# Społeczna Akademia Nauk

Kierunek studiów**:**

**Informatyka**

**Igor Gralak**

Numer albumu: 122548

# Aplikacja Webowa do Zarządzania Klubem Fitness – Implementacja Frontendowa w Angular oraz Backendowa w Spring Boot

Praca inżynierska napisana

pod kierunkiem Dr. Inż. Konrada Grzanka

**Rok 2024**

Spis treści

[Społeczna Akademia Nauk 1](#_Toc183117465)

[Aplikacja Webowa do Zarządzania Klubem Fitness – Implementacja Frontendowa w Angular oraz Backendowa w Spring Boot 1](#_Toc183117466)

[Streszczenie 3](#_Toc183117467)

[Abstrakt 3](#_Toc183117468)

[1. Wstęp 3](#_Toc183117469)

[1.1. Cel i zakres pracy 3](#_Toc183117470)

[1.2. Grono odbiorców 4](#_Toc183117471)

[1.3. Struktura pracy 4](#_Toc183117472)

[2. Kryteria projektowe 5](#_Toc183117473)

[2.1. Wprowadzenie 5](#_Toc183117474)

[2.2. Aktorzy 5](#_Toc183117475)

[2.3. Wymagania funkcjonalne 7](#_Toc183117476)

[2.4. Wymagania niefunkcjonalne 7](#_Toc183117477)

[2.5. Wytyczne dotyczące technologii 8](#_Toc183117478)

[3. Istniejące rozwiązania 8](#_Toc183117479)

[3.1. Wprowadzenie 8](#_Toc183117480)

[3.2. Przegląd dostępnych systemów i narzędzi 8](#_Toc183117481)

[3.3. Analiza rozwiązań konkurencyjnych 8](#_Toc183117482)

[4. Wytyczne projektowe 8](#_Toc183117483)

[4.1. Wprowadzenie 8](#_Toc183117484)

[4.2. Propozycje rozwiązań 8](#_Toc183117485)

[4.3. Oczekiwane rezultaty i ograniczenia 8](#_Toc183117486)

[5. Część praktyczna 8](#_Toc183117487)

[5.1. Wprowadzenie 8](#_Toc183117488)

[5.2. Implementacja systemu 8](#_Toc183117489)

[5.3. Wykorzystanie bazy danych 8](#_Toc183117490)

[5.4. Diagram bazy danych 8](#_Toc183117491)

[5.5. Widoki aplikacji 8](#_Toc183117492)

[5.6. Testy i weryfikacja 8](#_Toc183117493)

[6. Podsumowanie 8](#_Toc183117494)

[6.1. Wprowadzenie 8](#_Toc183117495)

[6.2. Wnioski końcowe 8](#_Toc183117496)

[6.3. Możliwości dalszego rozwoju 8](#_Toc183117497)

[Spis ilustracji 8](#_Toc183117498)

[Spis tabel 9](#_Toc183117499)

# Streszczenie

# Abstrakt

# Wstęp

## Cel i zakres pracy

**Celem niniejszej pracy** jest opracowanie projektu oraz implementacja systemu informatycznego, który usprawni procesy zarządzania siecią klubów fitness. System będzie podzielony na trzy główne moduły, z których każdy pełni określoną rolę:

* **Moduł aplikacji webowej** – Główne zadanie tego modułu polega na dostarczaniu intuicyjnego interfejsu użytkownika, który umożliwi wizualizację oraz przetwarzanie danych. Aplikacja webowa, zbudowana przy użyciu technologii frontendowych, pozwoli na łatwą obsługę przez personel oraz klientów klubu.
* **Moduł backendowy** – Jest to kluczowy element odpowiedzialny za logikę biznesową systemu oraz integrację danych. Backend obsługuje zapytania pochodzące z aplikacji webowej oraz przeprowadza procesowanie danych, które mogą obejmować walidację, przetwarzanie wstępne oraz formatowanie. Moduł backendowy zarządza przepływem informacji między aplikacją a bazą danych, zapewniając bezpieczeństwo oraz spójność danych.
* **Moduł bazy danych** – Trzeci moduł to baza danych, w której przechowywane są wszystkie informacje związane z funkcjonowaniem sieci klubów fitness. Baza danych, przechowuje dane użytkowników, informacje o rezerwacjach, historii członkostwa, dostępnych usługach oraz harmonogramach zajęć. Struktura bazy danych jest zaprojektowana w sposób, który umożliwia łatwe rozszerzanie i integrację z nowymi modułami oraz zapewnia wysoką wydajność i bezpieczeństwo danych.

System oparty na tych trzech modułach będzie wspierał codzienne operacje klubu, usprawniając komunikację i zarządzanie danymi oraz podnosząc jakość obsługi klientów.

## Grono odbiorców

Adresatami niniejszego systemu są przede wszystkim osoby związane ze światem sportu i fitnessu, w tym pracownicy klubów fitness, trenerzy personalni, a także klienci zainteresowani usługami oferowanymi przez kluby. Aplikacja jest skierowana do szerokiego grona odbiorców, w tym zarówno do osób o różnym poziomie zaawansowania technologicznego. Głównym celem jest zapewnienie intuicyjnego interfejsu, który umożliwi efektywne korzystanie z systemu bez potrzeby posiadania zaawansowanej wiedzy technicznej. System został zaprojektowany tak, aby był przystępny w użyciu dla każdego użytkownika, niezależnie od jego umiejętności komputerowych, co pozwala na łatwą obsługę i szybkie przyswajanie funkcjonalności, szczególnie przez osoby niemające wcześniejszych doświadczeń z tego typu oprogramowaniem.

## Struktura pracy

W rozdziale pierwszym przedstawiono wstęp, cel oraz zakres pracy, a także ogólną strukturę. Rozdział drugi poświęcony jest kryteriom projektowym, w tym szczegółowym wymaganiom funkcjonalnym i niefunkcjonalnym oraz wytycznym technologicznym, które kierują implementacją systemu. Rozdział trzeci opisuje istniejące rozwiązania na rynku, analizuje dostępne systemy i narzędzia oraz porównuje je z projektowanym systemem, aby wskazać potencjalne przewagi i unikalne cechy aplikacji. Czwarty rozdział dotyczy wytycznych projektowych, omawia propozycje rozwiązań, oczekiwane rezultaty, a także ograniczenia napotkane podczas projektowania. Piąty rozdział obejmuje część praktyczną pracy, w której znajduje się szczegółowy opis implementacji systemu, w tym wdrożenie bazy danych, diagram bazy danych oraz kluczowe widoki aplikacji. Rozdział ten uwzględnia również proces testowania i weryfikacji działania aplikacji, co pozwala na ocenę jej wydajności i zgodności z wymaganiami. W szóstym rozdziale zamieszczono podsumowanie oraz końcowe wnioski wynikające z przeprowadzonej pracy, jak również omówiono możliwości dalszego rozwoju systemu. Na końcu pracy znajduje się lista ilustracji i tabel, które ułatwiają przeglądanie treści oraz pozwalają na szybsze zrozumienie kluczowych informacji. Dodatek A zawiera instrukcje dotyczące instalacji i uruchomienia aplikacji, listę scenariuszy testowych oraz słownik użytych akronimów..

# Kryteria projektowe

## Wprowadzenie

Lorem ipsum

## Aktorzy

W aplikacji zdefiniowano cztery typy aktorów, różniących się zakresem dostępnych funkcji oraz poziomem uprawnień:

**Gość**  
Gość to użytkownik, który nie posiada jeszcze konta lub nie jest zalogowany w aplikacji. Ma dostęp do ograniczonego zakresu funkcji, które pozwalają mu zapoznać się z podstawowymi informacjami o klubie fitness. Do funkcji dostępnych dla Gościa należą:

* Przeglądanie strony głównej klubu fitness, aby poznać ofertę, godziny otwarcia, lokalizacje i inne ogólne informacje.
* Wypełnienie i wysyłanie formularzy związanych z logowaniem, rejestracją konta oraz przypomnieniem hasła.

Zalogowany użytkownik

Zalogowany użytkownik to osoba, która po rejestracji konta zalogowała się w aplikacji. Uzyskuje ona dodatkowe możliwości w stosunku do Gościa, dzięki czemu może zarządzać swoimi rezerwacjami i członkostwami. Funkcje dostępne dla zalogowanego użytkownika obejmują:

* Wszystkie funkcje Gościa.
* Przeglądanie dostępnych treningów oraz członkostw oferowanych przez klub fitness.
* Dołączanie do wybranych treningów oraz możliwość opuszczenia ich.
* Zakup członkostwa w klubie oraz rezygnację z aktywnego członkostwa.
* Generowanie kodu QR zawierającego dane użytkownika, który może służyć np. do identyfikacji przy wejściu do klubu lub na trening.

Trener

Trener jest szczególnym przypadkiem zalogowanego użytkownika, który posiada dodatkowe uprawnienia związane z prowadzeniem zajęć oraz zarządzaniem uczestnikami swoich treningów. Do jego funkcji należą:

* Wszystkie funkcje dostępne dla zalogowanego użytkownika.
* Przeglądanie listy użytkowników przypisanych do treningów, które prowadzi.
* Dodawanie nowych treningów do oferty klubu.
* Wyświetlanie listy użytkowników, którzy dołączyli do poszczególnych treningów prowadzonych przez trenera.
* Potwierdzanie obecności użytkowników na treningach oraz zaznaczanie ich nieobecności.

Administrator

Administrator to najwyżej uprzywilejowany aktor aplikacji, który dziedziczy wszystkie funkcjonalności Trenera oraz posiada dodatkowe uprawnienia administracyjne. Administrator może zarządzać zasobami systemu, w tym użytkownikami oraz członkostwami, aby zapewnić płynne działanie aplikacji i aktualizację danych. Funkcje dostępne dla Administratora obejmują:

* Wszystkie funkcje dostępne dla Trenera.
* Dodawanie, usuwanie oraz edytowanie kont użytkowników.
* Zarządzanie członkostwami, w tym dodawanie nowych rodzajów członkostw, edytowanie ich szczegółów oraz usuwanie istniejących członkostw.

## Wymagania funkcjonalne

1. Zarządzanie użytkownikami:
   1. Możliwość rejestracji i logowania użytkowników.
   2. Weryfikacja danych użytkowników przy logowaniu.
   3. Obsługa różnych ról użytkowników, takich jak administrator, trener i klient.
2. Zarządzanie członkostwami:
   1. Tworzenie, edytowanie i usuwanie członkostw.
   2. Przypisywanie członkostw do klientów.
   3. Automatyczne powiadamianie o wygasaniu członkostwa.
3. Zarządzanie zajęciami fitness:
   1. Dodawanie i edytowanie terminów zajęć fitness.
   2. Możliwość rezerwacji zajęć przez klientów.
   3. Wyświetlanie listy zapisanych uczestników dla danego terminu.
4. Zarządzanie harmonogramem:
   1. Generowanie kalendarza zajęć dostępnych w klubie fitness.
   2. Możliwość filtrowania zajęć według kategorii, dnia tygodnia i prowadzącego.
5. Zarządzanie pracownikami:
   1. Dodawanie, edytowanie i usuwanie profili pracowników, takich jak trenerzy.
   2. Przydzielanie zajęć konkretnym trenerom.
6. System płatności:
   1. Obsługa płatności za członkostwa i zajęcia.
   2. Generowanie historii transakcji dla użytkownika.
   3. Powiadomienia o zaległościach płatniczych.
7. Statystyki i raporty:
   1. Generowanie raportów dotyczących liczby użytkowników, zajęć i przychodów.
   2. Wyświetlanie statystyk dotyczących frekwencji na zajęciach.
8. Panel administracyjny:
   1. Dedykowany panel zarządzania dla administratora systemu.
   2. Możliwość zarządzania użytkownikami, członkostwami, harmonogramem i płatnościami.
9. Integracja z zewnętrznymi usługami:
   1. Możliwość wysyłania powiadomień e-mail lub SMS do użytkowników.
   2. Integracja z systemami płatności online.
10. Responsywność interfejsu użytkownika:
    1. Obsługa różnych urządzeń (komputery, tablety, smartfony).
    2. Intuicyjna nawigacja dostosowana do potrzeb użytkowników.
11. Bezpieczeństwo danych:
    1. Szyfrowanie danych użytkowników w bazie danych.
    2. Wdrożenie uwierzytelniania i autoryzacji dostępu do danych.
12. Wyszukiwanie i filtrowanie:
    1. Możliwość wyszukiwania użytkowników, członkostw i zajęć według różnych kryteriów.
    2. Obsługa dynamicznego filtrowania wyników w czasie rzeczywistym.
13. Historia aktywności:
    1. Przechowywanie historii rezerwacji, płatności i logowania dla każdego użytkownika.
    2. Wyświetlanie historii aktywności w panelu użytkownika.
14. Personalizacja:
    1. Możliwość edytowania profilu użytkownika, w tym danych kontaktowych i zdjęcia profilowego.
    2. Rekomendacje zajęć na podstawie historii uczestnictwa.

## Wymagania niefunkcjonalne

* Architektura warstwowa:
  + System powinien składać się z trzech głównych warstw:
    - Serwer bazy danych – obsługujący przechowywanie i zarządzanie danymi aplikacji w sposób transakcyjny (PostgreSQL).
    - Serwer aplikacji – obsługujący logikę biznesową i komunikację z warstwą bazy danych oraz interfejsem użytkownika (Spring Boot w backendzie).
    - Interfejs użytkownika – dostępny w przeglądarce internetowej, zbudowany z wykorzystaniem technologii Angular (SPA).
* Format przesyłanych danych:
  + Dane przesyłane między warstwami (frontend–backend, backend–baza danych) powinny być reprezentowane w formacie JSON, co zapewni kompatybilność i łatwą integrację z zewnętrznymi systemami.
  + Komunikacja z API powinna być realizowana za pomocą protokołu RESTful, aby uprościć procesy wymiany danych.
* Bezpieczeństwo i autoryzacja:
  + Dostęp do zasobów aplikacji musi być zabezpieczony mechanizmem autoryzacji:
  + JWT (JSON Web Tokens) powinien być używany do uwierzytelniania użytkowników.
  + Tylko uprawnieni użytkownicy z odpowiednimi rolami (np. administrator, trener, klient) powinni mieć dostęp do określonych funkcji systemu.
  + Widoczność i możliwość rezerwacji treningów powinna być ograniczona wyłącznie do użytkowników posiadających odpowiednie członkostwo.
* Responsywność i kompatybilność:
  + Interfejs użytkownika powinien być w pełni responsywny, umożliwiając dostosowanie układu do różnych rozdzielczości ekranów:
    - Urządzenia mobilne: łatwy dostęp do najważniejszych funkcji systemu z ekranów o przekątnej od 4 cali.
    - Tablety i laptopy: zoptymalizowane wyświetlanie z wykorzystaniem większej przestrzeni ekranu.
  + System musi działać poprawnie na najpopularniejszych przeglądarkach, takich jak Google Chrome, Mozilla Firefox, Microsoft Edge i Safari.
* Aplikacja jednostronicowa (SPA):
  + Interfejs użytkownika powinien być zbudowany jako aplikacja jednostronicowa (Single Page Application), zapewniająca płynne przechodzenie między sekcjami bez konieczności przeładowywania całej strony.
  + Dane z serwera powinny być pobierane dynamicznie przy użyciu technologii takich jak AJAX lub HTTP Client Angulara, co poprawi doświadczenie użytkownika.
* Wydajność i czas ładowania:
  + Całość interfejsu powinna być gotowa do interakcji w czasie nieprzekraczającym 3 sekund przy pierwszym ładowaniu strony.
  + Backend powinien być zoptymalizowany pod kątem obsługi wielu równoczesnych zapytań bez spadku wydajności – skalowalność aplikacji musi zapewniać obsługę co najmniej 500 jednoczesnych użytkowników.
* Obsługa sesji użytkowników:
  + Mechanizm sesji powinien umożliwiać automatyczne wylogowanie użytkownika po 30 minutach nieaktywności.
* Elastyczność konfiguracji:
  + Administrator systemu powinien mieć możliwość zmiany parametrów, takich jak:
    - Limity uczestników treningów.
    - Stawki za członkostwo.
    - Godziny działania klubu.
* Rozszerzalność:
  + Kod aplikacji powinien być napisany w sposób modułowy, umożliwiający łatwe dodawanie nowych funkcji, takich jak:
  + Programy lojalnościowe dla klientów.
  + Integracja z platformami płatniczymi (np. PayPal, Stripe).
* Zarządzanie czasem rzeczywistym:
  + Mechanizm synchronizacji czasu serwera z frontendem musi zapewniać dokładność co do sekundy.

## Wytyczne dotyczące technologii

System powinien być zaimplementowany w architekturze trójwarstwowej, obejmującej warstwę backendu, frontendową oraz bazę danych. Backend powinien być napisany w języku Java, z wykorzystaniem frameworka Spring Boot (w wersji 3.2.5), który zapewnia modularność, łatwość konfiguracji i wsparcie dla aplikacji webowych. Dla komunikacji z bazą danych zaleca się użycie Spring Data JPA z implementacją Hibernate, co umożliwi efektywne zarządzanie relacyjną bazą danych PostgreSQL. PostgreSQL, jako baza danych, zapewnia stabilność i wsparcie dla zaawansowanych operacji SQL, które są niezbędne w systemie o takiej funkcjonalności.

# Istniejące rozwiązania

## Wprowadzenie

## Przegląd dostępnych systemów i narzędzi

## Analiza rozwiązań konkurencyjnych

# Wytyczne projektowe

## Wprowadzenie

## Propozycje rozwiązań

## Oczekiwane rezultaty i ograniczenia

# Część praktyczna

## Wprowadzenie

## Implementacja systemu

## Wykorzystanie bazy danych

## Diagram bazy danych

## Widoki aplikacji

## Testy i weryfikacja

# Podsumowanie

## Wprowadzenie

## Wnioski końcowe

## Możliwości dalszego rozwoju

# Spis ilustracji

# Spis tabel