Opis wersji aplikacji – System Zarządzania Wypożyczalnią Filmów (SVD)

# 1. Wstęp

Dokument zawiera opis bieżącej wersji aplikacji MovieRental, obejmującej komponenty klienckie (WPF) oraz serwerowe (ASP.NET WebAPI). System umożliwia zarządzanie wypożyczeniami, generowanie raportów i eksport danych.

# 2. Zakres dokumentu

Opisuje komponenty, strukturę projektu, wymagania środowiskowe, sposób instalacji, listę zmian względem poprzedniej wersji oraz wykonane testy.

# 3. Informacje o wersji/release aplikacji

- Wersja: 1.0.0  
- Data wydania: 23.06.2025  
- Status: wersja produkcyjna / szkoleniowa  
- Moduły objęte: klient WPF, serwer WebAPI, baza danych SQL Server

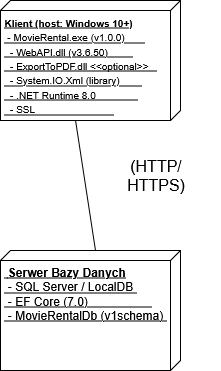
# 4. Charakterystyka komponentów/pakietów

- MovieRentalAPI – backend REST (ASP.NET Core), obsługa bazy i logiki danych  
- MovieRentalClient – aplikacja WPF (MVVM), interfejs operatora  
- Model + Repozytorium – komunikacja z API, serializacja, walidacja  
- Raporty – generowanie PDF i CSV, wykresy (OxyPlot)

# 5. Lista pakietów aplikacji

- Lista3.Model  
- Lista3.ViewModel  
- Lista3.View  
- RentalWebApi.Controllers  
- MovieRental.Data  
- OxyPlot, iTextSharp, RestSharp

# 6. Diagram instalacji UML aplikacji



# 7. Zmiany od poprzedniej wersji

(wersja 1.0.0 jest pierwszym wydaniem)  
- Utworzono API (GET/POST/PUT/DELETE)  
- Dodano WPF UI z MVVM  
- Obsługa eksportu danych  
- Generowanie wykresów aktywności  
- Obsługa daty zwrotu wypożyczenia

# 8. Konfiguracja środowiska przed instalacją

- .NET 6.0 SDK  
- SQL Server Express  
- Visual Studio 2022  
- Skonfigurowany connection string DefaultConnection w appsettings.json  
- Uprawnienia sieciowe (CORS) lokalnie

# 9. Instrukcja instalacji

1. Zbuduj rozwiązanie w Visual Studio.  
2. Upewnij się, że migracje EF Core są wykonane.  
3. Uruchom API (dotnet run lub IIS Express).  
4. Uruchom aplikację WPF (Rentals.xaml).  
5. Przetestuj połączenie (GET z /api/rentals).

# 10. Problemy i nieusunięte błędy

- Brak zabezpieczenia dostępu (brak autoryzacji)  
- Wersja API obsługuje tylko jedno źródło danych  
- Nieobsługiwane konflikty przy jednoczesnej edycji

# 11. Pakiety

# 11.1 Pakiet: Lista3.ViewModel

Zawiera logikę widoków, komendy i operacje na repozytorium.

# 11.2 Lista plików (przykładowa)

- RentalListViewModel.cs  
- ReportsWindowViewModel.cs  
- RelayCommand.cs

# 11.3 Instrukcja kompilacji

- Wymagany projekt .csproj (.NET WPF)  
- Restore NuGet (dotnet restore)  
- Build (dotnet build)

# 12. Specyfikacja unit testów

# 12.1 Lista wykonanych test case’ów

- Dodanie wypożyczenia (formularz → API → DB)  
- Edycja wypożyczenia (edycja → PUT → walidacja)  
- Eksport do PDF i CSV  
- Dodanie daty zwrotu (PUT /return)

# 12.2 Raport o błędach

- Eksport PDF wymaga ponownego renderu wykresu  
- API nie waliduje poprawności daty zwrotu (może być < RentalDate)