

Uniwersytet Rzeszowski  
Wydział Nauk Ścisłych i Technicznych

Michał Rajzer  
134966  
Informatyka I rok



***“FrameKeeper” – Aplikacja do zarządzania kolekcją filmów oraz  
użytkownikami aplikacji***

Dokumentacja Techniczna

Projekt zaliczeniowy przedmiotu Programowanie Obiektowe

Prowadzący: dr. inż. Wojciech Koziół

## 1. Wstęp

“FrameKeeper” jest aplikacją do zarządzania kolekcją filmów oraz jej użytkownikami pozwalającą wypożyczać filmy.

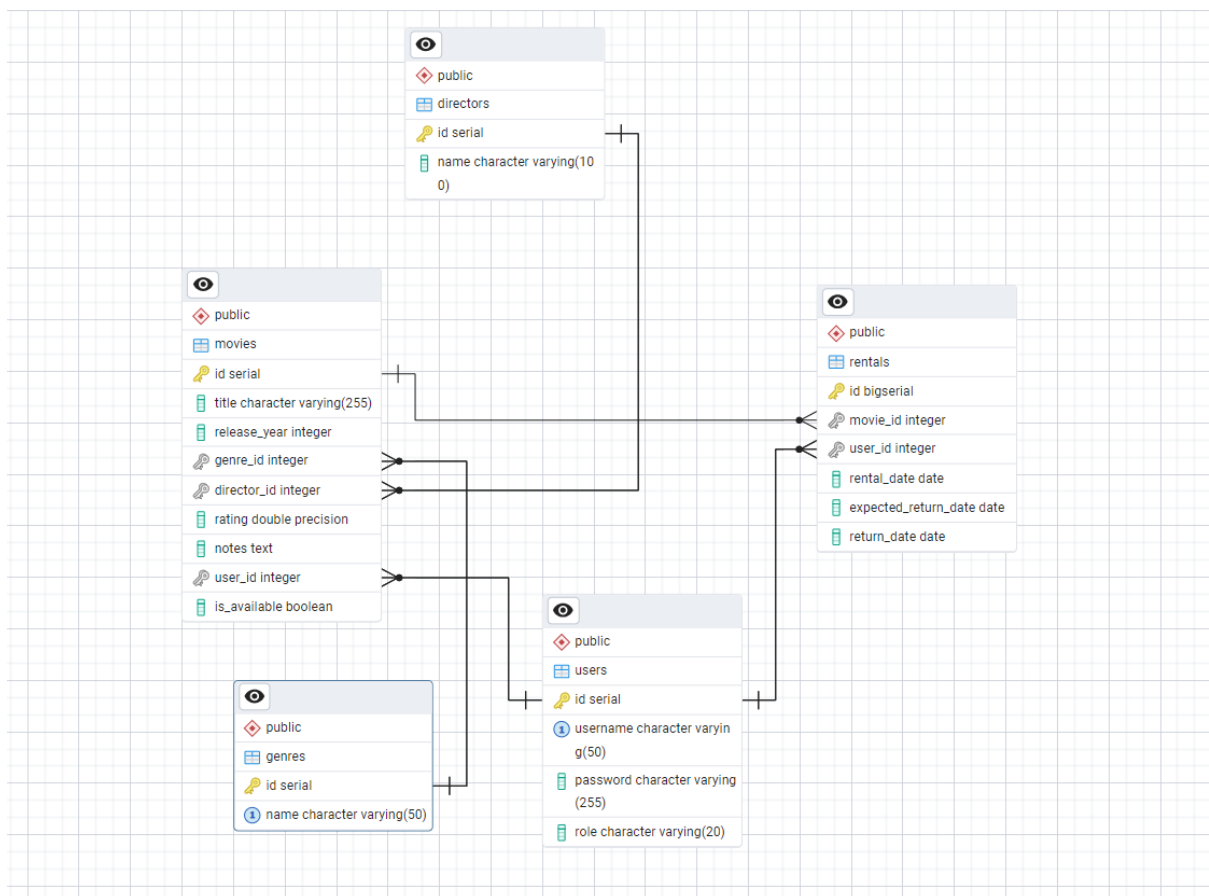
## 1.1 Wybrane Technologie

Baza danych: PostgreSQL

Biblioteka: JavaFX

Framework: Hibernate

## 1.2 Diagram ERD bazy danych



Rysunek 1: Rysunek przedstawiający połączenia bazy danych "Framekeeper"

## 1.3 Relację między tabelami

- Users [1 : n] Movies

Jeden użytkownik może dodać wiele filmów.

- Users [1 : n] Rentals

Jeden użytkownik może wypożyczyć wiele filmów.

- Directors [1 : n] Movies

Jeden reżyser może być przypisany do wielu filmów.

- Genres [1 : n] Movies

Jeden gatunek może być przypisany do wielu filmów.

- Movies [1 : n] Rentals

Jeden film może być wypożyczony wiele razy.

## 1.4 Połączenie z bazą danych

```
public class HibernateUtil {  @ Michal Rajzer *
    private static final Logger logger = LoggerFactory.getLogger(HibernateUtil.class);  4 usages
    private static StandardServiceRegistry registry;  6 usages
    private static SessionFactory sessionFactory;  4 usages

    public static SessionFactory getSessionFactory() {  20 usages  @ Michal Rajzer
        if (sessionFactory == null) {
            try {
                registry = new StandardServiceRegistryBuilder()
                    .configure()
                    .build();

                MetadataSources sources = new MetadataSources(registry);

                sources.addAnnotatedClass(Movie.class);
                sources.addAnnotatedClass(User.class);
                sources.addAnnotatedClass(Genre.class);
                sources.addAnnotatedClass(Director.class);
                sources.addAnnotatedClass(Rental.class);

                Metadata metadata = sources.getMetadataBuilder().build();

                sessionFactory = metadata.getSessionFactoryBuilder().build();

                logger.info("Hibernate SessionFactory została pomyślnie zainicjalizowana.");
            } catch (Exception e) {
                logger.error("Błąd podczas tworzenia SessionFactory: " + e.getMessage(), e);
                if (registry != null) {
                    StandardServiceRegistryBuilder.destroy(registry);
                    logger.warn("Rejestr usług Hibernate został zniszczony z powodu błędu.");
                }
                throw new RuntimeException("Błąd konfiguracji Hibernate: " + e.getMessage(), e);
            }
        }
    }
}
```

Rysunek 2: Rysunek przedstawiający połączenia aplikacji z bazą danych

```
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <property name="hibernate.connection.driver_class">org.postgresql.Driver</property>
        <property name="hibernate.connection.url">jdbc:postgresql://localhost:5432/CF</property>
        <property name="hibernate.connection.username">postgres</property>
        <property name="hibernate.connection.password">4469</property>

        <property name="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</property>
        <property name="hibernate.hbm2ddl.auto">update</property>
        <property name="hibernate.show_sql">true</property>
        <property name="hibernate.format_sql">true</property>

        <property name="hibernate.c3p0.min_size">5</property>
        <property name="hibernate.c3p0.max_size">20</property>
        <property name="hibernate.c3p0.timeout">300</property>
        <property name="hibernate.c3p0.max_statements">50</property>
        <property name="hibernate.c3p0.idle_test_period">3000</property>

    </session-factory>
</hibernate-configuration>
```

Rysunek 3: Rysunek przedstawiający plik konfiguracyjny hibernate.cfg.xml

```

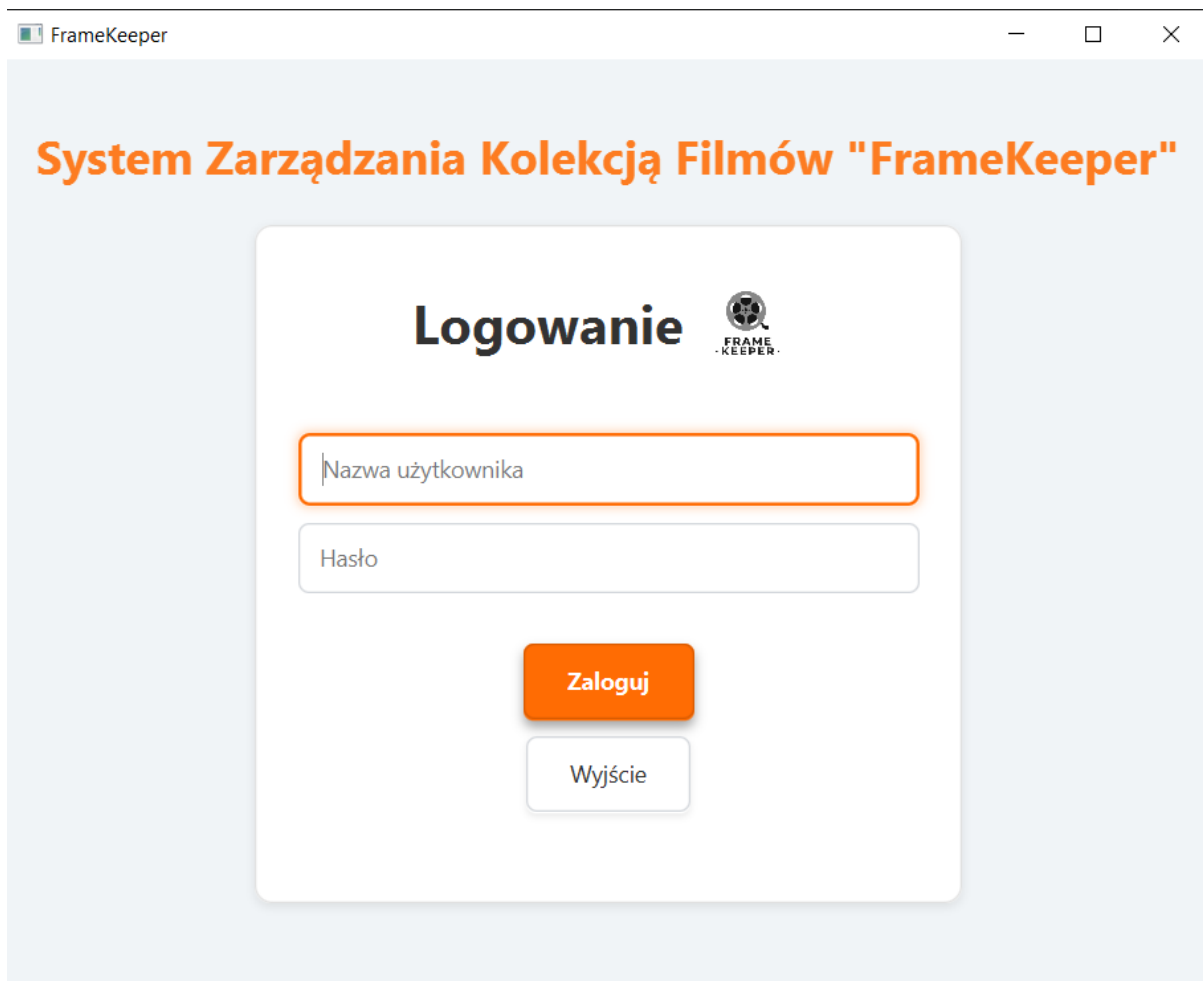
1 package com.example.projekt;
2
3 import jakarta.persistence.*;
4
5 import java.util.Objects;
6 import java.util.Set;
7
8 @Entity @Michał Rajzer *
9 @Table(name = "users")
10 public class User {
11
12     @Id
13     @GeneratedValue(strategy = GenerationType.IDENTITY)
14     private Integer id;
15
16     @Column(unique = true, nullable = false) 6 usages
17     private String username;
18
19     @Column(name = "password", nullable = false) 4 usages
20     private String passwordHash;
21
22     @Column(name = "role", nullable = false, length = 20) 3 usages
23     private String role;
24
25     @OneToMany(mappedBy = "user", cascade = CascadeType.ALL, fetch = FetchType.LAZY) 2 usages
26     private Set<Movie> movies;
27
28
29     public User() { @Michał Rajzer
30
31         this.role = "user";
32     }
33

```

*Rysunek 4: Tabele z bazy danych mają swoje odniesienie  
w kodzie dla przykładu tabela User*

## 1.4 Działanie aplikacji

- Logowanie do aplikacji “FrameKeeper”




Rysunek 5: okno logowania do aplikacji

FrameKeeper

System Zarządzania Kolekcją Filmów "FrameKeeper"

Logowanie



admin

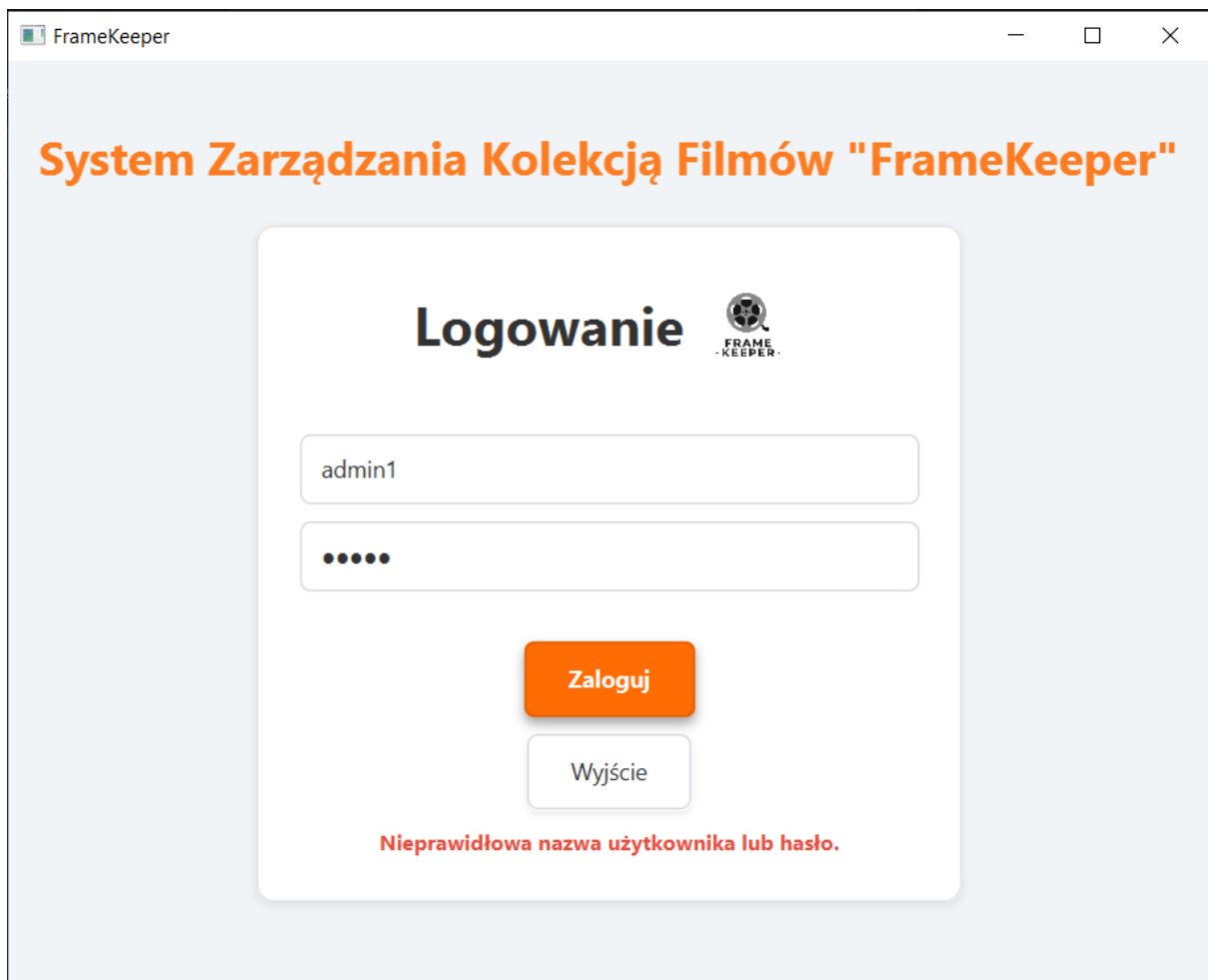
Hasło

Zaloguj

Wyjście

Proszę podać nazwę użytkownika i hasło.

Rysunek 6: Niepoprawne logowanie



Rysunek 7: Próba logowania do nieistniejącego użytkownika



```

private boolean authenticate(String username, String plainPassword) {
    Session session = null;
    Transaction tx = null;

    try {
        session = HibernateUtil.getSessionFactory().openSession();
        tx = session.beginTransaction();

        Query<User> query = session.createQuery(
            queryString: "SELECT u FROM User u WHERE u.username = :username", User.class
        );
        query.setParameter("username", username);

        User user = query.uniqueResult();

        tx.commit();

        if (user != null && user.checkPassword(plainPassword)) {
            this.currentUser = user;
            errorLabel.setText("");
            return true;
        } else {
            errorLabel.setText("Nieprawidłowa nazwa użytkownika lub hasło.");
            this.currentUser = null;
            return false;
        }
    } catch (Exception e) {
        if (tx != null && tx.isActive()) {
            tx.rollback();
        }
        System.err.println("Błąd autentykacji: " + e.getMessage());
        e.printStackTrace();
        errorLabel.setText("Błąd bazy danych podczas logowania.");
    }
}

```

Rysunek 8: Metoda authenticate otwiera nową sesję w Hibernate jeśli podany użytkownik istnieje wysyła zapytanie do bazy danych jeśli użytkownik nie istnieje wyrzuci błąd

```

private void handleLogin() {
    String username = usernameField.getText();
    String password = passwordField.getText();

    if (username.isEmpty() || password.isEmpty()) {
        errorLabel.setText("Proszę podać nazwę użytkownika i hasło.");
        return;
    }

    if (authenticate(username, password)) {
        try {
            ResourceBundle bundle = ResourceBundle.getBundle("messages", Locale.forLanguageTag("pl"));
            FXMLLoader loader;
            String fxmlPath;
            String appTitleKey;
            Parent rootNode;

            if (currentUser.getRole() != null && currentUser.getRole().equalsIgnoreCase("ADMIN")) {
                fxmlPath = "/com/example/projekt/hello-view.fxml";
                loader = new FXMLLoader(getClass().getResource(fxmlPath), bundle);
                rootNode = loader.load();
                MoviesController moviesController = loader.getController();
                moviesController.setCurrentUser(currentUser);
                appTitleKey = "app.title.admin";
            } else if (currentUser.getRole() != null && currentUser.getRole().equalsIgnoreCase("USER")) {
                fxmlPath = "/com/example/projekt/client-view.fxml";
                loader = new FXMLLoader(getClass().getResource(fxmlPath), bundle);
                rootNode = loader.load();
                ClientController clientController = loader.getController();
                clientController.setCurrentUser(currentUser);
                appTitleKey = "app.title.client";
            } else {

```

Rysunek 9: Metoda handlelogin jest wywoływane po kliknięciu przycisku "Zaloguj"

Sprawdza czy pola textfield są puste jeśli tak wyrzuca błąd natomiast jeżeli pola są wypełnione wywołuje metodę authenticate

```
@FXML @Michał Rajzer
private void handleExit() {
    Platform.exit();
    System.exit( status: 0);
}
```

Rysunek 10: Metoda handleExit po użyciu przycisku “Wyjście” zapewnia całkowite zamknięcie aplikacji

FrameKeeper

Zalogowano: admin Wyloguj

Tytuł	Rok	Gatunek	Reżyser	Ocena	Opis	Dostępność	Wypożyczony przez
Avengers: Wojna bez granic	2018	Science Fiction	Anthony Rus...	8.1	Potężny Thanos ...	Wypożycz...	Kamyl
Coco	2017	Animacja	Lee Unkrich, ...	8.3	Dwunastoletni ...	Wypożycz...	_Kowal
Czego nauczyła mnie ośm...	2020	Dokumentalny	Pippa Ehrlich...	8.1	Niezwykła opow...	Dostępny	-
Dzień Świra	2002	Komedia	Marek Koterski	7.8	Kultowa polska ...	Wypożycz...	Kamyl
Dziki robot	2024	Animacja	Chris Sanders	8.1	Roz, robot, rozbi...	Wypożycz...	NowaK
Forrest Gump	1994	Dramat	Quentin Tara...	8.8	Historia życia ni...	Wypożycz...	test
Gladiator	2000	Dramat	Ridley Scott	4.8	Epicki film histor...	Dostępny	-
Gwiezdne wojny: Część V ...	1980	Science Fiction	Irvin Kershner	8.1	Imperator rozka...	Dostępny	-
Incepcja	2010	Science Fiction	Christopher ...	9.5	Świetny film o s...	Dostępny	-
John Wick	2014	Akcja	Chad Stahels...	7.7	Stylowa akcja i c...	Wypożycz...	_Kowal
Kac Vegas	2009	Komedia	Todd Phillips	7.9	Szalona noc w L...	Wypożycz...	test
Król Lew	1994	Animacja	Roger Allers, ...	8.5	Klasyczna opowi...	Dostępny	-

Filtruj: Wyszukaj po tytule...

Gatunek:

Tytuł: np. Harry Potter

Rok: np. 2024

Gatunek:

Reżyser: Wybierz lub wpisz nowego

Ocena: 0.0 - 10.0

Opis: Dodaj opis filmu...

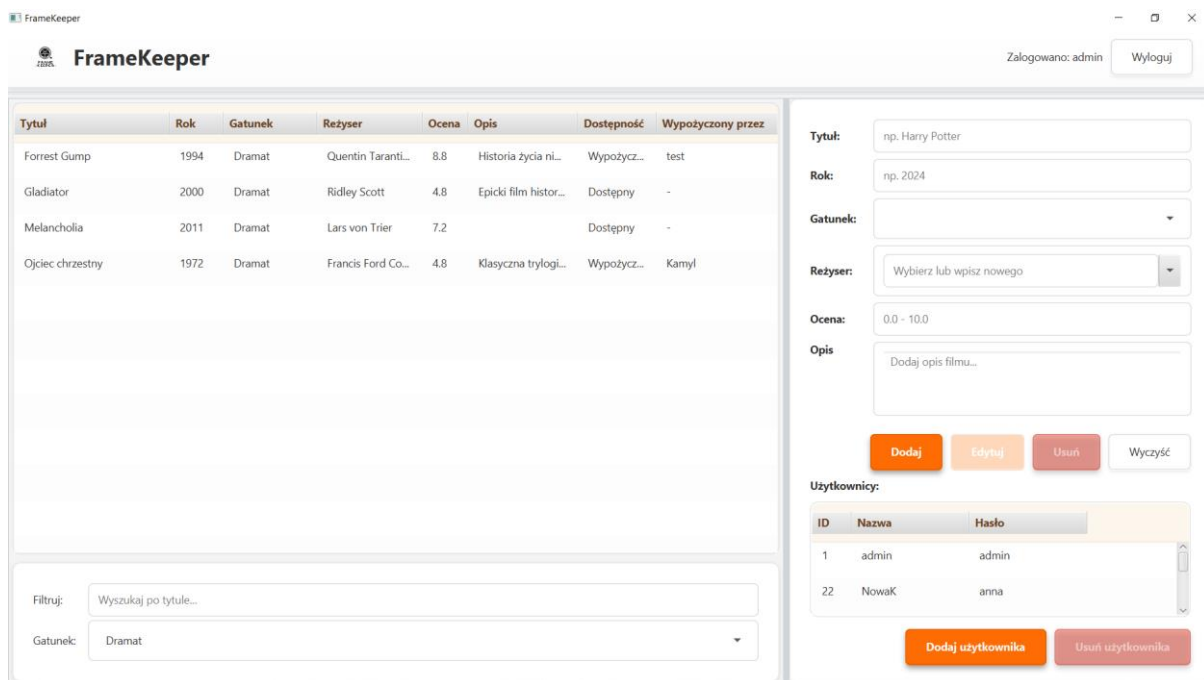
**Dodaj** **Edytuj** **Usuń** **Wyczyść**

Użytkownicy:

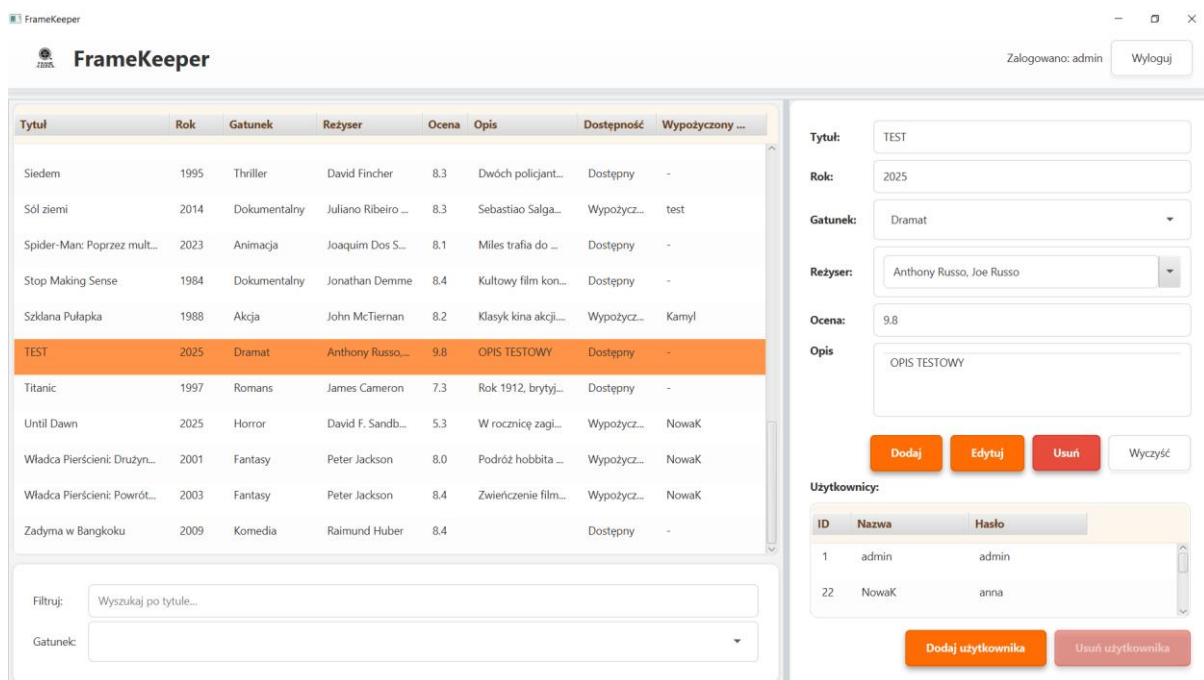
ID	Nazwa	Hasło
1	admin	admin
22	NowaK	anna

**Dodaj użytkownika** **Usuń użytkownika**

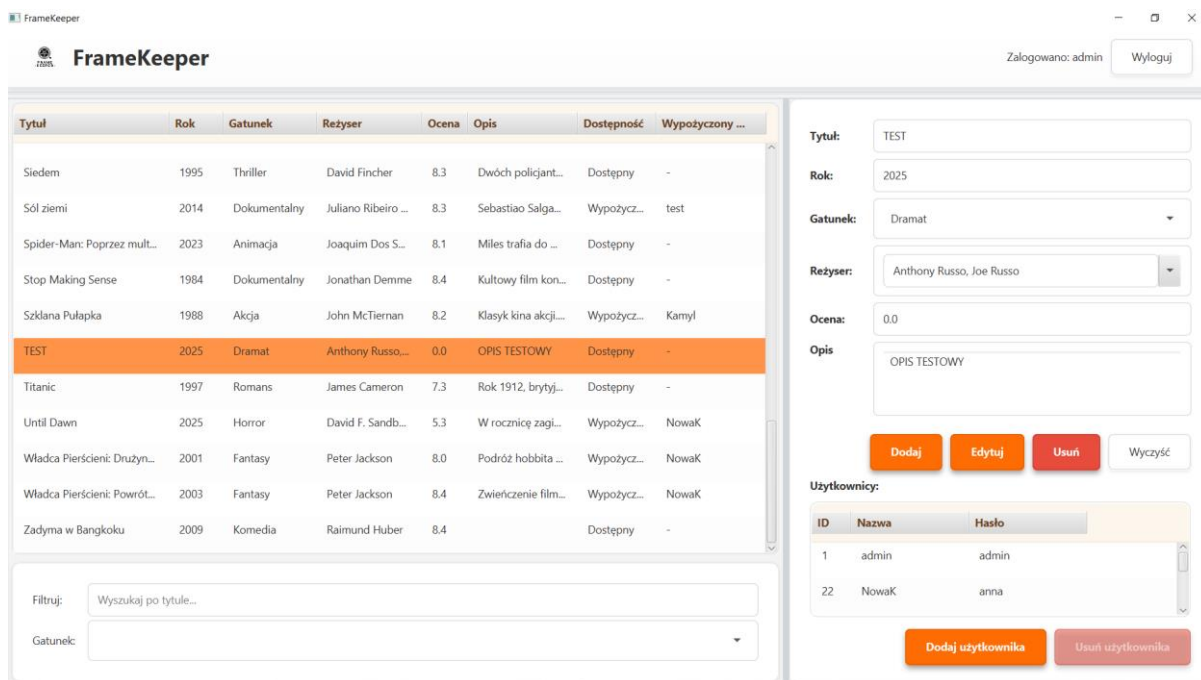
Rysunek 11: Po poprawnym zalogowaniu się w tym przypadku do konta "admin" ukazują się nam strona administratora aplikacji "FrameKeeper"



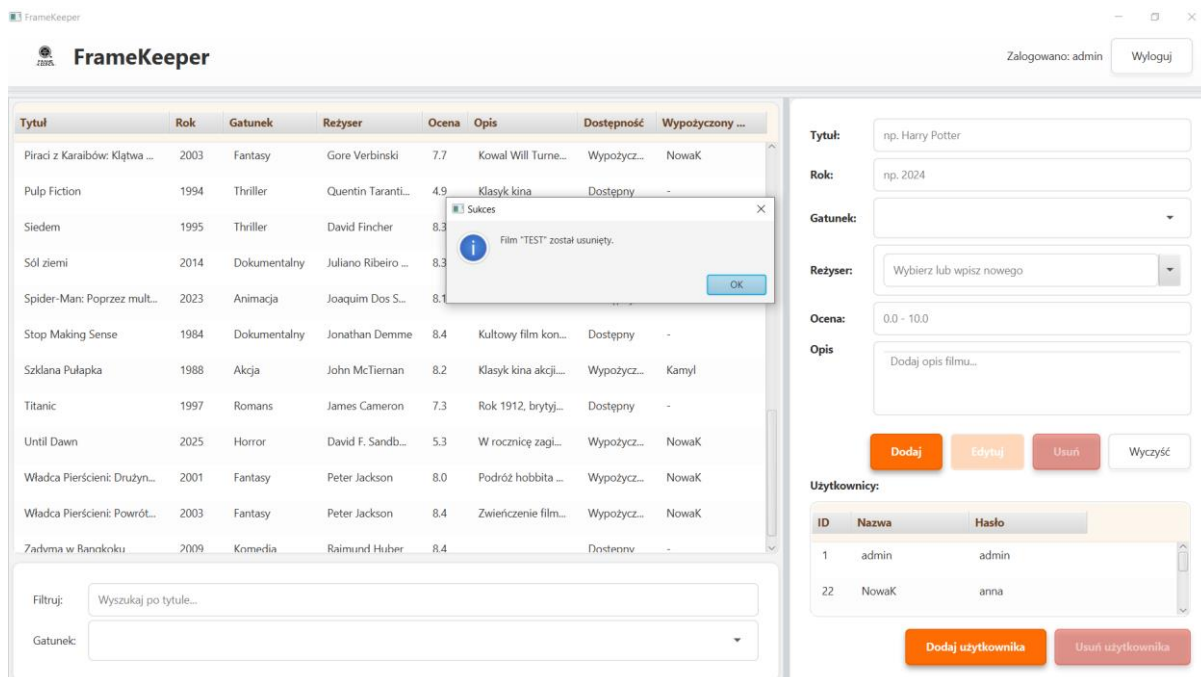
Rysunek 12: Filmy w aplikacji mogą być wyszukiwane po nazwie lub po gatunku filmu



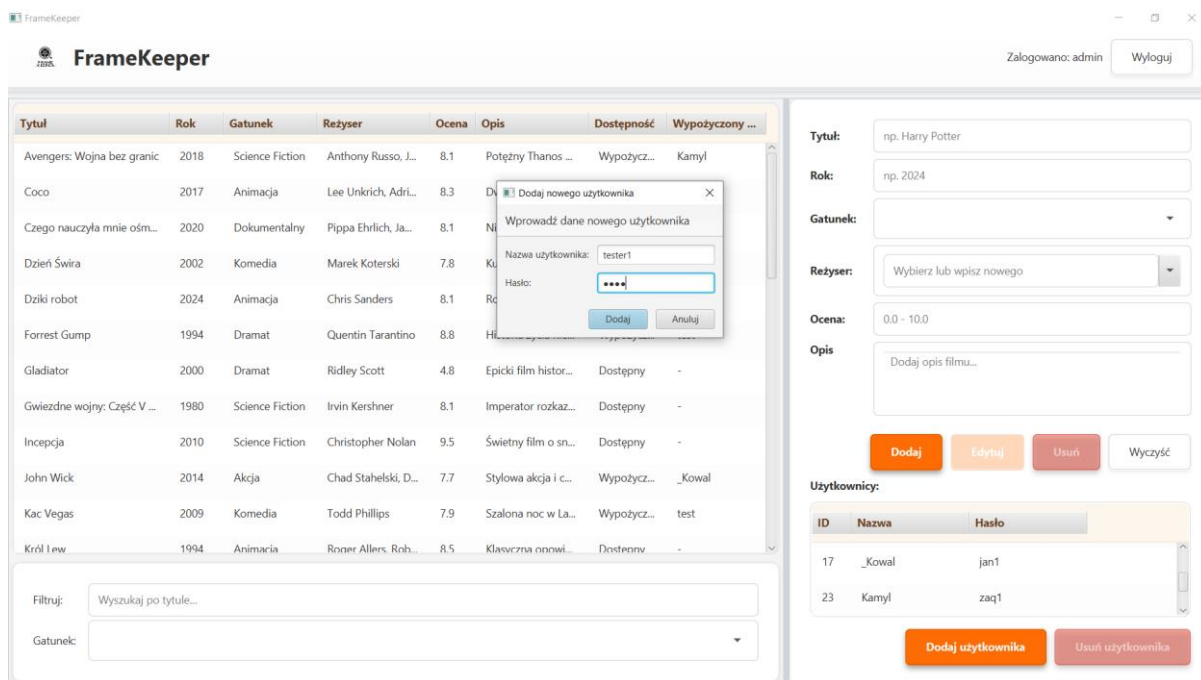
Rysunek 13: Administrator ma możliwość dodawania filmów do bazy danych



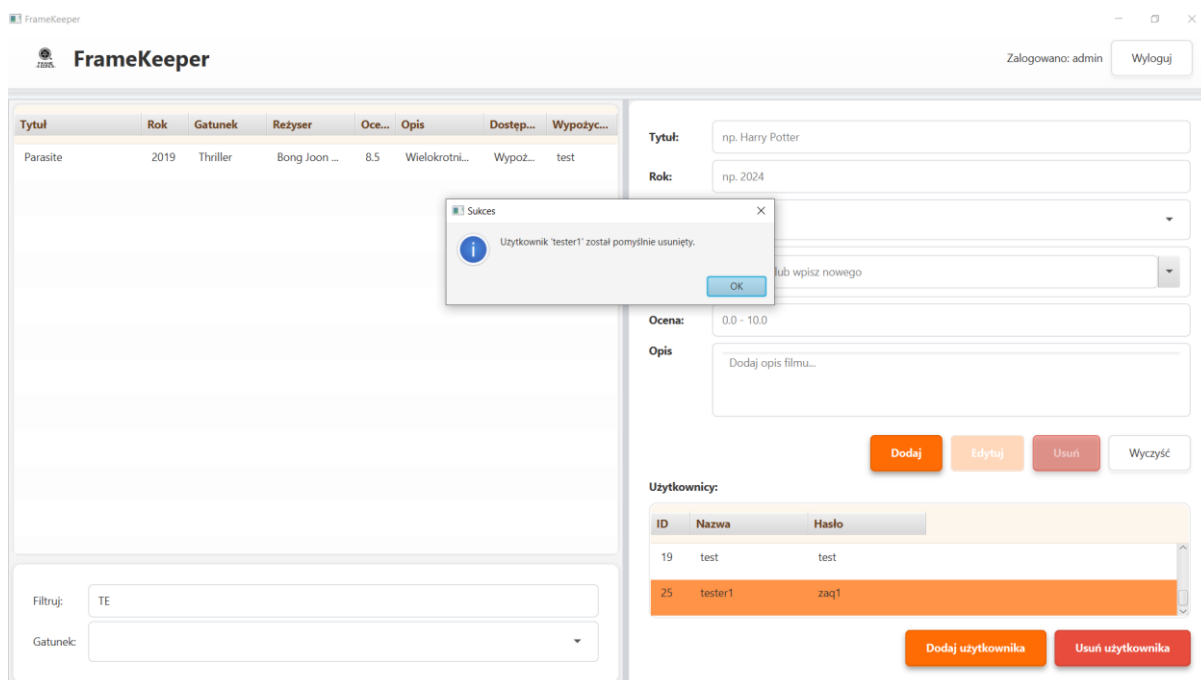
Rysunek 14: Administrator może edytować dodane filmy



Rysunek 15: Administrator może zarówno usuwać filmy z bazy danych

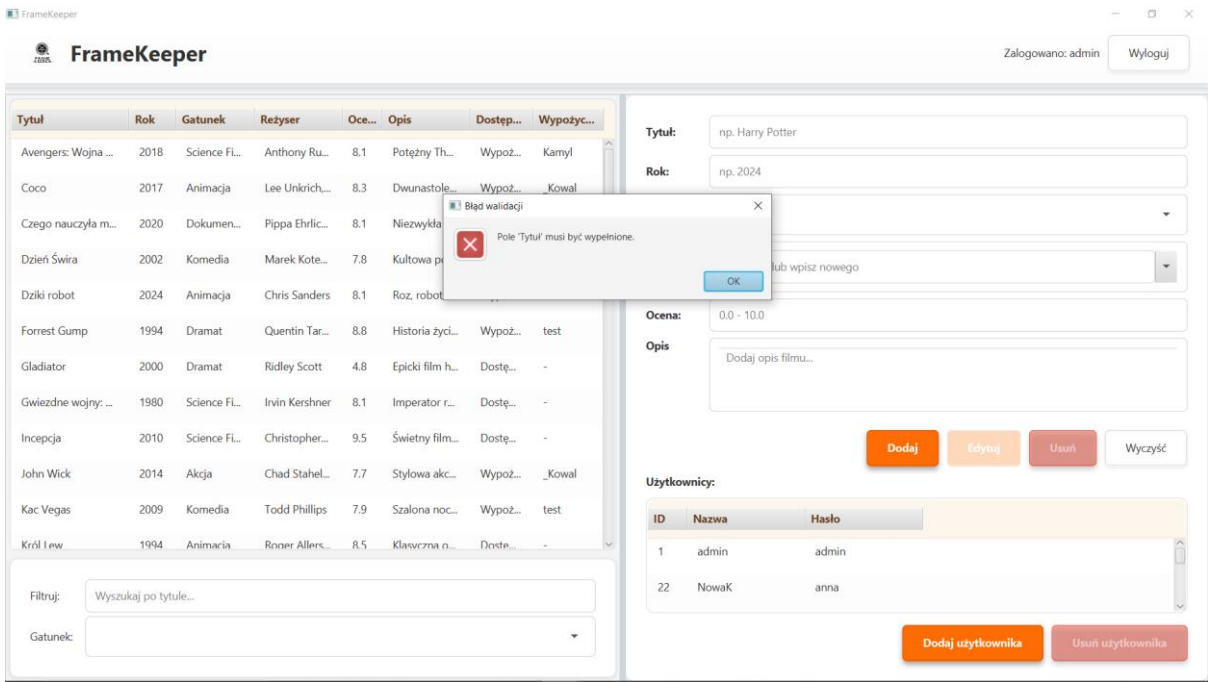


Rysunek 16: Konto admin może dodawać nowych użytkowników

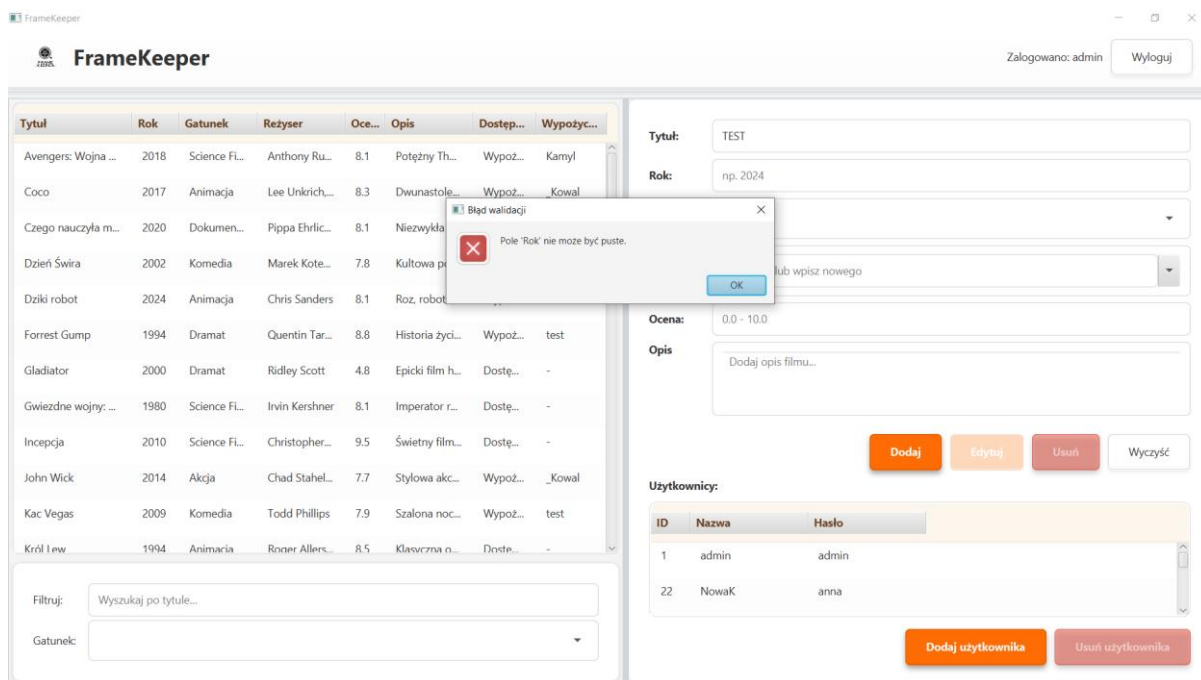


Rysunek 17: Usuwanie użytkownika

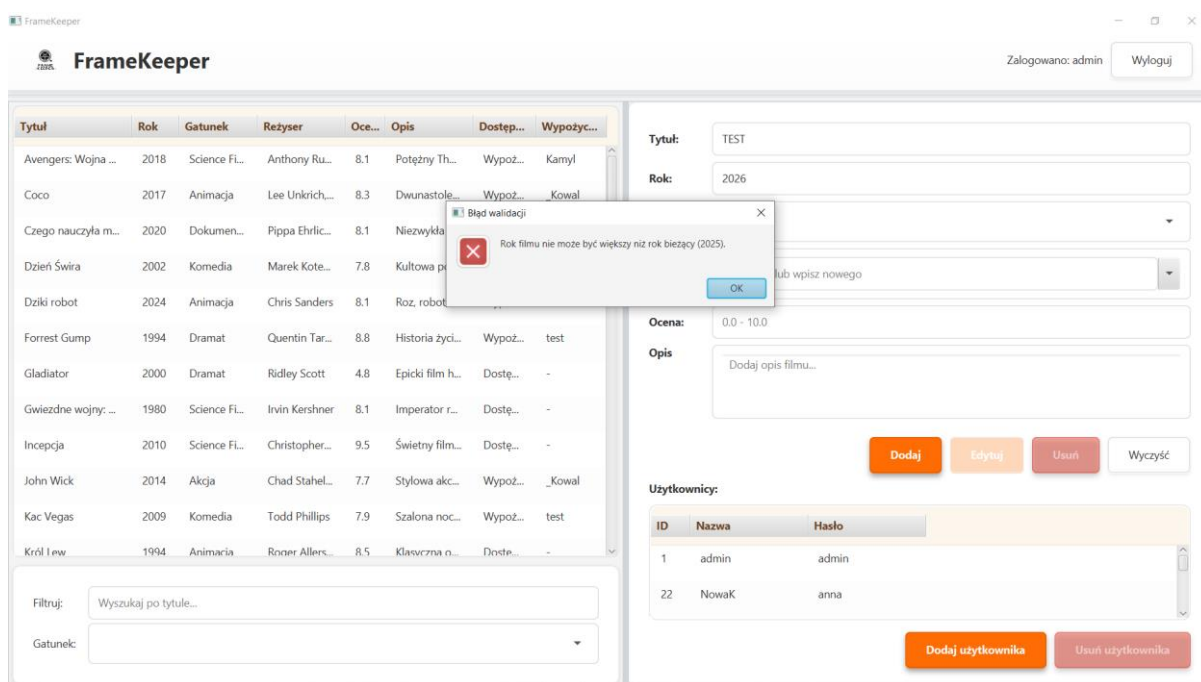
### 1.5 Obsługa błędów w koncie “admin”



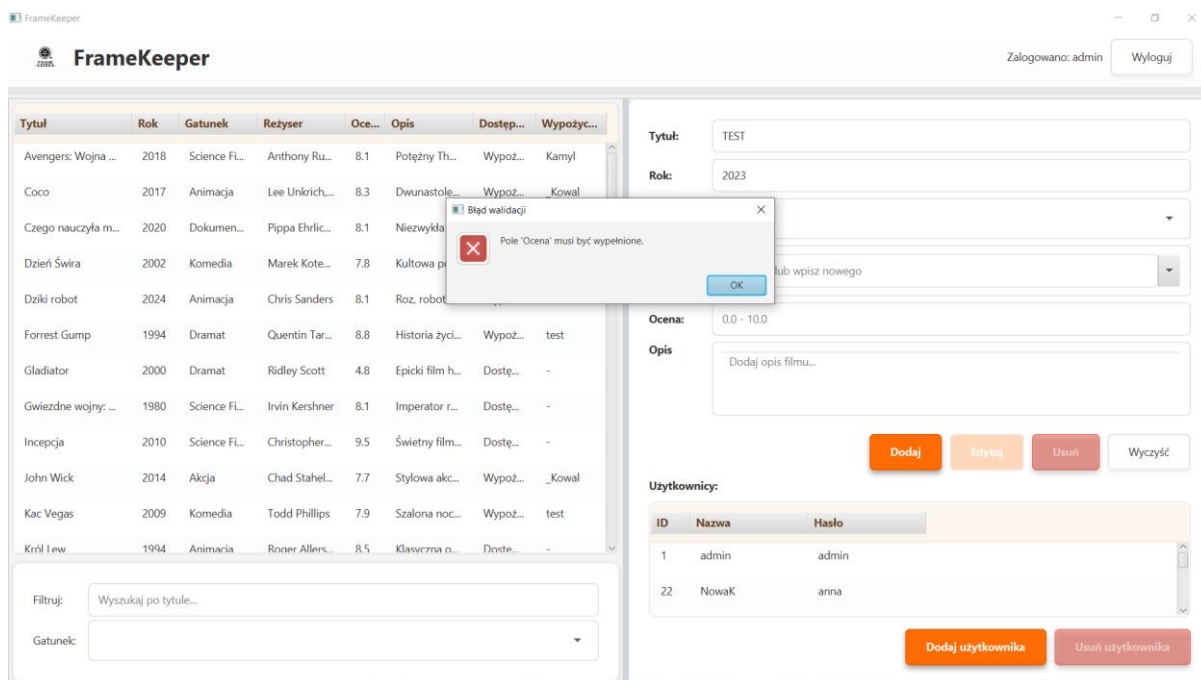
Rysunek 18: Pole tytuł nie może być puste



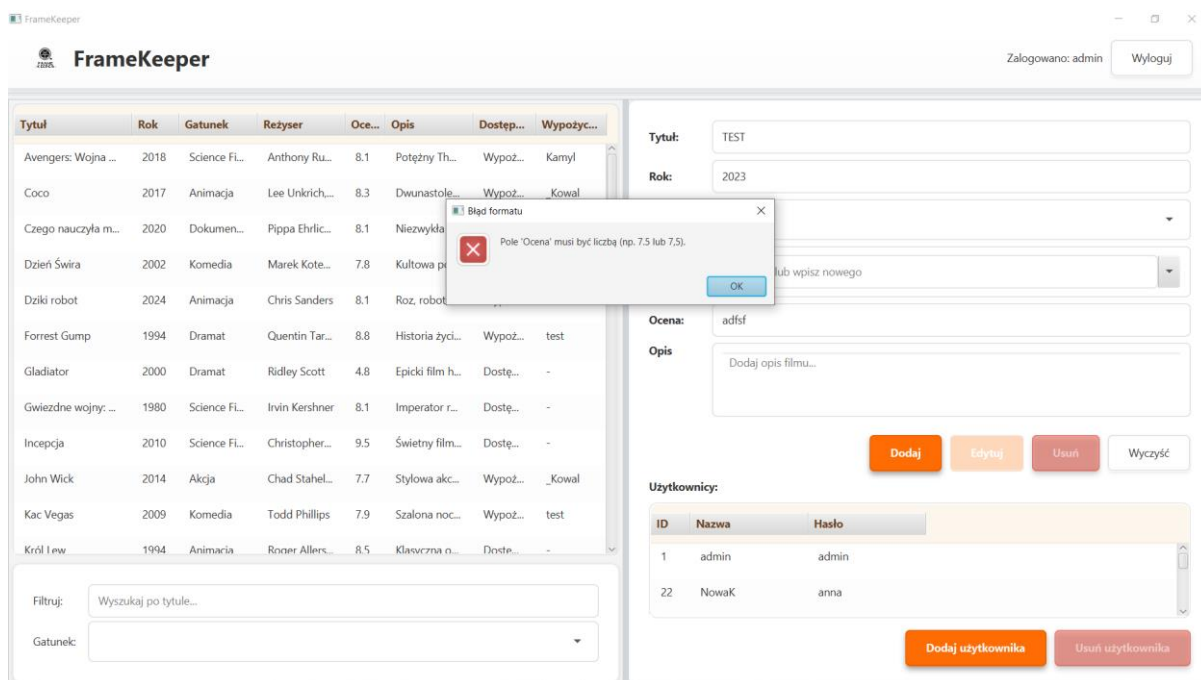
Rysunek 19: Pole rok nie może być puste



Rysunek 20: Pole rok nie może być większe niż rok bieżący

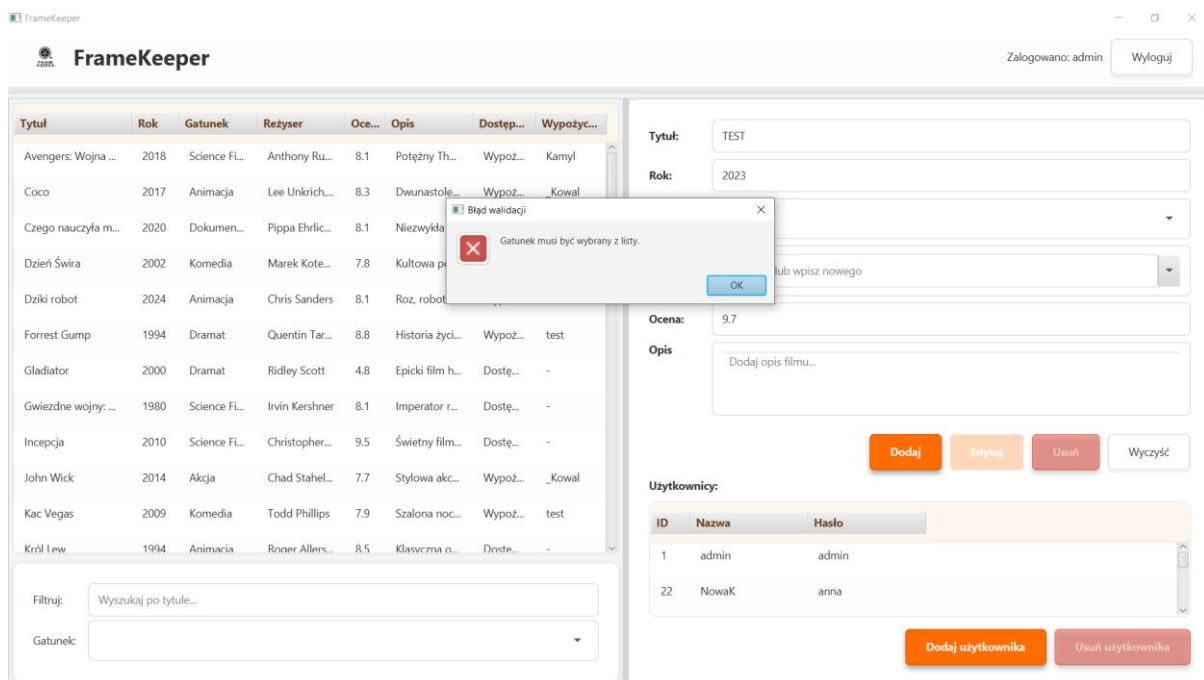


Rysunek 21: Pole ocena nie może być puste

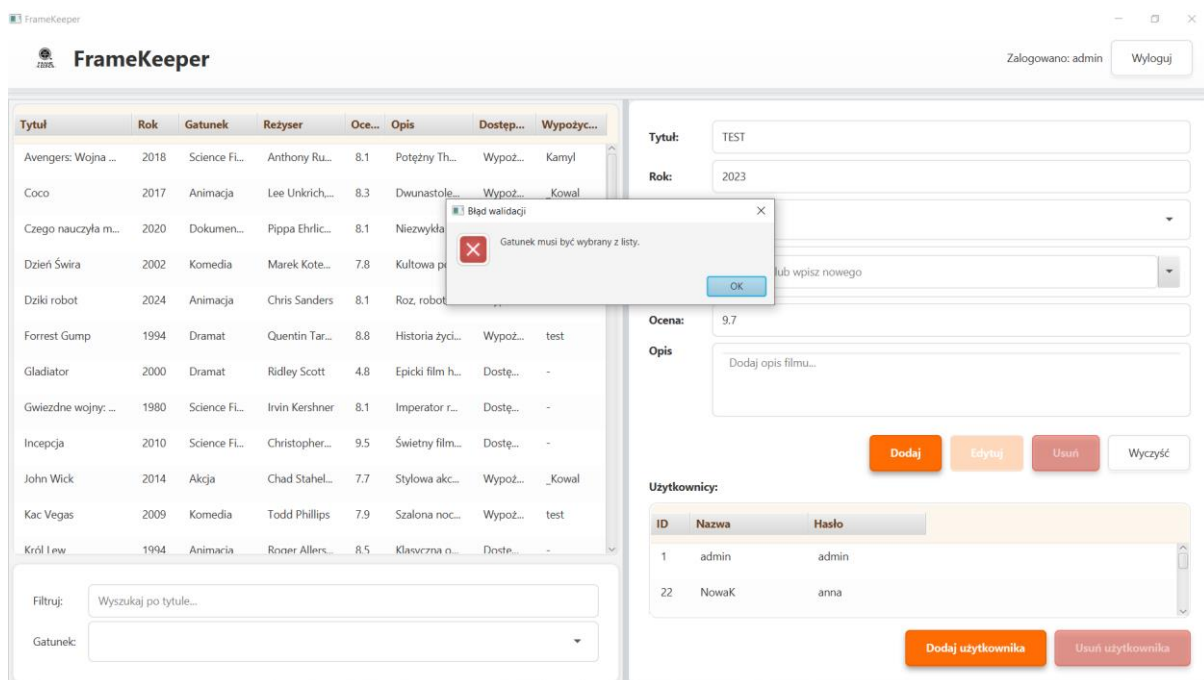


Rysunek 22: Pole ocena musi być liczbą typu 8 lub 8.1

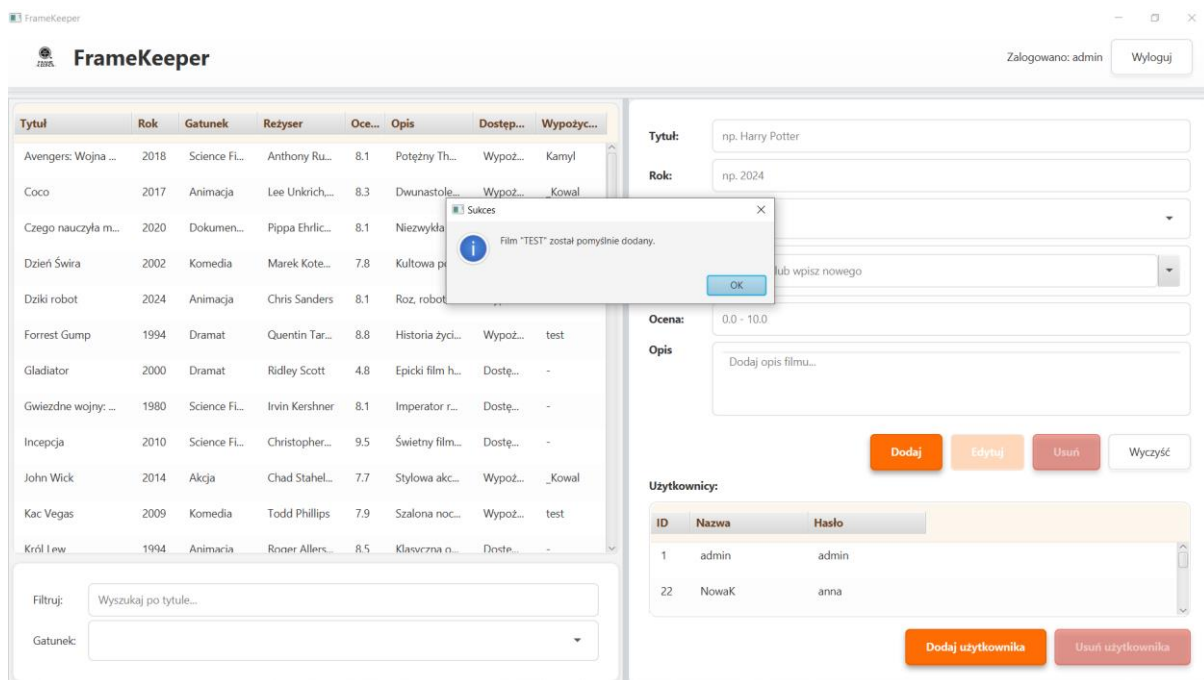




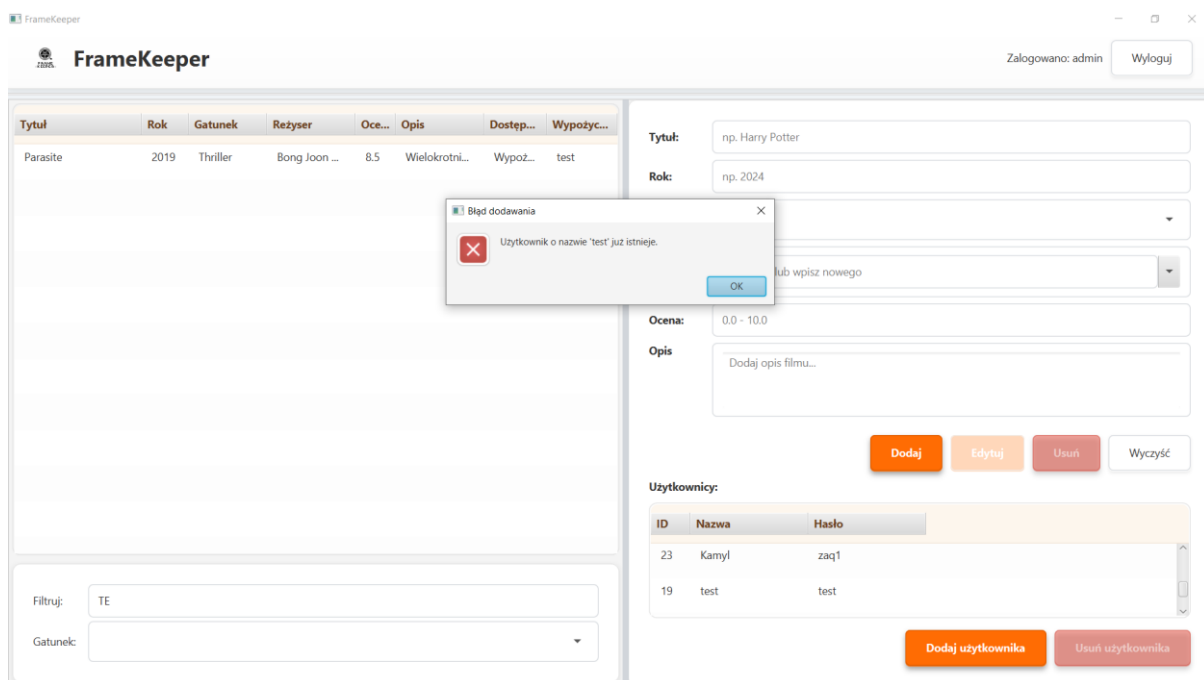
Rysunek 23: Pole gatunek musi zostać wybrane z listy



Rysunek 24: Pole reżyser musi być wybrane z listy

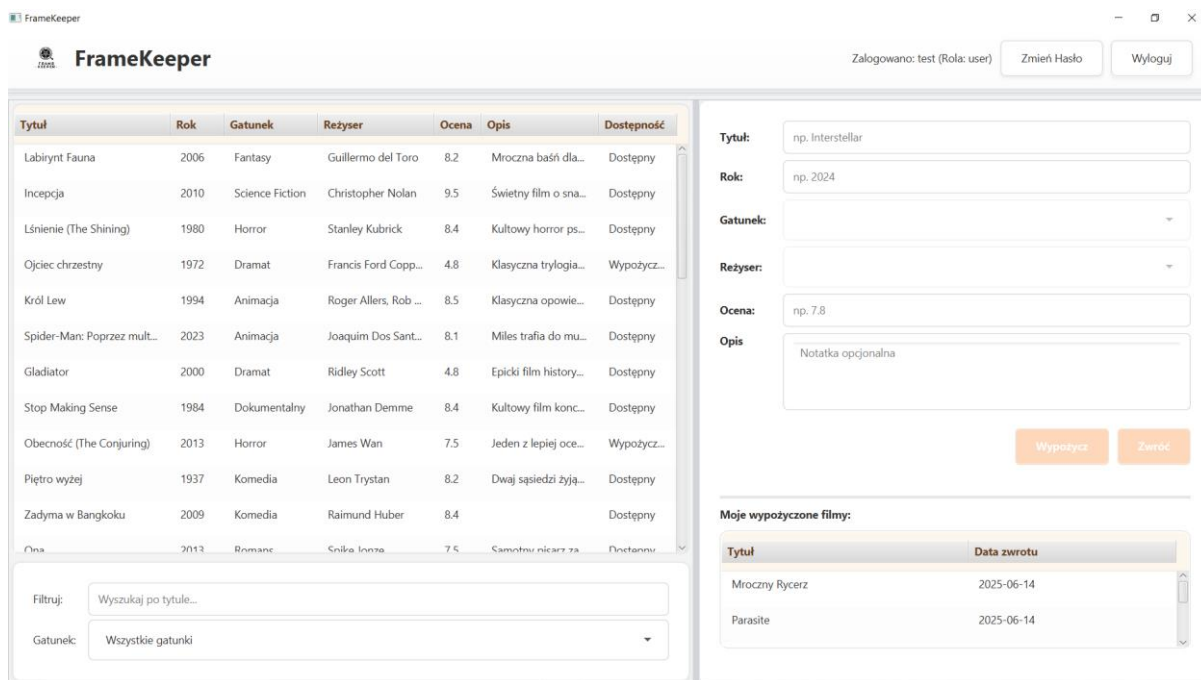


Rysunek 25: Pole opis nie jest wymagane, film został utworzony pomyślnie

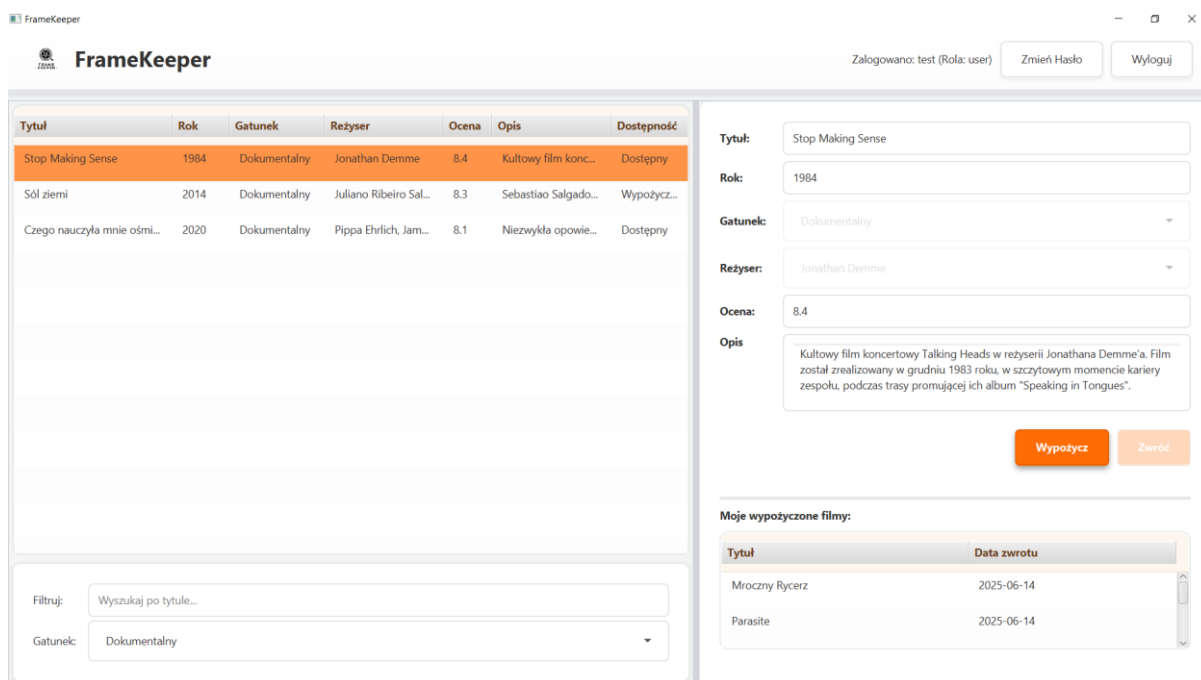


Rysunek 26: Tworząc użytkownika login nie może być taki sam jak login innego

## 1.6 Obsługa klienta



Rysunek 27: Widok po zalogowaniu się na klienta "test"



Rysunek 28: Sortowanie analogiczne do sortowania na koncie admin

FrameKeeper

Zalogowano: test (Rola: user)

Zmień Hasło

Wyloguj

Tytuł	Rok	Gatunek	Reżyser	Ocena	Opis	Dostępność
Labyrint Fauna	2006	Fantasy	Guillermo del Toro	8.2	Mroczna baśń dla...	Dostępny
Incepcja	2010	Science Fiction	Christopher Nolan	9.5	Świetny film o sna...	Dostępny
Lśnienie (The Shining)	1980	Horror	Stanley Kubrick	8.4	Kultowy horror ps...	Dostępny
Ojciec chrzestny	1972	Dramat	Francis Ford Copp...	4.8	Klasyczna trylogia...	Wypożycz...
Król Lew	1994	Animacja	Roger Allers, Rob ...	8.5	Klasyczna opowie...	Dostępny
Spider-Man: Poprzez mult...	2023	Animacja	Joaquim Dos Sant...	8.1	Miles trafia do mu...	Dostępny
Gladiator	2000	Dramat	Ridley Scott	4.8	Epicki film history...	Dostępny
Stop Making Sense	1984	Dokumentalny	Jonathan Demme	8.4	Kultowy film konc...	Dostępny
Obecność (The Conjuring)	2013	Horror	James Wan	7.5	Jeden z lepszej oce...	Wypożycz...
Piętro wyżej	1937	Komedia	Leon Trystan	8.2	Dwaj sąsiedzi żyją...	Dostępny
Zadyma w Bangkoku	2009	Komedia	Raimund Huber	8.4		Dostępny
...	...	...	...	...	...	...

Filtruj:

Wyszukaj po tytule...

Gatunek:

Wszystkie gatunki

Tytuł:

Labyrint Fauna

Rok:

2006

Gatunek:

Fantasy

Reżyser:

Guillermo del Toro

Ocena:

8.2

Opis

Mroczna baśń dla dorosłych, osadzona w realiach hiszpańskiej wojny domowej.

Wypożycz

Zwróć

Moje wypożyczone filmy:

Tytuł	Data zwrotu
Mroczny Rycerz	2025-06-14
Parasite	2025-06-14

Rysunek 29: Pola textfield oraz combobox są wyłączane z edycji

FrameKeeper

Zalogowano: test (Rola: user)

Zmień Hasło

Wyloguj

Tytuł	Rok	Gatunek	Reżyser	Ocena	Opis	Dostępność
Labyrint Fauna	2006	Fantasy	Guillermo del Toro	8.2	Mroczna baśń dla...	Dostępny
Incepcja	2010	Science Fiction	Christopher Nolan	9.5	Świetny film o sna...	Dostępny
Lśnienie (The Shining)	1980	Horror	Stanley Kubrick	8.4	Kultowy horror ps...	Dostępny
Ojciec chrzestny	1972	Dramat	Francis Ford Copp...	4.8	Klasyczna trylogia...	Wypożycz...
Król Lew	1994	Animacja	Roger Allers, Rob ...	8.5	Klasyczna opowie...	Dostępny
Spider-Man: Poprzez mult...	2023	Animacja	Joaquim Dos Sant...	8.1	Miles trafia do mu...	Dostępny
Gladiator	2000	Dramat	Ridley Scott	4.8	Epicki film history...	Dostępny
Stop Making Sense	1984	Dokumentalny	Jonathan Demme	8.4	Kultowy film konc...	Dostępny
Obecność (The Conjuring)	2013	Horror	James Wan	7.5	Jeden z lepszej oce...	Wypożycz...
Piętro wyżej	1937	Komedia	Leon Trystan	8.2	Dwaj sąsiedzi żyją...	Dostępny
Zadyma w Bangkoku	2009	Komedia	Raimund Huber	8.4		Dostępny
...	...	...	...	...	...	...

Filtruj:

Wyszukaj po tytule...

Gatunek:

Wszystkie gatunki

Tytuł:

Spider-Man: Poprzez multiwersum

Rok:

2023

Gatunek:

Fantasy

Reżyser:

Joaquim Dos Santos, Kemp Powers

Ocena:

8.1

Opis

Miles trafia do multiwersum, w którym spotyka innych Spider-Manów. Sam musi odkryć, co dla niego oznacza bycie bohaterem oraz jak uratować tych, na których mu zależy.

Wypożycz

Zwróć

Moje wypożyczone filmy:

Tytuł	Data zwrotu
Mroczny Rycerz	2025-06-14
Parasite	2025-06-14

Potwierdź wypożyczenie

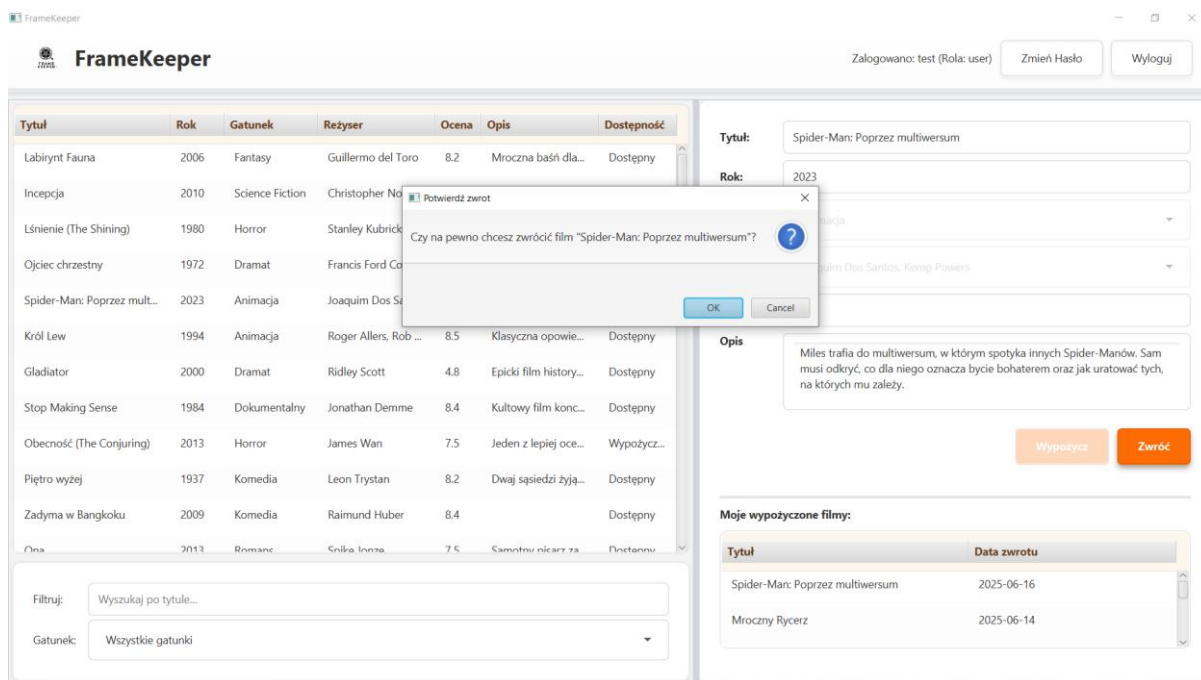
Czy na pewno chcesz wypożyczyć film "Spider-Man: Poprzez multiwersum"?

Okres wypożyczenia to 14 dni.

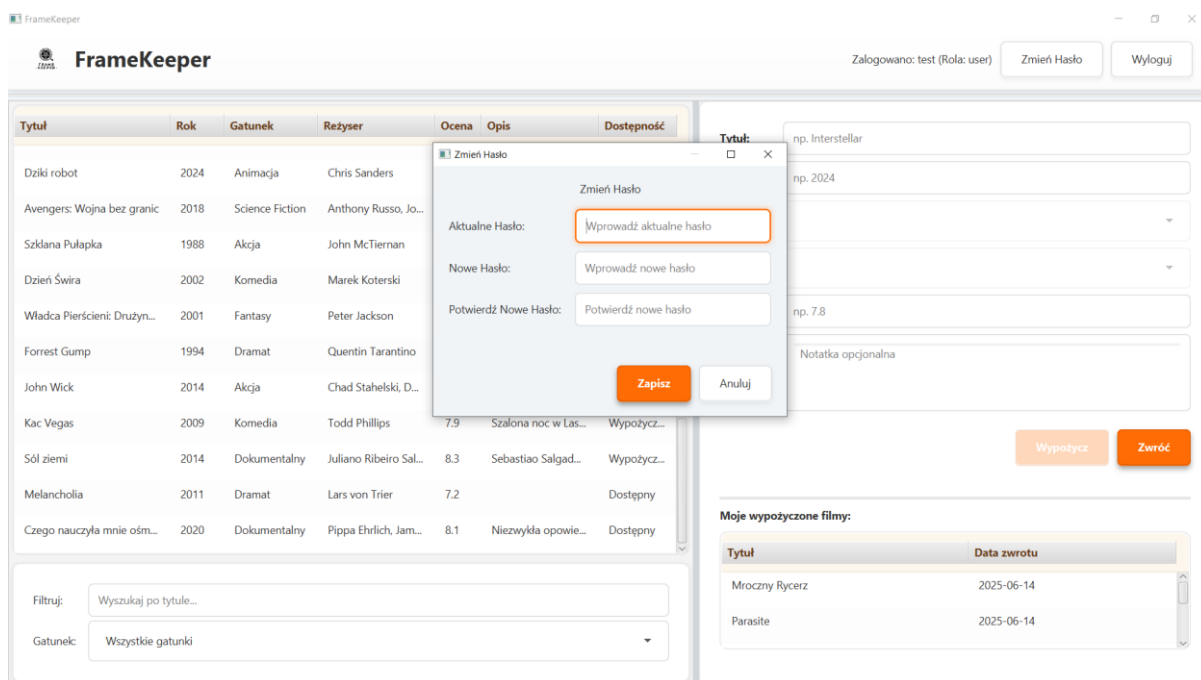
OK

Cancel

Rysunek 30: Wypożyczenie filmu na 14 dni

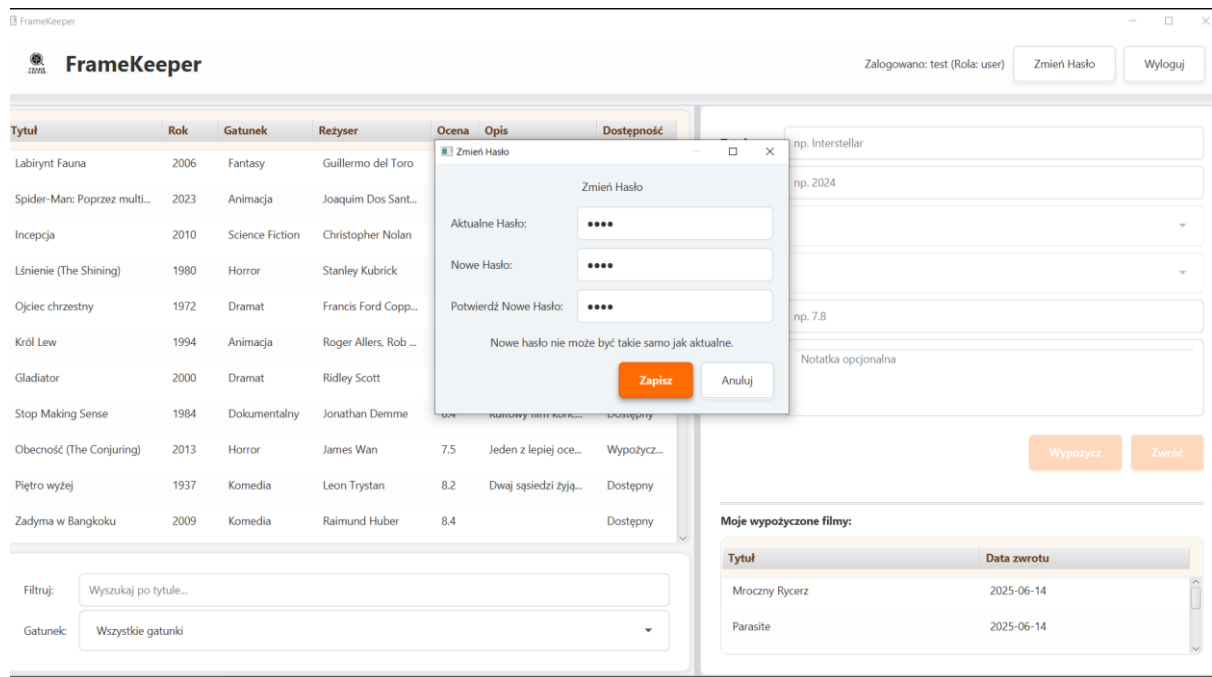


Rysunek 31: Zwrot filmu przed czasem



Rysunek 32: zmiana hasła użytkownika

## 1.7 Obsługa błędów na koncie użytkownika



Rysunek 33: hasło nie może być identyczne

## 2. Link do repozytorium projektu

[https://github.com/MR134966/Projekt\\_Java](https://github.com/MR134966/Projekt_Java)