true);
    }

    private void displayGrades()
    {
        JFrame gradeframe = new JFrame("Oceny");
        gradeframe.setLocationRelativeTo(null);
        gradeframe.setResizable(false);
        gradeframe.setLayout(null);

gradeframe.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
        gradeframe.setSize(600,160);

        JTextArea tGrades = new JTextArea();
        tGrades.setFont(new Font("Verdana",Font.PLAIN,30));
        tGrades.setMargin(new Insets(5,5,5,5));
        tGrades.setEditable(false);
        tGrades.setBackground(Color.LIGHT_GRAY);
        JScrollPane scGrades = new
JScrollPane(tGrades,ScrollPaneConstants.VERTICAL_SCROLLBAR_NEVER,ScrollPane
Constants.HORIZONTAL_SCROLLBAR_AS_NEEDED);
        scGrades.setBounds(30, 70,540,50);
        gradeframe.add(scGrades);

        JComboBox courseBox = new JComboBox();
        courseBox.setBounds(200,20,200,30);
        try
        {
            String query = "select C.Name "
                    + "from Courses C, Registry R, Users U "
```

```
                    + "where U.id = R.Student_id and R.Course_id = C.id and
U.id = ?";
            PreparedStatement ps = myConn.prepareStatement(query);
            ps.setInt(1,id);
            ResultSet rs = ps.executeQuery();
            int i=0;
            while(rs.next())
            {
                courseBox.addItem(rs.getString("Name"));
                if(i==0)
                {
                    String query2 = "select Value "
                            + "from Grades "
                            + "where Student_id = ? and Course_name = ?";
                    PreparedStatement ps2 =
myConn.prepareStatement(query2);
                    ps2.setInt(1,id);
                    ps2.setString(2,rs.getString("Name"));
                    ResultSet rs2 = ps2.executeQuery();
                    while(rs2.next())
                    {
                        tGrades.append(rs2.getDouble("Value")+" | ");
                    }
                }
                i+=1;
            }
            if(i!=0)
                courseBox.setSelectedIndex(0);
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }

        courseBox.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent actionEvent) {
                try
                {
                    tGrades.setText("");
                    String course = (String) courseBox.getSelectedItem();

                    String query = "select Value "
                            + "from Grades "
                            + "where Student_id = ? and Course_name = ?";
                    PreparedStatement ps = myConn.prepareStatement(query);
                    ps.setInt(1,id);
                    ps.setString(2,course);
                    ResultSet rs = ps.executeQuery();
```

```
                    while(rs.next())
                    {
                        tGrades.append(rs.getString("Value")+" | ");
                    }
                }
                catch (Exception e)
                {
                    e.printStackTrace();
                }
            }
        });
        gradeframe.add(courseBox);

        gradeframe.setVisible(true);
    }
}
```

LecturerWindow.java

```
import javax.swing.*;
import javax.swing.filechooser.FileSystemView;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileInputStream;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;

public class LecturerWindow extends UserWindow{
    private int counter;
    public LecturerWindow(Connection myConn)
    {
        super(myConn);
        mainframe.setTitle("Panel profesorski");

        bCourse.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent actionEvent) {
                createCourse();
            }
        });

        bGrades.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent actionEvent) {
                addGrades();
```

```
            }
        });
    }

    private void addCourseInstance(String name,JPanel p, GridBagConstraints
gbc)
    {
        JPanel pi = new JPanel();
        pi.setLayout(null);
        JLabel l = new JLabel(name);
        l.setBounds(20, 10, 200, 30);
        JButton b = new JButton("Pliki");
        b.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent actionEvent) {
                JFileChooser jfc = new
JFileChooser(FileSystemView.getFileSystemView().getHomeDirectory());

                int returnValue = jfc.showOpenDialog(null);

                if(returnValue==JFileChooser.APPROVE_OPTION)
                {
                    File selectedFile = jfc.getSelectedFile();
                    FileInputStream fis;
                    try {
                        fis = new FileInputStream(selectedFile);
                        String query = "insert into Materials "
                                + "(File, Course_id) "
                                + "select ?, id "
                                + "from Courses "
                                + "where Name = ?";
                        PreparedStatement ps =
myConn.prepareStatement(query);
                        ps.setBinaryStream(1,fis,(int)
selectedFile.length());
                        ps.setString(2,name);
                        ps.executeUpdate();
                    }
                    catch (Exception e)
                    {
                        e.printStackTrace();
                    }
                }
            }
        });
        b.setBounds(280, 10, 80, 30);
        pi.add(b);
        pi.add(l);
        gbc.gridx = 0;
```

```
        gbc.gridy = counter;
        gbc.ipadx = 380;
        gbc.ipady = 50;
        gbc.fill = GridBagConstraints.HORIZONTAL;
        p.add(pi, gbc);
        counter+=1;
    }

    private void createCourse()
    {
        JFrame courseframe = new JFrame("Kursy");
        courseframe.setLocationRelativeTo(null);

courseframe.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
        courseframe.setResizable(false);
        courseframe.setLayout(null);
        courseframe.setSize(500,600);

        JLabel lCour = new JLabel("Twoje Kursy");
        lCour.setBounds(50,30,400,50);
        lCour.setFont(new Font("Verdana",Font.BOLD,30));
        lCour.setHorizontalAlignment(SwingConstants.CENTER);
        courseframe.add(lCour);

        JPanel p = new JPanel();
        GridBagConstraints gbc = new GridBagConstraints();
        GridBagLayout gbl = new GridBagLayout();
        p.setLayout(gbl);
        counter=0;
        try {
            String query = "Select Name from Courses where Creator_id=?";
            PreparedStatement ps1 = myConn.prepareStatement(query);
            ps1.setInt(1,id);
            ResultSet rs = ps1.executeQuery();
            while(rs.next()) {
                addCourseInstance(rs.getString("Name"),p,gbc);
            }
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
        JScrollPane scroll = new
JScrollPane(p,JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED,JScrollPane.HORIZONT
AL_SCROLLBAR_AS_NEEDED);
        scroll.getVerticalScrollBar().setPreferredSize(new
Dimension(15,0));
        scroll.setBounds(50,100,400,400);
```

```
        JTextField tAdd = new JTextField();
        tAdd.setBounds(150,520,300,30);
        courseframe.add(tAdd);

        JButton bAdd = new JButton("Dodaj");
        bAdd.setBounds(50,520,80,30);
        bAdd.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent actionEvent) {
                if(!tAdd.getText().equals(""))
                {
                    String query = "insert into Courses "
                            +"(Name, Creator_id) "
                            +"Values(?,?)";
                    try {
                        PreparedStatement ps =
myConn.prepareStatement(query);
                        ps.setString(1,tAdd.getText());
                        ps.setInt(2,id);
                        ps.executeUpdate();
                        addCourseInstance(tAdd.getText(),p,gbc);
                        tAdd.setText("");
                        courseframe.setVisible(false);
                        courseframe.setVisible(true);
                    }
                    catch (Exception e)
                    {
                        e.printStackTrace();
                    }
                }
            }
        });
        courseframe.add(bAdd);

        courseframe.add(scroll);
        courseframe.setVisible(true);
    }

    private void addGrades()
    {
        JFrame gradeframe  = new JFrame("Oceny");
        gradeframe.setLayout(null);

gradeframe.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
        gradeframe.setResizable(false);
        gradeframe.setLocationRelativeTo(null);
        gradeframe.setSize(700,100);

        JComboBox courseBox = new JComboBox();
```

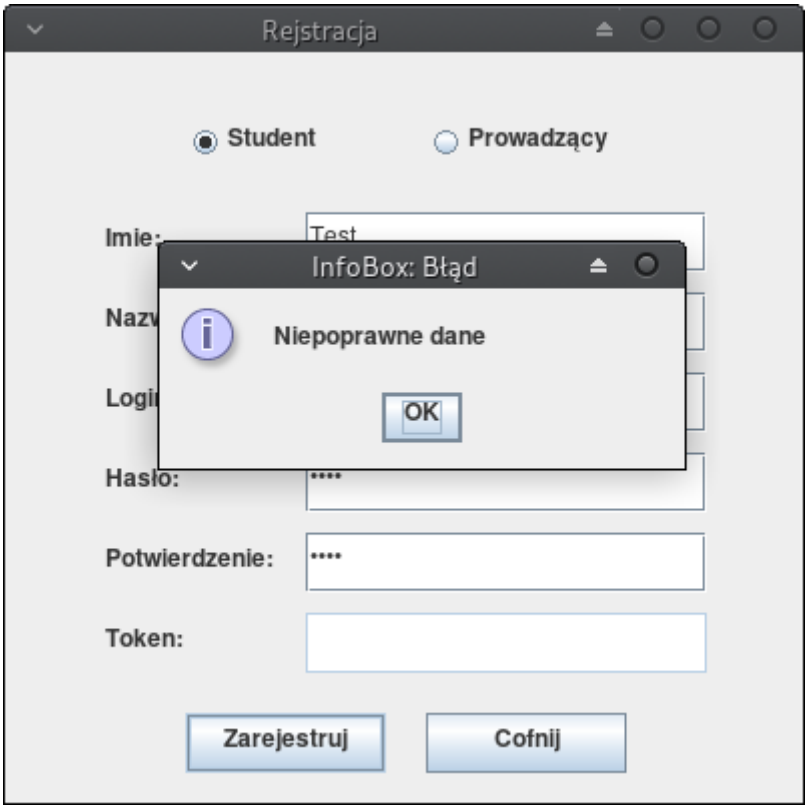
```
        courseBox.setBounds(270,20,230,30);
        gradeframe.add(courseBox);

        JComboBox studentBox  = new JComboBox();
        studentBox.setBounds(20,20,230,30);

        ArrayList<Integer> SID = new ArrayList<>();

        try
        {
            String query = "select U.id, U.Fullname "
                    + "from Users U, Registry R, Courses C "
                    + "where U.id = R.Student_id and R.Course_id = C.id and
C.Creator_id = ? "
                    + "group by U.Fullname";
            PreparedStatement ps = myConn.prepareStatement(query);
            ps.setInt(1,id);
            ResultSet rs = ps.executeQuery();
            int i=0;
            while(rs.next())
            {
                studentBox.addItem(rs.getString("Fullname"));
                SID.add(rs.getInt("id"));
                if(i==0)
                {
                    String query2 = "select C.Name "
                            + "from Courses C, Users U, Registry R "
                            + "where C.id = R.Course_id and R.Student_id =
U.id and U.Fullname = ?";
                    PreparedStatement ps2 =
myConn.prepareStatement(query2);
                    ps2.setString(1,rs.getString("Fullname"));
                    ResultSet rs2 = ps2.executeQuery();
                    while(rs2.next())
                    {
                        courseBox.addItem(rs2.getString("Name"));
                    }
                }
                i+=1;
            }
            if(i!=0)
                studentBox.setSelectedIndex(0);
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }

        studentBox.addActionListener(new ActionListener() {
```

```
            @Override
            public void actionPerformed(ActionEvent actionEvent) {
                try
                {
                    String query = "select C.Name "
                            + "from Courses C, Users U, Registry R "
                            + "where C.id = R.Course_id and R.Student_id =
U.id and U.Fullname = ?";
                    PreparedStatement ps = myConn.prepareStatement(query);
                    ps.setString(1,(String) studentBox.getSelectedItem());
                    ResultSet rs = ps.executeQuery();
                    courseBox.removeAllItems();
                    while(rs.next())
                    {
                        courseBox.addItem(rs.getString("Name"));
                    }
                }
                catch (Exception e)
                {
                    e.printStackTrace();
                }
            }
        });
        gradeframe.add(studentBox);

        JTextField tGrade = new JTextField("");
        tGrade.setBounds(520,20,40,30);
        gradeframe.add(tGrade);

        JButton bSubmit = new JButton("Wstaw");
        bSubmit.setBounds(580,20,100,30);
        bSubmit.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent actionEvent) {
                String grade = tGrade.getText();
                if(!grade.equals(""))
                {
                    try
                    {
                        Double d = Double.parseDouble(grade);
                        String query = "insert into Grades "
                                + "(Course_name, Value, Student_id) "
                                + "Values (?,?,?)";
                        PreparedStatement ps =
myConn.prepareStatement(query);
                        ps.setString(1,(String)
courseBox.getSelectedItem());
                        ps.setDouble(2,d);
```

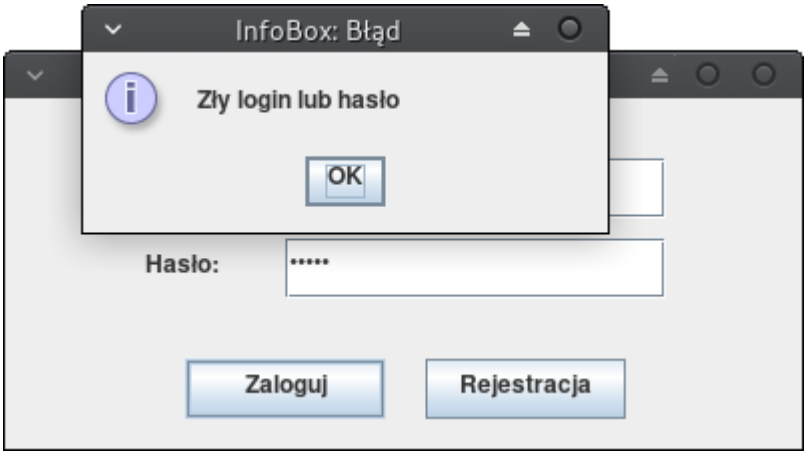
```
ps.setInt(3,SID.get(studentBox.getSelectedIndex()));
                        ps.executeUpdate();
                        tGrade.setText("");
                        JOptionPane.showMessageDialog(null, "Ocena dodana",
"InfoBox: " + "Sukces", JOptionPane.INFORMATION_MESSAGE);
                    }
                    catch (Exception e)
                    {
                        if(e.getClass()==NumberFormatException.class)
                            JOptionPane.showMessageDialog(null, "Ocena
powninna być liczbą…", "Error: " + "Błąd", JOptionPane.ERROR_MESSAGE);
                        else
                            e.printStackTrace();
                    }
                }
            }
        });
        gradeframe.add(bSubmit);

        gradeframe.setVisible(true);
    }
}
```

Testy Aplikacji

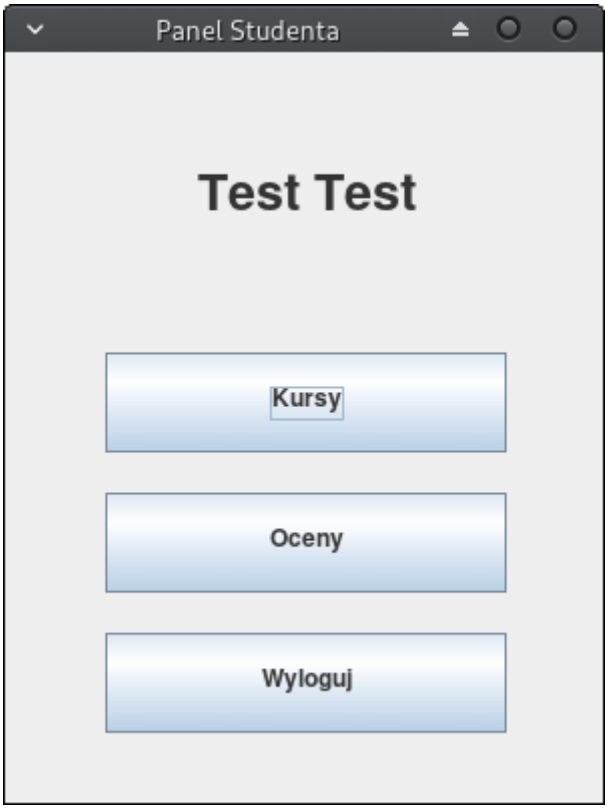
Nieudana Rejestracja



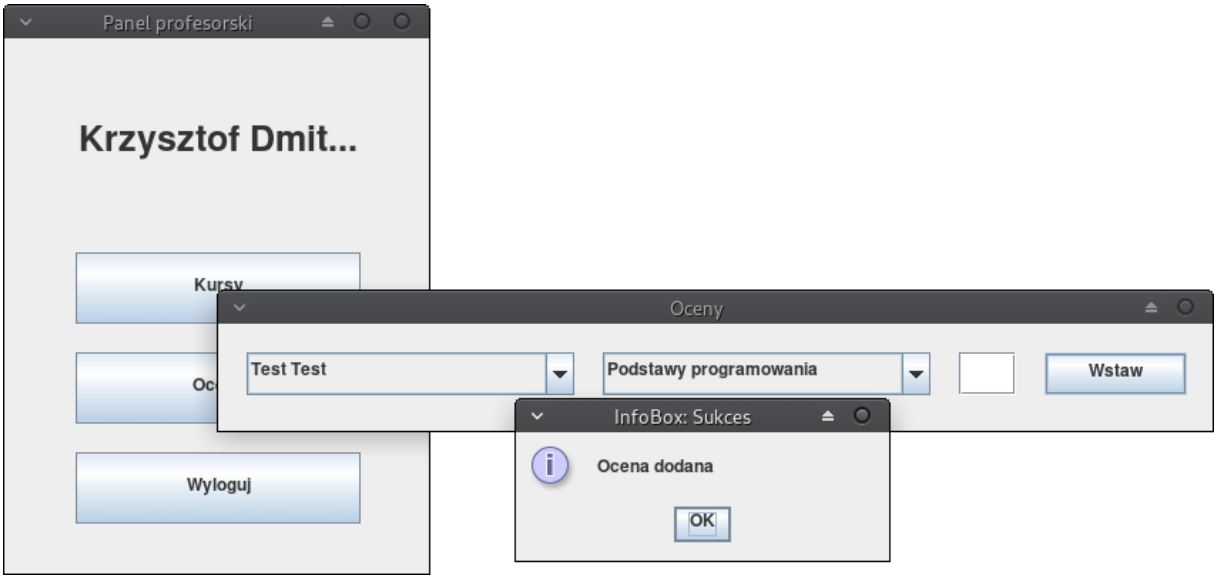
Nieudane Logowanie



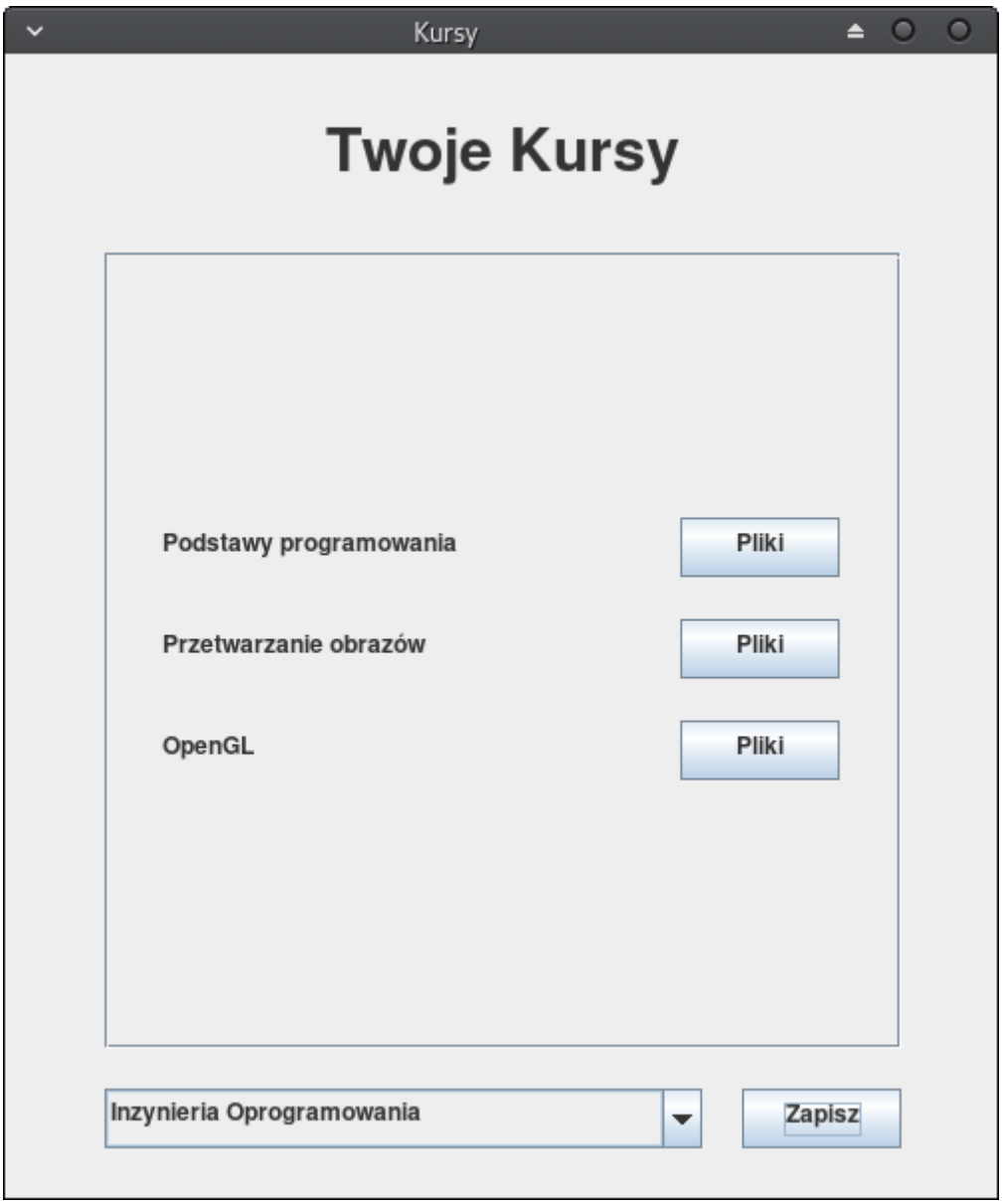
Udane logowanie – Panel użytkownika



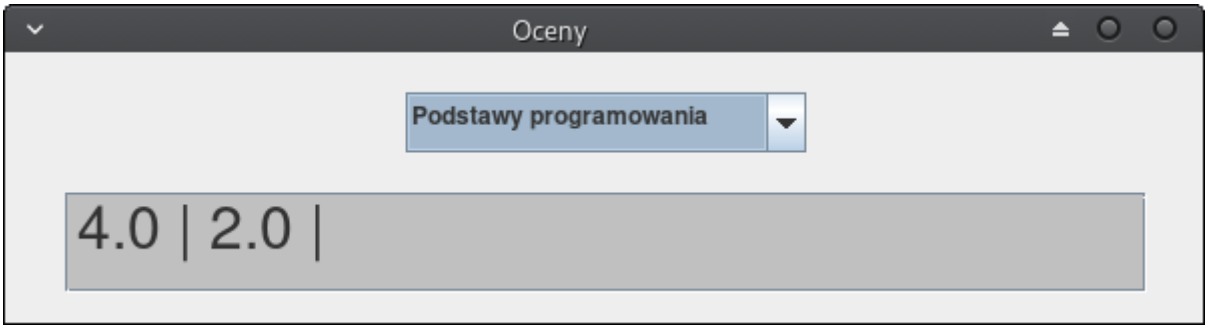
Udane dodanie oceny.



Udany zapis na kurs



Udane wyświetlenie ocen



Schemat bazy danych