Dokument: Projekt aplikacji - szablon v.2022-03-18

**Projekt aplikacji**

Software Architecture Design (SAD)

Spis treści

1. [Wstęp 2](#_TOC_250012)
2. [Cel i zakres dokumentu 2](#_TOC_250011)
3. [Definicja architektury aplikacji 2](#_TOC_250010)
4. [Cele i ograniczenia architektury 2](#_TOC_250009)
5. [Obraz logiczny aplikacji 2](#_TOC_250008)
   1. [Charakterystyka pakietów 2](#_TOC_250007)
   2. [Diagram klas aplikacji 2](#_TOC_250006)
   3. [Specyfikacja funkcji i metod aplikacji 2](#_TOC_250005)
6. [Dynamiczny obraz modelowanej aplikacji 2](#_TOC_250004)
   1. [Diagram sekwencji UML dla obiektów 2](#_TOC_250003)
   2. [Diagram aktywności UML dla obiektów 2](#_TOC_250002)
7. [Statyczny obraz modelowanej aplikacji 2](#_TOC_250001)
8. [Projekt bazy danych 2](#_TOC_250000)

# Wstęp

# Cel i zakres dokumentu

# Definicja architektury aplikacji

# Cele i ograniczenia architektury

# Obraz logiczny aplikacji

## Charakterystyka pakietów

## Ze względu na niską złożoność aplikacji nie ma podziału ma odrębne pakiety

## Diagram klas aplikacji

## Specyfikacja funkcji i metod aplikacji

btnEdytuj.setBackground(new Color(240, 230, 140));

btnEdytuj.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

int i = tabelaFilmow.getSelectedRow();

if (i >= 0) {

Connection con = *con*();

try {

int id = Integer.*parseInt*(model.getValueAt(i, 0).toString());

String query = "UPDATE kinoinformacje set Rezyser =? , Scenarzysta= ? ,Tytul=?,Gatunek=?, Ocena=?, Zdjecie=? where id=? ";

ArrayList<Filmy> list = new ArrayList<>();

PreparedStatement ps = con.prepareStatement(query);

ps.setString(1, textRezyser.getText());

ps.setString(2, textScenarzysta.getText());

ps.setString(3, textTytul.getText());

ps.setString(4, textGatunek.getText());

String cOcena = comboBoxOcena.getSelectedItem().toString();

ps.setString(5, cOcena);

ps.setBytes(6, zdjecieFilm);

ps.setInt(7, id);

ps.execute();

ps.close();

con.close();

} catch (Exception m) {

System.***out***.print("Bledna aktualizacja " + m);

}

model.setValueAt(textRezyser.getText(), i, 1);

model.setValueAt(textScenarzysta.getText(), i, 2);

model.setValueAt(textTytul.getText(), i, 3);

model.setValueAt(textGatunek.getText(), i, 4);

model.setValueAt(comboBoxOcena.getSelectedItem().toString(), i, 5);

JOptionPane.*showMessageDialog*(null, "Edytowano film na pozycji:" + (i + 1));

} else {

JOptionPane.*showMessageDialog*(null, "Aby edytować film najpierw wybierz pozycję!");

}

}

});

btnUsun.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

int i = tabelaFilmow.getSelectedRow();

if (i >= 0) {

Connection con = *con*();

try {

int id = Integer.*parseInt*(model.getValueAt(i, 0).toString());

String query = "DELETE FROM kinoinformacje where id=? ";

ArrayList<Filmy> list = new ArrayList<>();

PreparedStatement ps = con.prepareStatement(query);

ps.setInt(1, id);

ps.execute();

ps.close();

con.close();

} catch (Exception a) {

System.***out***.print("Blad w usuwaniu");

}

model.removeRow(i);

odswiezTabele();

JOptionPane.*showMessageDialog*(null, "Usunięto film na pozycji:" + (i + 1));

btnRezerwacja.setEnabled(false);

} else {

JOptionPane.*showMessageDialog*(null, "Aby usunąć film najpierw wybierz pozycję!");

btnRezerwacja.setEnabled(false);

}

}

});

btnDodaj.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (textRezyser.getText().equals("") || textScenarzysta.getText().equals("")

|| textTytul.getText().equals("") || textGatunek.getText().equals("")) {

JOptionPane.*showMessageDialog*(null, "Wprowadzone dane do filmu są niepoprawne!");

} else {

Connection con = *con*();

row[0] = textID1.getText();

row[1] = textRezyser.getText();

row[2] = textScenarzysta.getText();

row[3] = textTytul.getText();

row[4] = textGatunek.getText();

row[5] = comboBoxOcena.getSelectedItem().toString();

try {

String dodaj = "INSERT INTO javakinodb.kinoinformacje (ID,Rezyser,Scenarzysta,Tytul,Gatunek,Ocena,Zdjecie) VALUES (?,?,?,?,?,?,?)";

PreparedStatement pa = con.prepareStatement(dodaj);

ArrayList<Filmy> list = new ArrayList<>();

int j = list.size() + 1;

System.***out***.println(j);

pa.setInt(1, j);

pa.setString(2, textRezyser.getText());

pa.setString(3, textScenarzysta.getText());

pa.setString(4, textTytul.getText());

pa.setString(5, textGatunek.getText());

String cOcena = comboBoxOcena.getSelectedItem().toString();

pa.setString(6, cOcena);

pa.setBytes(7, zdjecieFilm);

pa.execute();

} catch (Exception l) {

l.printStackTrace();

}

model.addRow(row);

int j = list.size() + 1;

textRezyser.setText("");

textScenarzysta.setText("");

textTytul.setText("");

textGatunek.setText("");

comboBoxOcena.setSelectedItem(null);

lblZdjecie.setText("");

odswiezTabele();

btnRezerwacja.setEnabled(false);

JOptionPane.*showMessageDialog*(null, "Wprowadzono nowy film!");

}

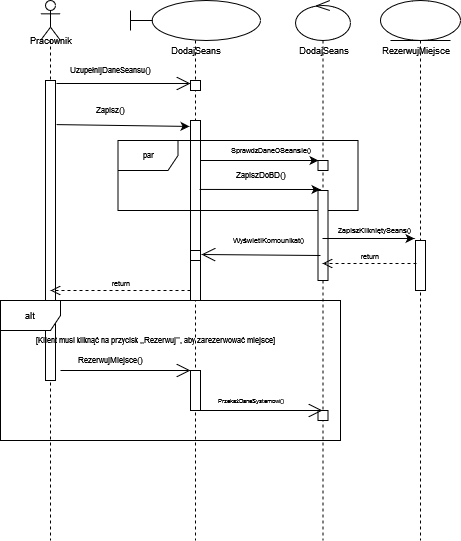
}

});

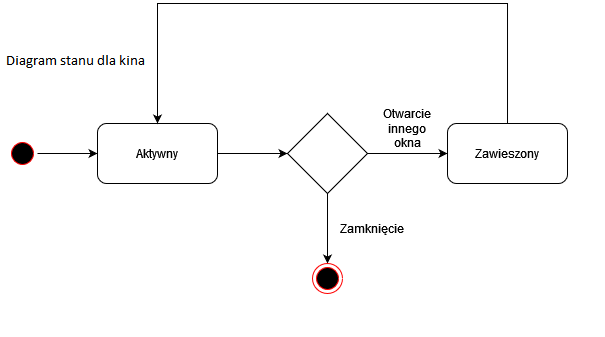
}

# Dynamiczny obraz modelowanej aplikacji

## Diagram sekwencji UML dla obiektów



## Diagram aktywności UML dla obiektów

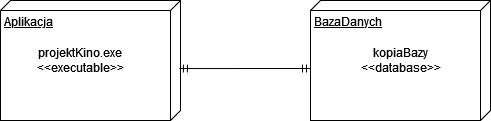


# Statyczny obraz modelowanej aplikacji

### Diagram komponentów UML

### 

### Diagram instalacji UML



# Projekt bazy danych



Wybrany jeden seans (kinoinformacje) może mieć jedną lub więcej rezerwacji miejsc (miejscainformacje).