

Podjęmowanie decyzji - projekt zaliczeniowy; wersja 0.1.0

Piotr Kopron Bartłomiej Słupik

2023.03.03

Spis treści

1	Wstęp	2
2	Specyfikacja	2
2.1	Zakres projektu	2
2.1.1	Cele projektu	2
2.1.2	Przykładowy Use Case, który powinien być możliwy do zrealizowania	2
2.2	Architektura	3
2.3	Konfiguracja	3
2.4	Zarządzanie testowaniem	4
2.5	Zarządzanie ryzykiem	4
3	Część projektowa	5
3.1	Założenia przed rozpoczęciem realizacji	5
3.2	Ograniczenia	5
3.3	Wzorce architektoniczne	5
3.4	Diagram komponentów	6
3.5	Diagram przepływu danych	7
3.6	Model bazy danych	8
3.7	Harmonogram realizacji	9

1 Wstęp

W niniejszym dokumencie przedstawione są założenia i projekt aplikacji zaliczeniowej z przedmiotu "Podjęmowanie Decyzji". System powinien umożliwiać użytkownikowi tworzenie rankingów obiektów, których pary mogą porównywać klienci. Następnie wyliczany będzie ranking końcowy w oparciu o jedną z metod EVM i GMM. W części 'Specyfikacja' przedstawione są dokładne założenia działania aplikacji z punktu widzenia użytkownika, takie jak przypadki użycia, przykładowy schemat interakcji, czy plan wdrażania kolejnych funkcjonalności. Następnie w części 'Projekt' zamieszczono plan od strony technicznej, w tym schemat działania back-endu i konstrukcję bazy danych.

2 Specyfikacja

2.1 Zakres projektu

2.1.1 Cele projektu

Podstawowym celem projektu jest przygotowanie aplikacji do grupowego podejmowania decyzji metodą AHP (metodą porównywania parami).

- aplikacja powinna wykorzystywać metodę EVM i/lub GMM do tworzenia rankingów oraz odpowiednie metody AIJ i/lub AIR do grupowania osądów (macierzy porównywania parami) pochodzących od różnych ekspertów.
- aplikacja powinna - pozwalać na używanie tzw. skali fundamentalnej ale również na definiowanie swojej własnej skali porównań przez użytkownika, w tym bezpośrednio skali numerycznej.
- aplikacja powinna działać w architekturze klient-serwer
- aplikacja powinna umożliwiać pełny eksport danych rankingowych (wszystkich danych wejściowych) do pliku JSON (przykładowy plik załączony w sekcji Ramowy Opis Projektu).

2.1.2 Przykładowy Use Case, który powinien być możliwy do zrealizowania

- Facylitator
 - Tworzy nowy ranking
 - * Definiuje nowe alternatywy
 - * Definiuje kryteria
 - * Definiuje pod-kryteria (jeśli to konieczne)
 - * Definiuje uczestników rankingu
 - * Określa parametry rankingu takie jak:

- Sposób liczenia rankingu (EVM / GMM etc).
 - Czy ranking ma być kompletny czy nie kompletny
 - Skale pomiarową
 - Sposób liczenia niespójności rankingu
 - Kolejność zadawanych pytań (losowa / konkretna)
 - Datę i czas od której ranking ma być dostępny
 - Datę i czas do której ranking ma być dostępny
- * Rozsyła zaproszenie do udziału w rankingu ekspertom (uczestnikom rankingu)
- Eksperci
 - Odpowiadają na pytania o porównanie parami alternatyw (na jednym ekranie powinno znaleźć się jedno pytanie (jedno porównanie parami dwóch alternatyw)
 - Odpowiadają na pytania o porównanie parami kryteriów (podobnie na jednym ekranie jedno porównanie)
 - "Naciskają" guzik [submit], informujący system, że z ich strony ocena się zakończyła.
- Facylitator
 - Sprawdza zebrane wyniki
 - Nadzoruje wykonanie rankingu
 - Rozsyła wyniki uczestnikom/ekspertom procesu (oraz decydentom zewnętrznym).
 - Eksportuje wszystkie dane procesu do formatu JSON.

2.2 Architektura

- Serwer: REST-based ASP .NET Core (C#)
- Klient: React+Next.js+TypeScript - zgodnie z mockami w osobnych dokumentach
- Wewnętrzne REST API - zgodnie z osobnym dokumentem
- Baza danych - EntityFramework Code First

2.3 Konfiguracja

- Frontend development
 - cd repooroot/web
 - npm run dev
 - Aplikacja uruchomiona na http: localhost:3000
- Backend

2.4 Zarządzanie testowaniem

BLOCKED BY: Brak zatwierdzenia specyfikacji przez Klienta

2.5 Zarządzanie ryzykiem

Podstawowym źródłem ryzyka są nieprzewidziane okoliczności. Zalecane ostrożne estymaty czasu pracy nad funkcjonalnościami. Przy tworzeniu aplikacji będą priorytetyzowane funkcjonalności krytyczne ponad błędami oraz aspektami wizualnymi i płynnością działania.

3 Część projektowa

3.1 Założenia przed rozpoczęciem realizacji

- Wykonanie testowej bazy w technice Code First aby opanować sprawne tworzenie migracji w Entity Framework

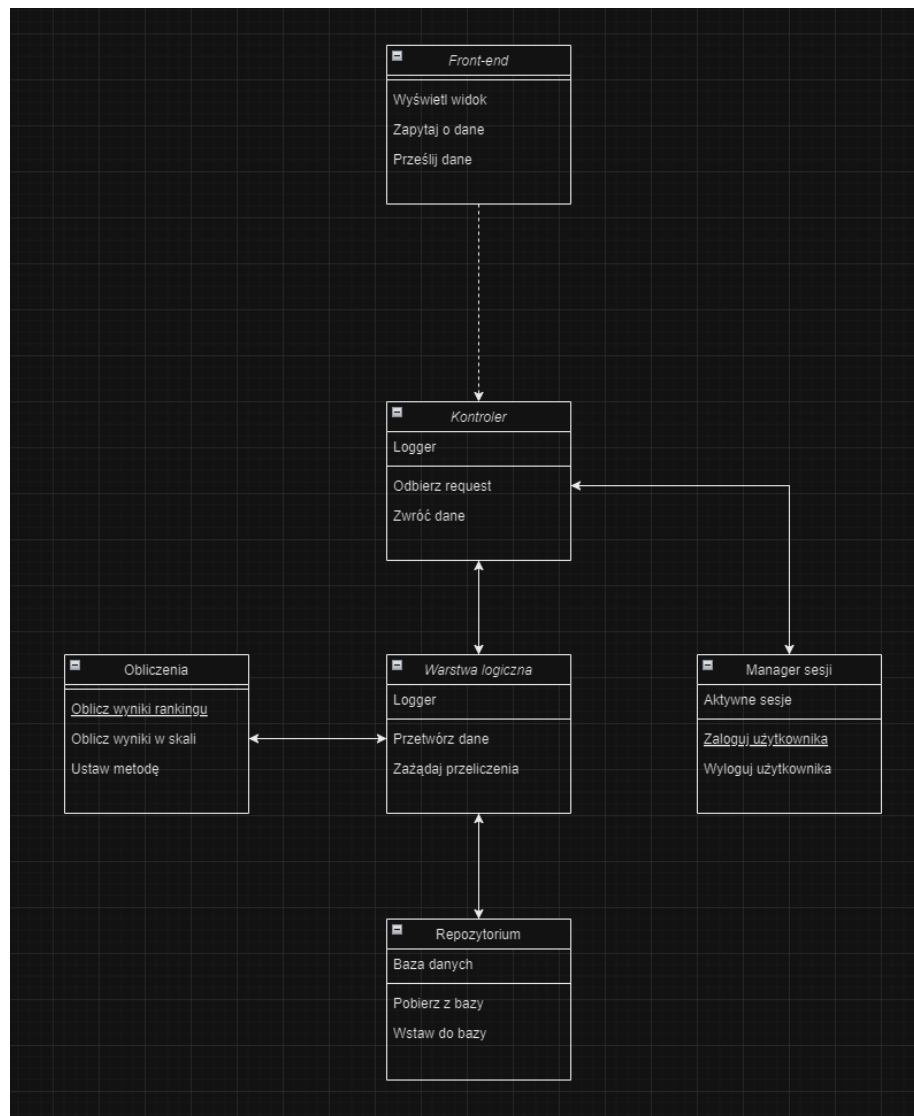
3.2 Ograniczenia

- Użytkownik będzie korzystał z aplikacji poprzez przeglądarkę internetową. Nie przewidujemy wykonania aplikacji mobilnej.

3.3 Wzorce architektoniczne

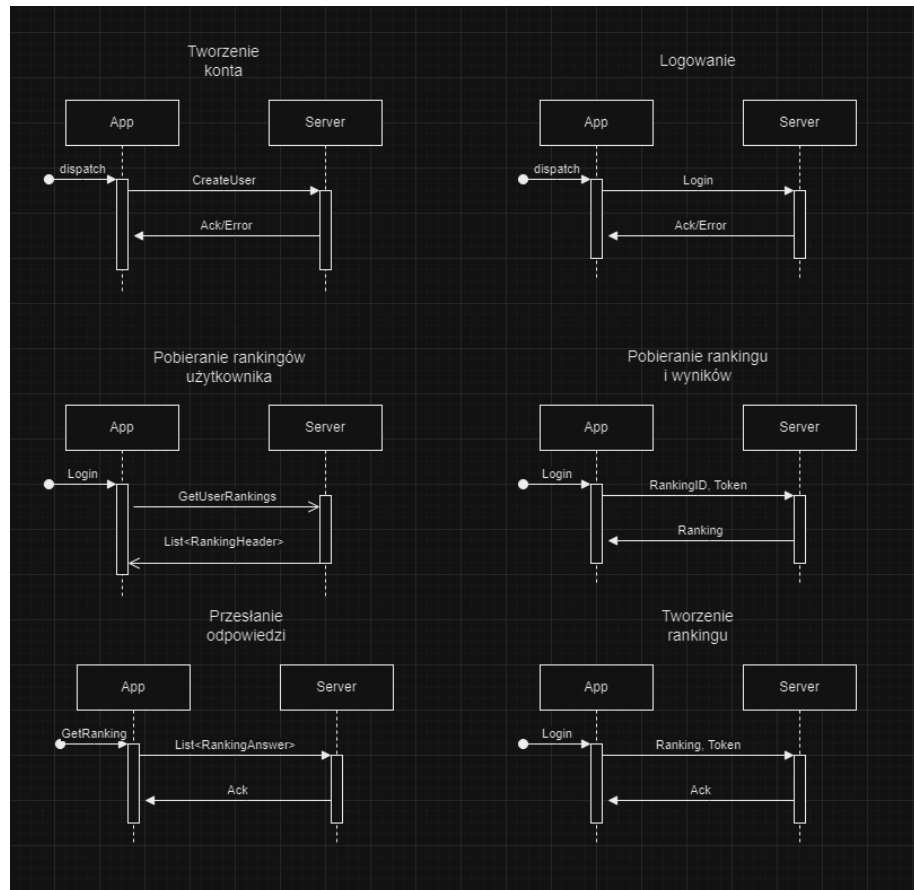
- Aplikacja będzie działać w modelu klient-serwer, opierając się na komunikacji przy pomocy kontrolera i modelu (MVC)
- Serwer będzie podzielony na warstwy przejmujące różne jego role, tj. dostępu aplikacji, obliczeniowa, dostępu do bazy danych.

3.4 Diagram komponentów



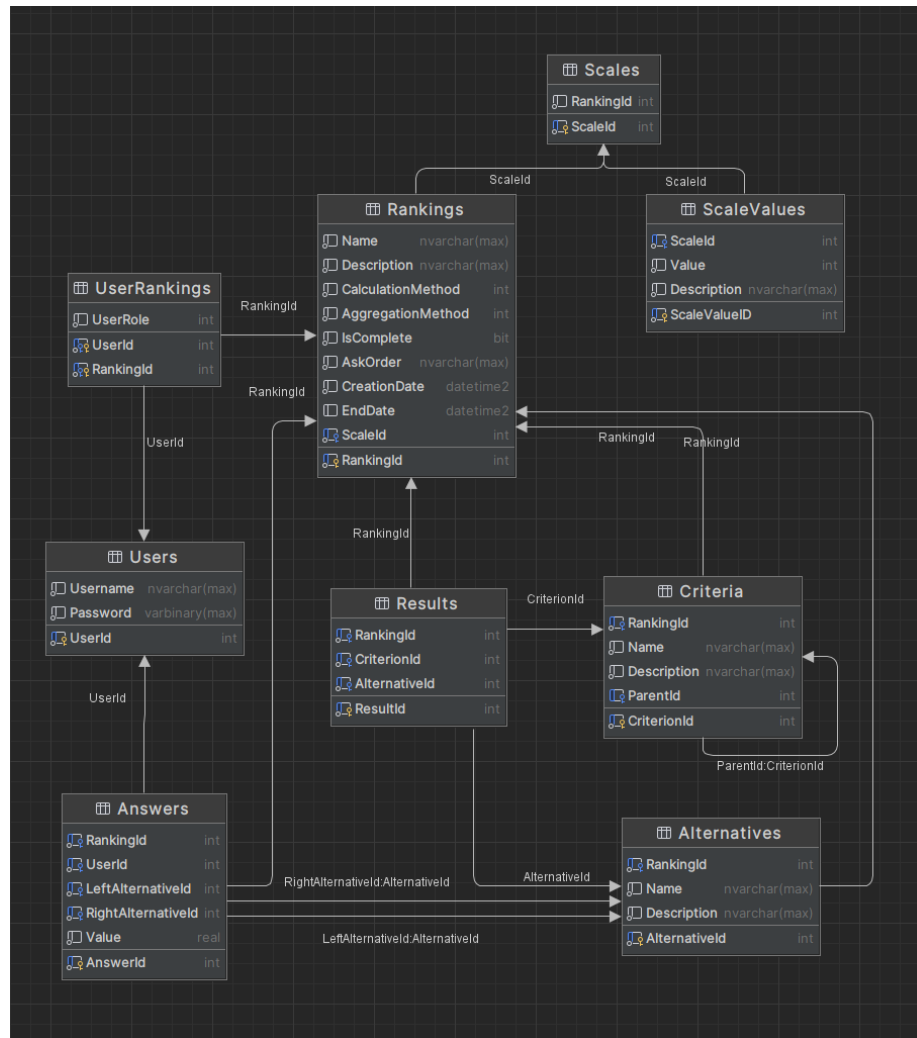
Rysunek 1: Diagram komponentów. Naszkicowane moduły odpowiadają w dużej mierze docelowym klasom, które będą wykonywały te zadania.

3.5 Diagram przepływu danych



Rysunek 2: Diagram przepływu danych. Opisane sekwencje są odzwierciedlane przez projektowane API.

3.6 Model bazy danych



Rysunek 3: Schemat bazy danych

3.7 Harmonogram realizacji

- Implementacja modelu i kontrolera zwracającego mockowe dane (21.11.2023)
- Stworzenie warstwy dostępu do bazy danych i funkcjonalności tworzenia i usuwania obiektów (5.12.2023)
- Zaimplementowanie algorytmów obliczających rankingi wraz z testami (19.12.2023)
- Ulepszenie komponentu logowania użytkownika o lepsze bezpieczeństwo i zachowanie sesji (2.01.2024)