**UNIWERSYTET WARMIŃSKO-MAZURSKI W OLSZTYNIE**

**WYDZIAŁ MATEMATYKI I INFORMATYKI**

**ALEKSANDER ŻUK**

**Field of study: Information system engineering**

**Amateur league managment system**

**Praca inżynierska wykonana w Katedrze Metod Matematycznych Informatyki pod kierunkiem dra Pawła Drozdy**

**Olsztyn 2025**

**UNIVERSITY OF WARMIA AND MAZURY IN OLSZTYN FACULTY OF MATHEMATICS AND COMPUTER SCIENCE**

**ALEKSANDER ŻUK**

**Kierunek: Inżynieria systemów informatycznych**

**System zarządzania ligami amatorskimi**

**Engineering Thesis written in Department of Mathematical Methods in Computer Science under supervison of dr Paweł Drozda**

**Olsztyn 2025**

Spis treści

[Streszczenie 4](#_Toc187679178)

[Abstract 4](#_Toc187679179)

[1. Wstęp 4](#_Toc187679180)

[2. Wykorzystane technologie 4](#_Toc187679181)

[2.1 Backend 4](#_Toc187679182)

[2.1.1 Python 4](#_Toc187679183)

[2.1.2 Django 4](#_Toc187679184)

[2.1.3 Sqlite3 4](#_Toc187679185)

[2.2 Frontend 4](#_Toc187679186)

[2.2.1 HTML 4](#_Toc187679187)

[2.2.2 CSS 4](#_Toc187679188)

[2.2.3 JavaScript 4](#_Toc187679189)

[2.2.4 React 4](#_Toc187679190)

[2.2.5 Bootstrap 4](#_Toc187679191)

[3. Projekt techniczny systemu 4](#_Toc187679192)

[3.1 Wymagania funkcjonalne i niefunkcjonalne 4](#_Toc187679193)

[3.1.1 Wymagania funkcjonalne 4](#_Toc187679194)

[3.1.2 Wymagania niefunkcjonalne 4](#_Toc187679195)

[3.2 Schemat Bazy danych 4](#_Toc187679196)

[3.3 Diagramy przypadków użycia 4](#_Toc187679197)

[3.4 Scenariusze przypadków użycia 4](#_Toc187679198)

[4. Instrukcja użytkownika 4](#_Toc187679199)

[5. Podsumowanie 5](#_Toc187679200)

[6. Bibliografia 5](#_Toc187679201)

# 

# Streszczenie

# Abstract

# Wstęp

Piłka nożna jest jedną z najpopularniejszych dyscyplin sportowych na świecie - uprawiają ją miliony sportowców na świecie, a jeszcze więcej osób śledzi rozgrywki swoich ulubionych drużyn jako kibice. Można śmiało powiedzieć, że jest to najpopularniejszy i najbardziej angażujący sport na świecie. Co warte podkreślenia, nie ogranicza się wyłącznie do profesjonalnego poziomu klubowego i międzynarodowego; wielkim zainteresowaniem cieszą się również rozgrywki na niskich poziomach, a także amatorskie formy sportu takie jak chociażby mecze rozgrywane w mniej osób, czy na nietypowych boiskach, zasadach. Przyciąga to zainteresowanie zarówno początkujących graczy, wielkich miłośników sportu jak i wieloletnich weteranów piłkarskiego rzemiosła. W ostatnich latach coraz bardziej popularnymi stają się amatorskie rozgrywki szóstek, siódemek, czy dziewiątek piłkarskich, a część z nich nawet w formie halowej. Takie spotkania zapewniają zawodnikom nie tylko rozrywkę i możliwość zachowania aktywności fizycznej, lecz także element rywalizacji i budowania więzi społecznych w drużynie

Niemniej jednak organizacja amatorskich rozgrywek wymaga organizatorów ligi nierzadko sporo wysiłku. Rejestracja lig, tworzenie terminarza meczów, zarządzanie wynikami spotkań wraz z aktualizacją tabeli w konsekwencji może wprowadzać zamieszanie i liczne problemy natury logistycznej i administracyjnej. W przypadku większej liczby drużyn, z równoległymi ligami, a szczególnie, gdy liczba zespołów jest nieparzysta, ręczne tworzenie i aktualizowanie harmonogramów staje się procesem czasochłonnym i szczególnie podatnym na błędy ludzkie.

Z myślą o usprawnieniu tego procesu powstał niniejszy system zarządzania ligami amatorskimi. Projekt uwzględnia kluczowe i najważniejsze funkcjonalności takie jak: rejestracja użytkownika, stworzenie ligi oraz zespołu, przypisywanie drużyn do poszczególnych rozgrywek, dodawanie zawodników do zespołu, a także automatyczne generowanie terminarza meczów między zespołami w lidze. Wspomniane rozwiązanie pozwala również na śledzenie aktualnego stanu punktacji między zespołami w lidze dzięki automatycznie aktualizującej się tabeli, jako że administrator ligi ma możliwość edycji wyników spotkań.

System napisany został w oparciu o framework Django (język Python) po stronie serwerowej oraz framework React (język Javascript i HTML) z wykorzystaniem frameworka Bootstrap w warstwie interfejsu użytkownika. Django zapewnia wydajną komunikacją z bazą danych, natomiast React pozwala na budowę responsywnego, nowoczesnego i dynamicznego interfejsu, co przekłada się na czytelną i przyjemną w obsłudze stronę internetową. Dzięki kombinacji dwóch technologii projekt jest łatwy w rozbudowanie i modyfikacjach dzięki odseparowaniu ich do siebie.

Inspirację do stworzenia systemu stanowiło osobiste zainteresowanie sportem, a także zaangażowanie autora w amatorskie rozgrywki piłkarskie w Warszawie. Udział w lokalnej lidze szóstek piłkarskich w Warszawie - w „Lidze Fanów” - pozwolił lepiej zrozumieć potrzeby organizatorów i zawodników. Decyzja o wyborze tego tematu wynikała więc z pasji, co znacznie ułatwiło proces tworzenia aplikacji.

Dodatkowo, pomimo tego, że system jest w dużej mierze dopasowany do potrzeb piłki nożnej, można go w przyszłości częściowo zaadaptować na potrzeby innych dyscyplin sportowych. Jego elastyczność w zakresie rozwoju pozwala na wprowadzenie innych reguł punktowych właściwych dla danej dyscypliny. Na moment pisania niniejszej pracy aplikacja jest już gotowa do użytku - powstało w niej kilka przykładowych lig testowych celem zademonstrowania działania systemu.

Głównym celem jest zatem przedstawienie kompletnego procesu tworzenia i działani systemu, począwszy od opisu wymagań oraz doboru technologii, poprzez projekt architektury i bazy danych, aż po instrukcję obsługi dla użytkowników.

Dzięki temu praca stanowi nie tylko dokumentację techniczną projektu, ale i wskazówkę dla osób zainteresowanych rozwijaniem podobnych aplikacji. Stworzone narzędzie może na rynku wyróżniać się prostotą i intuicyjnością w obsłudze. Nowoczesny interfejs może decydować o tym jak użytkownik odbiera korzystanie z aplikacji.

Podsumowując, praca ma na celu zaprezentowanie systemu zarządzania ligami amatorskimi, który odpowiada na rzeczywiste potrzeby organizatorów i uczestników takich rozgrywek. Kolejne rozdziały wprowadzą czytelnika w szczegółowe zagadnienia techniczne oraz praktyczne.

# Wykorzystane technologie

## Backend

### Python

Python to wysokopoziomowy język programowania stworzony z myślą o czytelności kodu i szybkości tworzenia oprogramowania. Jego rozbudowany system bibliotek oraz prosty składniowo kod sprawiają, że jest często wybierany do tworzenia aplikacji webowych, analizy danych, czy uczenia maszynowego. W projekcie został wykorzystany do napisania logiki po stronie serwera oraz do implementacji frameworka Django.

### Django

Django jest popularnym frameworkiem do tworzenia aplikacji webowych napisanym w języku Python. Framework oferuje zintegorwany panel administracyjny, system obsługi uwierzytelniania, zarządzania sesjami, a także rozbudowane narzędzia do generowanie URL i widoków. Oparłem swoją pracę na tym narzędziu ze względu na prostotę użytkowania języka Python, jak i samego frameworka.

### Sqlite3

SQLite3 jest niewielką, plikową bazą danych, które nie wymaga instalacji serwera bazodanowego. Jest lekka i wydajna w przypadku mniejszych projektów i aplikacji. W kontekście aplikacji Django stanowi często domyślne rozwiązanie. Jej prostota konfiguracji i brak szczególnie istotnych dla mnie wad zdecydował o pozostawieniu jej w projekcie po domyślnym ustawieniu frameworka Django.

## Frontend

### HTML

HTML to podstawowy język znaczników używany do tworzenia struktury stron internetowych. Definiuje takie elementy jak nagłówki, akapity, obrazy, czy hiperłącza. W projekcie, mimo że interfejs jest tworzony za pomocą React i plikuw typu .jsx, to HTML pozostaje kluczową podstawą, jako że to na jego składni opiero się koncepcja JSX (JavaScript XML), która wprowadza do kodu JavaScript znacznikową formę zbliżoną do HTML.

### CSS

CSS to język służący do opisywania wyglądu i formatowania elementów języka HTML. Dzięki niemu można kontrolować między innymi kolory, czcionki, rozmiary, czy położenie poszczególnych bloków na stronie. W projekcie wykorzystuję CSS zarówno w postaci własnych arkuszy stylów, jak i w połączeniu z gotowymi komponentami (przykładowo frameworkiem Bootstrap), aby zapewnić spójny wygląd aplikacji.

### JavaScript

JavaScript jest językiem skryptowym, który umożliwia dynamicznie modyfikować zawartość i wygląd strony internetowej po stronie klienta. Współpracuje z HTML i CSS, nadając interaktywność elementom w przeglądarce. Jest czynnikiem odpowiadającym za reagowanie na zdarzenia wywoływane przez użytkownika takie jak kliknięcie, wpisywanie tekstu, a następnie aktualizowanie fragmentów strony bez konieczności przeładowywania całego dokumentu. W projekcie stanowi istotną część warstwy frontendowej, jako że na tym języku opiera się React, z którego korzystałem.

### React

Jest to biblioteka języka JavaScript opracowana przez Facebook. Kluczową zaletą React jest podejście oparte na komponentach, czyli reużywalnych elementach interfejsu, które pomagają w programowaniu zgodnie z konwencją unikania redundancji kodu. React korzysta z Virtual DOM, co optymalizuje aktualizacje w przeglądarce i czyni aplikacje szybszą w działaniu. W projekcie React obsługuje logikę przedstawiania danych uzyskanych w wyniku komunikacji z backendem.

### Bootstrap

Bootstrap to popularny framework CSS, który ułatwia tworzenie responsywnych i estetycznych interfejsów graficznych. Jego zaletą jest to, że zawiera gotowe style i komponenty (przyciski, formularze), co pozwala szybko zbudować spójny wygląd aplikacji. W połączeniu z React i HTML, Bootstrap umożliwia tworzenie nowoczesnych przyjaznych dla użytkownika stron, dobrze dopasowanych do urządzeń mobilnych. Łatwość użytkowania i oszczędność czasu i miejsca w kodzie zdecydowała o użyciu tego frameworka w projekcie.

# Projekt techniczny systemu

## Wymagania funkcjonalne i niefunkcjonalne

### Wymagania funkcjonalne

* System powinien umożliwiać rejestrację i logowanie użytkowników
* System powinien umożliwiać tworzenie zespołu
* System powinien umożliwiać tworzenie ligi
* System powinien umożliwiać organizatorowi przypisywać zespoły do ligi
* System powinien umożliwiać właścicielowi przypisywać zawodników do zespołu
* System powinien umożliwiać generowania struktury terminarza meczów między zespołami w lidze
* System powinien umożliwiać organizatorowi ligi edytowania wyników spotkań między zespołami
* System powinien umożliwiać organizatorowi usunięcie terminarzu i samej ligi
* System powinien generować aktualną tabelę dla każdej ligi
* System powinien umożliwiać wyszukiwanie i filtrowanie lig po nazwie

### Wymagania niefunkcjonalne

* Interfejs powinien być przyjazny i intuicyjny dla użytkownika
* System powinien zapewniać bezpieczeństwo
* System powinien być skalowalny
* System powinien być niezawodny
* System powinien być łatwy do wdrożenia

## Schemat Bazy danych

## Diagramy przypadków użycia

## Scenariusze przypadków użycia

# Instrukcja użytkownika

# Podsumowanie

# Bibliografia