Plan QA – System Zarządzania Wypożyczalnią Filmów

# 1. Cel dokumentu

Celem niniejszego dokumentu jest zdefiniowanie strategii zapewnienia jakości (QA) dla projektu MovieRental oraz opisanie standardów, procedur, zasobów i odpowiedzialności związanych z jakością oprogramowania na każdym etapie jego wytwarzania.

# 2. Zakres dokumentu

Zakres QA obejmuje analizę, projekt, implementację, testowanie, integrację oraz wdrożenie aplikacji klient-serwer. QA odnosi się do kodu aplikacji WPF, API oraz integracji z bazą danych.

# 3. Charakterystyka projektu i budowanego systemu

System składa się z aplikacji desktopowej (WPF) oraz serwera RESTful (ASP.NET Core WebAPI) zintegrowanych z bazą danych SQL Server. System umożliwia zarządzanie wypożyczeniami, generowanie raportów i eksport danych.

# 4. Identyfikacja obszaru QA

QA obejmuje następujące komponenty:  
- RentalListViewModel, ReportsWindowViewModel  
- RentalRepository, MovieRentalContext  
- RentalsController  
- integrację z bazą danych  
- eksport do PDF i CSV  
- komunikację REST API

# 5. Standardy i procedury QA

- QA będzie prowadzone zgodnie ze standardami .NET, MVVM, REST  
- Wersjonowanie i kontrola zmian: Git  
- Review kodu przed integracją  
- Każda funkcjonalność musi być przetestowana (manualnie lub automatycznie)

# 6. Relacja do innych planów projektu

QA jest skoordynowane z planem projektu i dokumentacją SAD oraz planem testów. QA weryfikuje zgodność implementacji z wymaganiami zawartymi w dokumentach analizy i projektu.

# 7. Referencje

- Analiza specyfikacji wymagań  
- SAD – Projekt aplikacji  
- Plan projektu  
- Opis wersji (SVD)

# 8. QA w strukturze organizacyjnej projektu

- Odpowiedzialność za QA ponosi wykonawca projektu.  
- Proces QA obejmuje działania w zakresie testowania, walidacji kodu, dokumentowania i raportowania defektów.

# 9. Zasoby QA

- Sprzęt: komputer z Windows, narzędzia debugowania  
- Oprogramowanie: Visual Studio, Postman, OxyPlot, iTextSharp  
- Czas: 1 tydzień testów integracyjnych i funkcjonalnych

# 10. Zadania QA

- Weryfikacja implementacji względem specyfikacji  
- Przeprowadzanie testów jednostkowych i integracyjnych  
- Rejestrowanie i śledzenie błędów  
- Weryfikacja naprawy defektów

# 11. Zakres odpowiedzialności za procesy QA

- Twórca aplikacji: testy funkcjonalne, poprawki  
- Tester: rejestracja i analiza błędów  
- Dokumentalista: aktualizacja opisu wersji i historii zmian

# 12. Harmonogram

QA następuje po implementacji i przed wydaniem wersji finalnej:  
- Tydzień 6–7: testowanie i QA  
- Tydzień 8: poprawki i zatwierdzenie

# 13. Dokumentacja

- Test Plan  
- Checklisty funkcjonalne  
- Raport testów  
- Lista błędów i ich statusów

# 14. Standardy i procedury

- Kodowanie zgodne z C# .NET Core  
- MVVM dla klienta  
- REST dla API  
- EF Core dla bazy  
- Procedury testowe: checklisty + przypadki testowe

# 15. Testy

- Testy jednostkowe (manualne): CRUD, eksport  
- Testy integracyjne: komunikacja z API  
- Testy systemowe: scenariusze użytkownika

# 16. Zasady rejestracji i raportowania o błędach

- Rejestracja błędów w arkuszu Excel lub GitHub Issues  
- Każdy błąd posiada: ID, opis, data zgłoszenia, status, osoba odpowiedzialna  
- Codzienna weryfikacja statusu defektów

# 17. Zarządzanie ryzykiem

Tabela ryzyk

| Ryzyko | Skutek | Działanie zapobiegawcze |
| --- | --- | --- |
| Brak testów API | Złe dane w systemie | Testy Postman, walidacja |
| Błąd w eksporcie | Uszkodzony raport | Testy PDF/CSV przed wydaniem |
| Niezgodność z wymaganiami | Brak akceptacji | Przeglądy i testy regresji |