



**Politechnika
Śląska**

Dokumentacja projektowa

Zarządzanie Systemami Informatycznymi

*Zarządzanie projektem i koordynacja zespołu developerskiego
przy tworzeniu aplikacji webowej z użyciem Gita, Trello i
Dockera.*

Kierunek: Informatyka

Członkowie zespołu:
Paweł Lewandowski
Michał Kuta

Gliwice, 2024/2025

Spis treści

1	Wprowadzenie	2
1.1	Role w projekcie	2
1.2	Cel projektu	2
2	Założenia projektowe	3
2.1	Założenia techniczne i nietechniczne	3
2.1.1	Założenia techniczne	3
2.1.2	Założenia nietechniczne	3
2.2	Stos technologiczny	3
2.3	Oczekiwane rezultaty projektu	4
3	Realizacja projektu	5
3.1	Opis sprintów	5
3.2	Wykresy postępu projektu	8
4	Konkluzja	9
5	Bibliografia	10

1 Wprowadzenie

1.1 Role w projekcie

W naszym zespole odpowiedzialność została rozdzielona w następujący sposób:

- **Michał Kuta** – przygotowanie szkieletu projektu, implementacja wzorca MVC, testowanie.
- **Paweł Lewandowski** – organizacja pracy, zaprojektowanie bazy danych, stworzenie dokumentacji.

Podział obowiązków pozwolił na efektywną współpracę i realizację zadań zgodnie z harmonogramem.

1.2 Cel projektu

Celem projektu było stworzenie aplikacji umożliwiającej zapisywanie wyników rozgrywek gier planszowych. Projekt zakłada opracowanie funkcjonalnego narzędzia, które umożliwia:

- Dodawanie graczy.
- Dodawanie gier planszowych.
- Zapisywanie rozgrywek wraz z wynikami.
- Przeglądanie historii rozgrywek.
- Przeglądanie rankingu graczy.

Wszystko w wygodnym dla użytkownika interfejsie.

2 Założenia projektowe

2.1 Założenia techniczne i nietechniczne

2.1.1 Założenia techniczne

- Aplikacja webowa oparta na wzorcu projektowym MVC.
- Backend oraz logika aplikacji zaimplementowane w języku PHP.
- Aplikacja uruchamiana w kontenerze Docker.
- Warstwa prezentacji oparta na szablonach HTML/CSS.
- Komunikacja z bazą danych za pomocą warstwy modelu.
- Dane przechowywane w relacyjnej bazie danych MySQL.
- Możliwość lokalnego uruchomienia poprzez docker-compose.

2.1.2 Założenia nietechniczne

- Intuicyjny interfejs użytkownika dostosowany do komputerów i tabletów.
- Możliwość rejestrowania wyników wielu graczy w różnych grach.
- Funkcja przeglądania historii rozgrywek i statystyk.
- Możliwość łatwej rozbudowy o nowe funkcjonalności.
- Nacisk na prostotę obsługi i niezawodność.

2.2 Stos technologiczny

- Frontend: HTML, CSS.
- Backend: PHP z wykorzystaniem wzorca MVC.
- Baza danych: MySQL.
- System kontroli wersji: Git, GitHub.
- Zarządzanie: Trello.
- Konteneryzacja: Docker, Docker Compose.

2.3 Oczekiwane rezultaty projektu

- W pełni funkcjonalna aplikacja webowa umożliwiająca rejestrowanie wyników gier planszowych.
- Intuicyjny interfejs użytkownika umożliwiający łatwe dodawanie i przeglądanie rozgrywek.
- System zapisu danych z wykorzystaniem bazy danych (MySQL).
- Środowisko uruchomieniowe oparte na Dockerze.
- Dokumentacja techniczna i użytkowa aplikacji.

3 Realizacja projektu

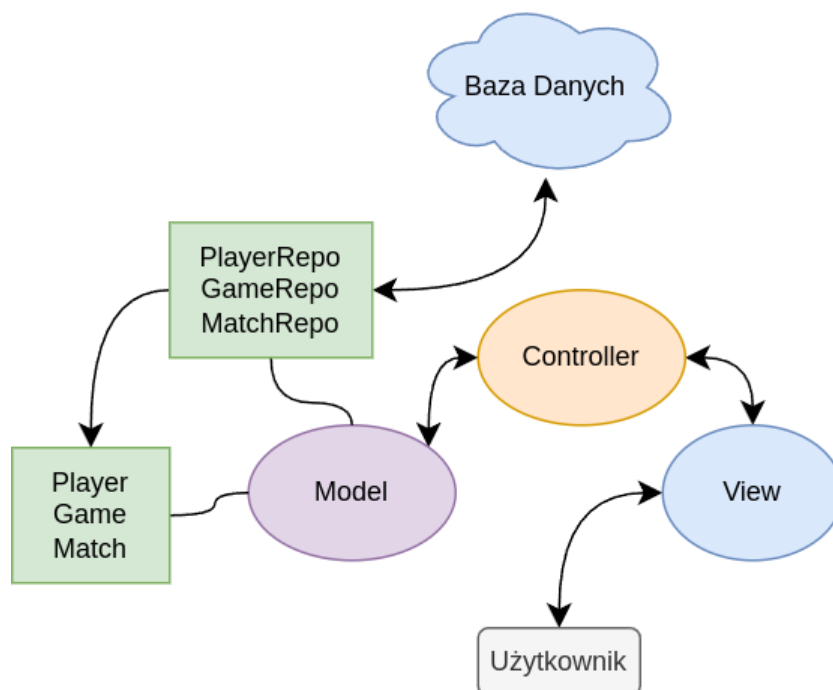
Nasz projekt wstępnie podzieliliśmy na 5 sprintów i zapisaliśmy je na naszej tablicy Trello.

3.1 Opis sprintów

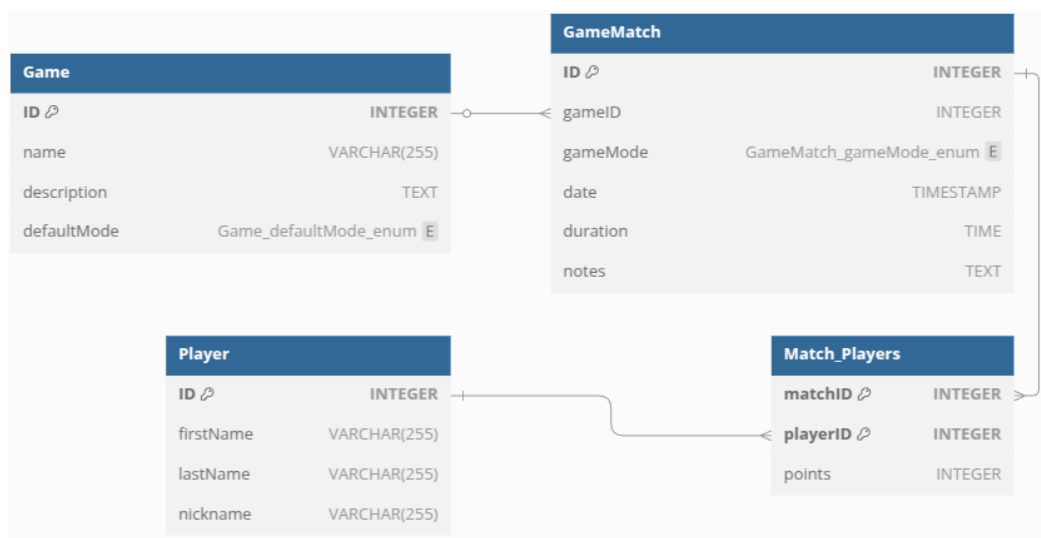
Projekt został zrealizowany w następujących etapach:

1. Sprint 1 – Planowanie:

- Identyfikacja funkcjonalności aplikacji (dodawanie gier, graczy, przeglądanie rekordów).
- Utworzenie tablicy Kanban w Trello do zarządzania zadaniami.
- Wybór stosu technologicznego: PHP + MySQL + Docker, z wykorzystaniem wzorca MVC.
- Zaplanowanie struktury katalogów i komponentów aplikacji.
- Opracowanie planu sprintów oraz roadmapy rozwoju.



Rysunek 1: Architektura MVC w naszym projekcie. Użytkownik ma jedynie kontakt z warstwą View która jest kontrolowana przez warstwę Controller. Controller natomiast korzysta z warstwy Model która jest warstwą danych, Controller je przetwarza i wyświetla w warstwie View.



Rysunek 2: Wspólnie zaprojektowana baza danych.

2. Sprint 2 – Tworzenie podstaw:

- Utworzenie środowiska kontenerowego Docker (PHP + MySQL).
- Implementacja szkieletu aplikacji zgodnie z wzorcem MVC.
- Stworzenie warstwy modelu i połączenia z bazą danych.
- Testowanie komunikacji z bazą MySQL za pomocą PDO.
- Przygotowanie roboczych widoków do testowego wyświetlania danych.

3. Sprint 3 – Główne programowanie:

- Implementacja logiki warstwy kontrolera.
- Zastąpienie widoków roboczych finalnymi szablonami HTML i CSS.
- Połączenie warstw modelu, widoku i kontrolera w spójną całość.
- Wprowadzenie mechanizmów obsługi błędów.

4. Sprint 4 – Testowanie:

- Intensywne testowanie aplikacji pod kątem poprawności działania.
- Poprawa wykrytych błędów i usprawnienie mechanizmów walidacji.
- Weryfikacja stabilności komunikacji między warstwami aplikacji.
- Testy środowiska Docker (np. restart kontenerów, poprawność konfiguracji).

5. Sprint 5 – Faza końcowa:

- Uzupełnienie i uporządkowanie dokumentacji technicznej projektu.
- Weryfikacja wykonania zadań z tablicy Kanban.
- Finalne poprawki w kodzie oraz przygotowanie aplikacji do prezentacji.

Stats: Kamil Mortadela "Zozo"

Back

Global Winrate
Matches Played: 3
Wins: 2
Winrate: 66.67%

Most Played Games

Game	Times Played
Szachy	2
Warcaby	1

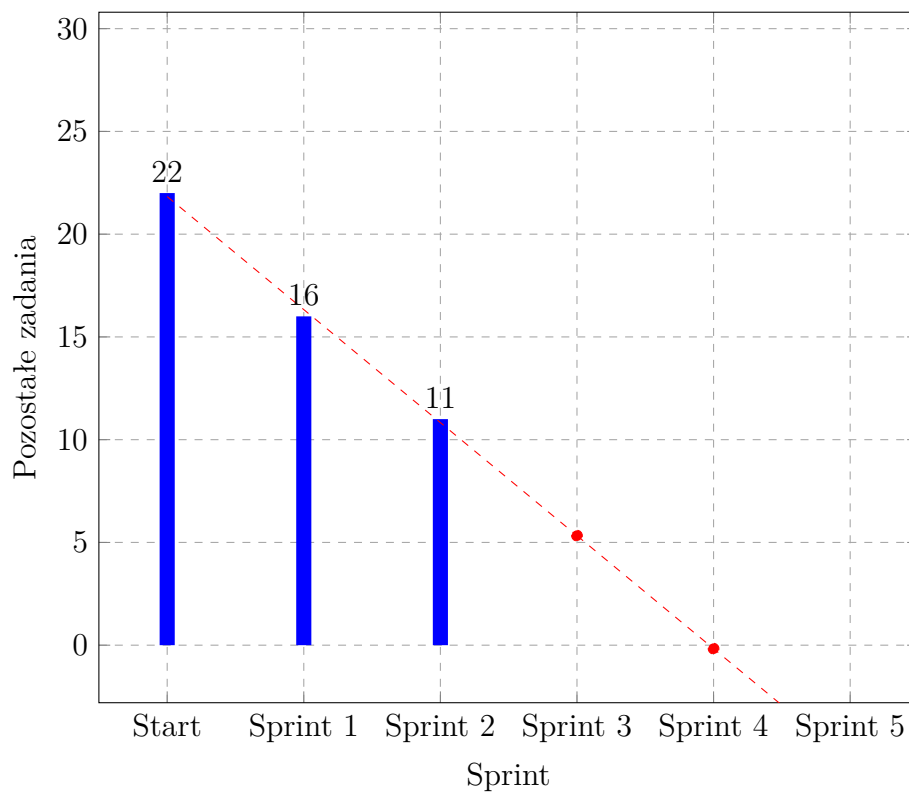
Best Winrate Games

Game	Games Played	Wins	Winrate
Warcaby	1	1	100%
Szachy	2	1	50%

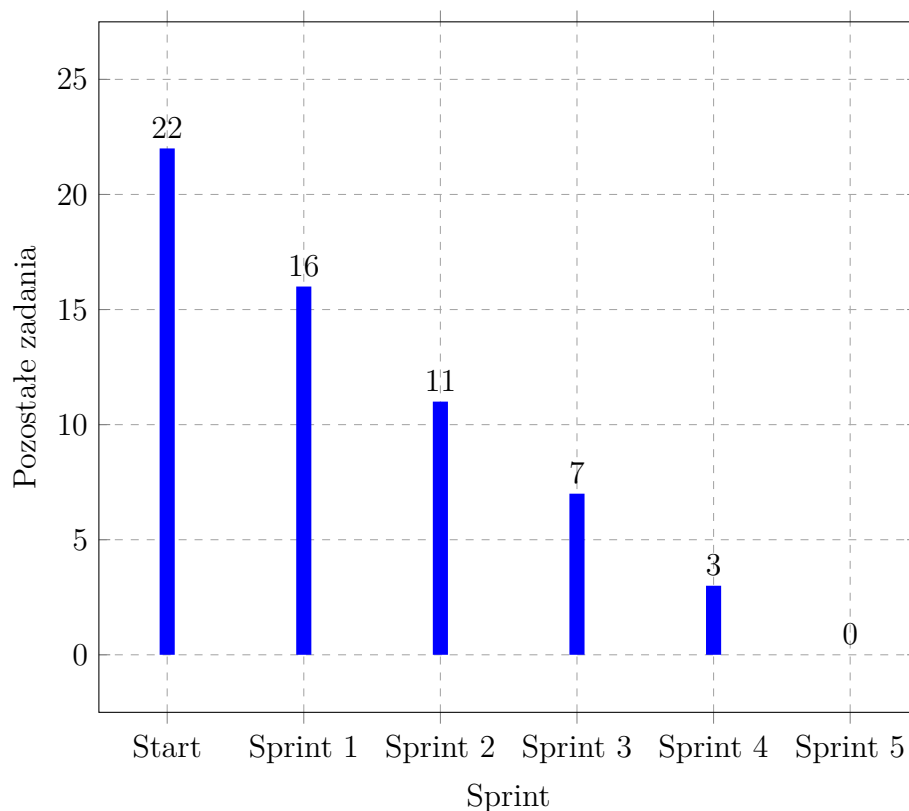
Rysunek 3: Zrzut ekranu aplikacji

3.2 Wykresy postępu projektu

Poniżej przedstawiono dwa wykresy ilustrujące postęp prac nad projektem w kolejnych sprintach.



Rysunek 4: Wykres regresji liniowej wykonanych zadań oraz prognoza zakończenia projektu. Linia trendu sugeruje, że projekt powinien zostać ukończony około sprintu 4.



Rysunek 5: Rzeczywisty przebieg liczby pozostałych zadań w kolejnych sprintach. Widać systematyczny postęp i redukcję liczby zadań aż do zera.

4 Konkluzja

Projekt został zrealizowany zgodnie z założeniami, dostarczając funkcjonalną aplikację webową do rejestrowania wyników gier planszowych. Wykorzystanie wzorca MVC, PHP, MySQL oraz konteneryzacji Docker zapewniło stabilność i skalowalność rozwiązania. Podział obowiązków w zespole oraz efektywne zarządzanie zadaniami za pomocą Trello pozwoliły na terminowe ukończenie wszystkich sprintów. Aplikacja spełnia wszystkie wymagania funkcjonalne i nietechniczne, oferując intuicyjny interfejs oraz możliwość łatwej rozbudowy w przyszłości.

5 Bibliografia

1. Oficjalna dokumentacja PHP – <https://www.php.net/docs.php>
2. Dokumentacja Docker – <https://docs.docker.com/>
3. Dokumentacja MySQL – <https://dev.mysql.com/doc/>
4. Artykuł: „Understanding MVC Architecture”
– <https://www.geeksforgeeks.org/mvc-design-pattern/>
5. Oficjalna dokumentacja PDO –
<https://www.php.net/manual/en/book.pdo.php>
6. Dokumentacja Git – <https://git-scm.com/doc>
7. Trello – <https://trello.com>
8. dbdiagram.io – <https://dbdiagram.io>