****

Specyfikacja wymagań aplikacji „Musify Management Panel”

Opracował: Jakub Toma

Spis treści

[1. Wymagania funkcjonalne 3](#_Toc115201948)

[2. Wymagania niefunkcjonalne 3](#_Toc115201949)

[3. Przypadki użycia (Use case’y) 5](#_Toc115201950)

[3.1. Użytkownik zarządza utworami 5](#_Toc115201951)

[3.2. Użytkownik generuje raport 6](#_Toc115201952)

# Wymagania funkcjonalne

* Aplikacja powinna pozwolić użytkownikowi zarządzać biblioteką utworów, która jest sczytywana z bazy danych
* Aplikacja powinna umożliwić operacje tj. Dodawanie do bazy, Usuwanie, Wczytywanie i edycja
* Aplikacja powinna posiadać funkcję wygenerowania raportu z bazy danych

# Wymagania niefunkcjonalne

**Aplikacja:**

* Formularz do dodawania powinien być w osobnym oknie
* Wygenerowany raport powinien pojawić się w nowym oknie
* Aplikacja ma rozpoznać edytowany utwór lub do usunięcia po jego zaznaczeniu na liście
* Operacja na krotce nie powinna trwać dłużej niż 2 sekundy
* Aplikacja nie powinna potrzebować więcej niż 1 GB pamięci RAM
* Generowanie raportu nie powinno trwać dłużej niż 10 sekund
* Interfejs graficzny powinien być przejrzysty i łatwy w obsłudze
* Interfejs graficzny powinien mieć jak najmniej pustej przestrzeni
* Aplikacja nie powinna pozwolić na wprowadzenie błędnych danych
* Aplikacja powinna być dostępna 24/7/365
* Wszystkie błędy krytyczne powinny być naprawione w ciągu 2 dni roboczych
* Wdrożenie aplikacji nie powinno trwać dłużej niż 1 dzień

**Baza danych:**

* System bazy danych powinien umożliwić komunikację z aplikacją napisaną w C#
  + - * System bazy danych powinien obsługiwać zapytania SQL
      * Osoby nieupoważnione nie mają dostępu do bazy danych
      * System bazy danych powinien cyklicznie wykonywać kopię bazy danych
      * Model bazy danych odzwierciedla świat rzeczywisty (w tym przypadku katalog utworów)
      * Baza danych powinna być dostępna 24/7/365 średnio 97% czasu

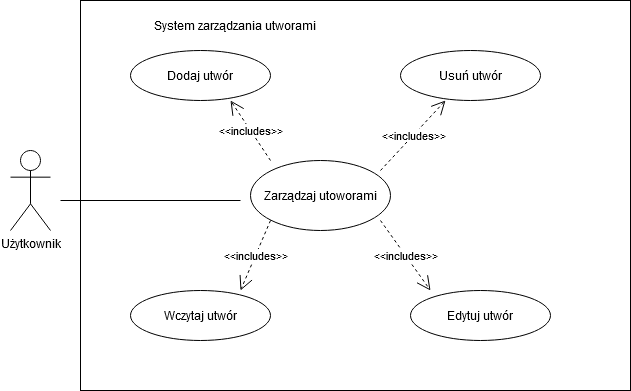
**Stosowane Standardy:**

* Aplikacja powinna bazować na programowaniu obiektowym
* Aplikacja bazuje na języku C#
* Aplikacja powinna korzystać z silnika graficznego WPF
* Aplikacja korzysta z bindowania danych
* Baza danych bazuje na Systemie SQL Server 2019
* System generowania raportów bazuje na systemie SQL Reporting Services

# Przypadki użycia (Use case’y)

## Użytkownik zarządza utworami

|  |  |
| --- | --- |
| Numer | 1 |
| Nazwa | Zarządzanie rekordami utworów |
| Autor | Jakub Toma |
| Typ use case’a | ogólny |
| Aktorzy | Użytkownik, system zarządzania utworami |
| Krótki opis | Użytkownik zarządza utworami z bazy danych |
| Główny przepływ zdarzeń | Użytkownik korzysta z programu w celu dodania nowego utworu do bazy danych |
| Alternatywny przepływ zdarzeń | Użytkownik korzysta z programu w celu usunięcia utworu z bazy danych |
| Alternatywny przepływ zdarzeń | Użytkownik korzysta z programu w celu edycji utworu |
| Alternatywny przepływ zdarzeń | Użytkownik korzysta z programu w celu wyświetlenia utworów z bazy danych |
| Specjalne wymagania |  |
| Notatki |  |



## Użytkownik generuje raport

|  |  |
| --- | --- |
| Numer | 1 |
| Nazwa | Obsługa raportów |
| Autor | Jakub Toma |
| Typ use case’a | ogólny |
| Aktorzy | Użytkownik, system generowania raportów |
| Krótki opis | Użytkownik generuje raport |
| Główny przepływ zdarzeń | Użytkownik korzysta z programu w celu wygenerowania raportu wszystkich utworów z bazy danych |
| Specjalne wymagania |  |
| Notatki |  |

