Jarosław Kuchta

System Quest

13.09.2025

# Potrzeba

System Quest jest implementacją i udoskonaleniem arkusza kalkulacyjnego QUEST.XLSM. Arkusz Quest był wykorzystywany w ramach przedmiotu Jakość Systemów Informatycznych do oceny jakości dokumentów projektowych. Został napisany po raz pierwszy w 2003 roku dla aplikacji Microsoft Excel. W ciągu 20 lat aplikacja ta podlegała licznym zmianom, w wyniku których wykorzystanie możliwości liczenia średniej ważonej potrzebnej do przeliczania ocen stało się utrudnione. Ponadto wielu studentów odmawia wykorzystywania tej aplikacji w pełnej wersji płatnej, a wersja online udostępniana przez PG dla studentów nie posiada tej możliwości. Studenci wolą również korzystać z alternatywnych aplikacji kalkulacyjnych, które są darmowe i co prawda umożliwiają eksportowania danych do i z aplikacji MS Excel, ale ostateczne przeliczenie danych i tak wymaga dostępu do płatnej wersji MS Excel.

Funkcjonalność arkusza Quest była dość uboga i wymagała od studentów precyzji rozumienia stosowanych ocen, np. ocena „1” nie mogła być stosowana automatycznie, lecz rozważana w powiązaniu z ocenami „brak oceny”, „nie wymagane”. Studenci często zaniedbywali takie rozważania, co wymagało zwrócenia uwagi ze strony prowadzącego zajęcia.

Powyższe problemy stanęły u podstaw koncepcji opracowania rozwiązania programistycznego opartego o nowoczesne technologie, które zastąpiłoby arkusz kalkulacyjny.

# Wymagania

Rozwiązanie to powinno mieć następujące cechy (uporządkowane w kolejności od najwyższego do najniższego priorytetu):

1. Powinno naśladować funkcjonalność arkusza kalkulacyjnego Quest w zakresie:
   1. przeglądania arkusza ocen w formie drzewa metryk jakości,
   2. wprowadzania ocen w enumeratywnie określonej skali,
   3. wprowadzania uzasadnienia ocen w postaci krótkiego tekstu,
   4. przeliczania ocen metodą średniej ważonej w drzewie, przy czym wagi muszą być proponowane zgodnie z szablonem drzewa jakości,
   5. oceniania wielu dokumentów projektowych,
   6. agregacji ocen wielu dokumentów do oceny całej dokumentacji projektowej,
   7. zapisywania ocen szczegółowych i zagregowanych w pliku, który może być przekazywanych dla innych studentów.
   8. przedstawiania ocen zagregowanych w formie tabelarycznej i wykresów słupowych zarówno dla każdego dokumentu osobno, jak i dla całej dokumentacji,
   9. eksportowania tabel i wykresów ocen zagregowanych w formie umożliwiającej włączenie ich do raportu jakości (dokumentu tworzonego przez studentów w formacie DOCX).
2. W pierwszej wersji dopuszczalne jest zapisywanie danych na komputerze lokalnym. W drugiej system Quest musi umożliwiać:
   1. rejestrowanie studentów w kilkuosobowych zespołach oceniających tę samą dokumentację,
   2. równoczesne ocenianie dokumentów i zapisywanie jednej wspólnej wersji ocen, przy czym w przypadku ocen rozbieżnych system powinien wyróżniać rolę mediatora, którego ocena jest decydująca.
3. W pierwszej i drugiej wersji szablony oceny są ustalone z góry i zapisane w formie plików tekstowych. W wersji trzeciej system powinien wyróżnić rolę prowadzącego zajęcia i umożliwić mu:
   1. modyfikację szablonów ocen przed ich udostępnieniem dla studentów,
   2. modyfikację wag metryk jakości przed ostatecznym przeliczeniem ocen,
   3. przeglądanie arkuszy ocen zespołów studenckich oraz ich ocenianie zarówno punktowe, jak i opisowe.
4. W wersji czwartej konieczne jest:
   1. warunkowanie dostępności części arkusza ocen od odpowiedzi studentów na postawione pytania (np. pominięcie części arkusza przy braku pewnych elementów ocenianego projektu). To powinno wyeliminować niepewność studentów co do sposobu oceniania.
5. W wersji piątej system jest przygotowywany do pracy zdalnej. W tym celu potrzebne jest:
   1. prowadzenie rozmowy (czatu) z pozostałymi studentami w zespole.
   2. synchroniczne wyświetlanie ocenianych dokumentów PDF tak, aby jeden student mógł zwrócić uwagę innym na pewien fragment dokumentacji,

Ta funkcjonalność może być zrealizowana za pomocą osobnego narzędzia (np. MS Teams). Wówczas system Quest powinien zapewniać integrację z tym narzędziem.

# Koncepcja systemu

System Quest powinien mieć architekturę modułową, wielowarstwową, możliwą do wdrożenia na komputerach osobistych, a następnie na wielu serwerach. Początkowo do przyjęcia jest ograniczenie do komputerów osobistych pracujących w systemie Windows 10/11. Architektura modułowa zapewni możliwość rozbudowy systemu, a wielowarstwowa – rozluźnienie zależności między modułami. Trzeba pamiętać o perspektywie wdrożenia na wielu serwerach (zwłaszcza pracujących w systemie Linux), dlatego w miarę możliwości korzystać z frameworków pracujących na .NET Core, który może być zainstalowany zarówno pod Windows, jak i Linux. Nie przewiduje się pracy systemu na urządzeniach mobilnych typu smartfon.

Na początek przewidywanych jest sześć modułów:

1. moduł danych modelu jakości (QuestQDM) – który będzie przechowywał i udostępniał modele jakości,
2. moduł danych podsystemu ocen (QuestRDM) – który będzie przechowywał wystawione oceny,
3. moduł importu danych (QuestIMP) – który będzie pomocny przy importowaniu arkuszy ocen z aplikacji MS Access,
4. moduł eksportu wyników (QuestEXP) – który będzie służył do eksportu tabel i wykresów do schowka i do plików,
5. moduł silnika ocen (QuestENG) – który będzie umożliwiał wprowadzanie i przeliczanie ocen, a także wykonywanie różnych operacji na połączonych modelach danych,
6. moduł prezentacji (QuestWPF) – który będzie służył do przeglądania i edycji ocen.

Wszystkie te sześć modułów będzie tworzyć aplikację użytkownika system Quest, przy czym pierwsze pięć modułów będzie miało postać bibliotek DLL, a moduł prezentacji będzie modułem głównym (programem EXE). Przewiduje się, że cała aplikacja zostanie napisana w technologii .NET Core dostępnej w systemach Windows i Linux, a moduł główny aplikacji zostanie napisany w technologii WPF, która jest dostępna w systemie Windows. W dalszej perspektywie przewiduje się użycie technologii Blazor, ASPNET Core i/lub .NET Aspire.

Przewiduje się, że aplikacja będzie się komunikowała z użytkownikiem w dwóch językach: polskim i angielskim – do wyboru przez użytkownika.

Przewidywana struktura wielowarstwowa wykorzystuje model MVVM i składa się z następujących warstw:

1. Warstwa danych (Data Layer) – jest to najniższa warstwa, która uniezależnia warstwy wyższe od sposobu przechowywania i przekazywania danych. Składa się z dwóch warstw podrzędnych:
   1. warstwy przechowywania danych (Data Storage) – do przechowywania danych będzie wykorzystywany początkowo tylko system plików, a następnie baza danych lub chmura,
   2. warstwy modelu danych (Data Models) – zawiera definicje danych w postaci zbioru klas, warstwa danych udostępnia dane dla wyższych warstw za pomocą tych właśnie klas.
2. Warstwa logiki systemowej (System Logic) – warstwa pośrednia, która również składa się z dwóch warstw podrzędnych:
   1. warstwy logiki wykonawczej (Executive Logic) – jest to warstwa zawierająca klasy grupujące metody operowania na połączonych modelach danych,
   2. warstwy modelu widoku (ViewModels) – jest to warstwa definiująca klasy modeli widoków udostępniające model danych w formie gotowej do prezentacji i komendy przekazujące żądania użytkownika do warstwy logiki wykonawczej.
3. Warstwa prezentacji (Presentation Layer) – warstwa najwyższa, która umożliwia użytkownikowi przeglądanie i edycję danych oraz wysyłanie żądań do warstwy logiki systemowej i przeglądanie wyników. Ponownie składa się z dwóch podwarstw:
   1. warstwy zasobów aplikacji (Resources) – zawiera zasoby tekstowe w różnych językach (w tym przypadku w języku polskim i angielskim), zasoby graficzne (obrazki i ikony), style stosowane do ustalania kolorystyki aplikacji oraz obiekty pomocnicze, takie jak konwertery wartości (przekształcające dane obiektowe na postać tekstową),
   2. warstwy widoków (Views) – zawiera deklaracje widoków i okien z wyróżnionym oknem głównym aplikacji.

Zależności między modułami a warstwami przedstawiono na rys. 1.

QuestWPF

Views

Resources

QuestEng

ViewModels

Executive Logic

QuestIMP

QuestEXP

Data Models

QuestQDM

QuestRDM

Data Storage

Rys. . Warstwy architektoniczne i moduły

Warstwa prezentacji jest w całości zaimplementowana w module głównym aplikacji. Warstwa modeli widoku jest zawarta w module QuestENG. Warstwa logiki wykonawczej jest obejmuje dodatkowo moduł importu i eksportu. Warstwa modeli danych jest zaimplementowana w modułach danych jakości i ocen. Warstwa przechowywania danych jest zaimplementowana częściowo w modułach danych, a częściowo we frameworkach poza systemem Quest.