Projekt aplikacji – System Zarządzania Wypożyczalnią Filmów (SAD)

# 1. Wstęp

Dokument opisuje architekturę techniczną oraz projekt aplikacji desktopowej i serwera API do obsługi wypożyczalni filmów. Projekt realizowany jest w technologii .NET, z podziałem na warstwę prezentacji, logiki biznesowej oraz danych.

# 2. Cel i zakres dokumentu

Celem dokumentu jest przedstawienie:  
- podziału aplikacji na warstwy,  
- architektury klas i pakietów,  
- interakcji między komponentami,  
- dynamiki działania aplikacji,  
- modelu danych i operacji bazodanowych.

# 3. Definicja architektury aplikacji

System składa się z dwóch głównych komponentów:  
- Aplikacja WPF (.NET) – frontend dla operatora  
- ASP.NET Core WebAPI – backend z bazą danych  
  
Architektura oparta jest na podejściu klient–serwer oraz wzorcu MVVM po stronie klienta.

# 4. Cele i ograniczenia architektury

Cele:  
- separacja warstw,  
- możliwość rozbudowy (np. o logowanie, RESTAuth),  
- obsługa raportów i eksportu danych.  
  
Ograniczenia:  
- brak uwierzytelniania użytkowników,  
- API musi być lokalnie hostowane.

# 5. Obraz logiczny aplikacji

# 5.1 Charakterystyka pakietów

- Lista3.Model – definicja modelu Rental, repozytorium REST  
- Lista3.ViewModel – logika akcji użytkownika, obsługa GUI  
- Lista3.View – interfejs użytkownika (Add, Edit, Reports)  
- RentalWebApi.Controllers – RESTful API dla Rentals  
- MovieRental.Data – EF Core + DbContext

# 5.2 Diagram klas aplikacji

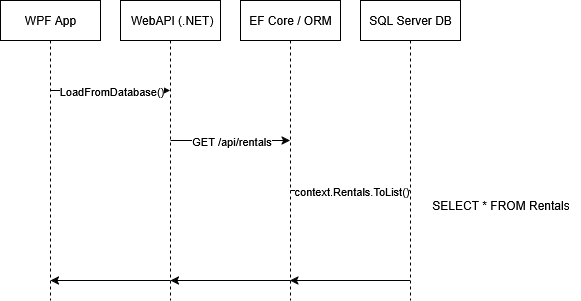
[Diagram UML klas: Rental, RentalRepository, ViewModel, APIController]

# 5.3 Specyfikacja funkcji i metod

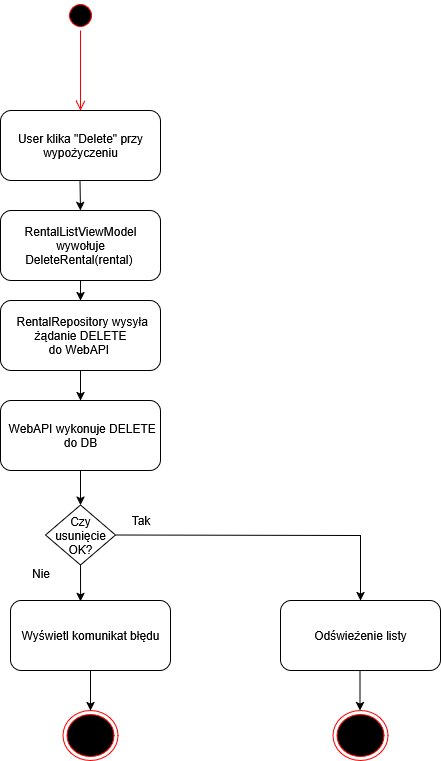
- RentalListViewModel.LoadRentals() – ładuje dane z API  
- RentalListViewModel.AddRental() – tworzy nowe wypożyczenie  
- ReportsWindowViewModel.ExportToPdf() – eksport raportu  
- RentalsController.PostRental() – zapis nowego wypożyczenia  
- MovieRentalContext.SaveChangesAsync() – operacja DB

# 6. Dynamiczny obraz modelowanej aplikacji

**6.1 Diagram sekwencji UML dla obiektów:**

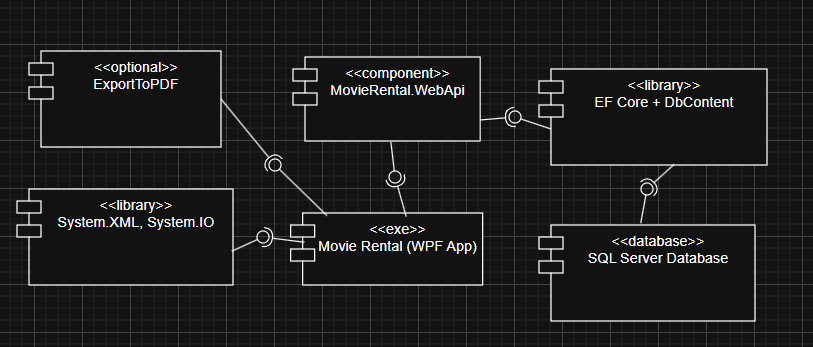


# 6.2 Diagram aktywności UML dla obiektów 6.2.1 Diagram aktywności UML dla usuwania wypożyczeń

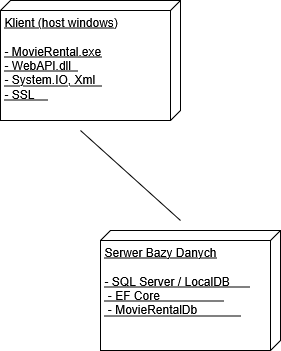


# 7. Statyczny obraz modelowanej aplikacji

# 7.1 Diagram komponentów UML



# 7.2 Diagram instalacji UML



# 8. Diagram use-caseów

# 9. Projekt bazy danych

- Encja: Rental  
 - Id: int (PK)  
 - MovieTitle: string  
 - CustomerName: string  
 - CustomerSurname: string  
 - RentalDate: DateTime  
 - ReturnDate: DateTime?  
  
- Operacje:  
 - POST /api/rentals – dodaj  
 - PUT /api/rentals/{id} – edytuj  
 - DELETE /api/rentals/{id} – usuń  
 - PUT /api/rentals/return/{id} – zwrot  
  
- Pliki XML: eksport/import z WPF (OpenFileDialog / SaveFileDialog)  
- Skrypty EF Core: automatyczna migracja + MovieRentalContext