Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat: **Klub Sportów Walki „Lowkick”**

Autorzy: **Vladislav Zosimchuk, Kacper Wróbel, Jakub Gozdek**

Grupa: I1-223A

Kierunek: informatyka

Rok akademicki: 2023/24

Poziom i semestr: I/4

Tryb studiów: stacjonarne

# Spis treści

Spis treści

[1. Spis treści 1](#_Toc166053141)

[2. Odnośniki do innych źródeł 3](#_Toc166053142)

[3. Słownik pojęć 3](#_Toc166053143)

[4. Wprowadzenie 3](#_Toc166053144)

[4.1 Cel dokumentacji 3](#_Toc166053145)

[4.2 Przeznaczenie dokumentacji 4](#_Toc166053146)

[4.3 Opis organizacji lub analiza rynku 4](#_Toc166053147)

[4.4 Analiza SWOT organizacji 5](#_Toc166053148)

[5. Specyfikacja wymagań 5](#_Toc166053149)

[5.1 Charakterystyka ogólna 5](#_Toc166053150)

[5.1.1 Definicja produktu 5](#_Toc166053151)

[5.1.2 Podstawowe założenia 5](#_Toc166053152)

[5.1.3 Cel biznesowy 6](#_Toc166053153)

[5.1.4 Użytkownicy 6](#_Toc166053154)

[5.1.5 Korzyści z systemu 6](#_Toc166053155)

[5.1.6 Ograniczenia projektowe i wdrożeniowe 6](#_Toc166053156)

[5.2 Wymagania funkcjonalne 6](#_Toc166053157)

[5.2.1 Lista wymagań 6](#_Toc166053158)

[5.2.2 Diagramy przypadków użycia 7](#_Toc166053159)

[5.2.3 Szczegółowy opis wymagań 9](#_Toc166053160)

[5.3 Wymagania niefunkcjonalne 15](#_Toc166053161)

[6. Zarządzanie projektem 16](#_Toc166053162)

[6.1 Zasoby ludzkie 16](#_Toc166053163)

[6.2 Harmonogram prac 16](#_Toc166053164)

[6.3 Etapy/kamienie milowe projektu 16](#_Toc166053165)

[7. Zarządzanie ryzykiem 16](#_Toc166053166)

[7.1 Lista czynników ryzyka 16](#_Toc166053167)

[7.2 Ocena ryzyka 16](#_Toc166053168)

[7.3 Plan reakcji na ryzyko 17](#_Toc166053169)

[8. Zarządzanie jakością 17](#_Toc166053170)

[8.1 Scenariusze i przypadki testowe 17](#_Toc166053171)

[9. Projekt techniczny 17](#_Toc166053172)

[9.1 Opis architektury systemu 17](#_Toc166053173)

[9.2 Technologie implementacji systemu 19](#_Toc166053174)

[9.3 Diagramy UML 19](#_Toc166053175)

[9.3.1 Diagram(-y) klas 20](#_Toc166053176)

[9.3.2 Diagram(-y) czynności 20](#_Toc166053177)

[9.3.3 Diagramy sekwencji 21](#_Toc166053178)

[9.3.4 Inne diagramy 25](#_Toc166053179)

[9.4 Charakterystyka zastosowanych wzorców projektowych 26](#_Toc166053180)

[9.5 Projekt bazy danych 26](#_Toc166053181)

[9.5.1 Schemat 26](#_Toc166053182)

[9.5.2 Projekty szczegółowe tabel 27](#_Toc166053183)

[9.6 Projekt interfejsu użytkownika 27](#_Toc166053184)

[9.6.1 Lista głównych elementów interfejsu 27](#_Toc166053185)

[9.6.2 Przejścia między głównymi elementami 27](#_Toc166053186)

[9.6.3 Projekty szczegółowe poszczególnych elementów 29](#_Toc166053187)

[9.7 Procedura wdrożenia 35](#_Toc166053188)

[10. Dokumentacja dla użytkownika 35](#_Toc166053189)

[11. Podsumowanie 35](#_Toc166053190)

[Szczegółowe nakłady projektowe członków zespołu 35](#_Toc166053191)

[12. Inne informacje 36](#_Toc166053192)

# Odnośniki do innych źródeł

tj. do wykorzystywanych narzędzi / projektów w tych narzędziach

* + [https://trello.com/b/u1g032hK/in%C5%BCyneria-opragromowania](https://trello.com/b/u1g032hK/inżyneria-opragromowania)
  + <https://github.com/Kapikek/In-ynieria-oprogramowania---praca-zespo-owa.git>

# Słownik pojęć

Organizacja – tj. Klub Sportów Walki

# Wprowadzenie

## 4.1 Cel dokumentacji

Celem tej dokumentacji jest przedstawienie systemu projektowanego na rzecz KSW „Lowkick”. Zawiera ona:

-Specyfikację wymagań

-Projekt architektury systemu

-Dokumentację analityczną

-Dokumentację użytkownika(?)

## 4.2 Przeznaczenie dokumentacji

1. Dla programistów i projektantów systemu informatycznego dla organizacji KSW „Lowkick”
2. Dla właściciela i kierownika KSW „Lowkick”

## 4.3 Opis organizacji lub analiza rynku

KSW „Lowkick” jest klubem sportowym z wieloletnią tradycją, zespołem doświadczonych trenerów i wieloma osiągnieciami sportowymi. W ramach swoich usług oferuje szereg sekcji do treningów grupowych oraz możliwość wykupienia zajęć indywidualnych. W chwili obecnej klub posiada jedną salę treningową, wyposażoną w najnowszy sprzęt służący poprawie kondycji fizycznej oraz specjalistyczne narzędzia treningowe z myślą o sportach walki.

Klient może dołączyć „z ulicy” bądź drogą internetową, zapisać się na jeden z kilku kursów dla początkujących bez żadnych wstępnych wymagań. Istnieje możliwość pożyczenia podstawowego sprzętu treningowego (rękawic/ochraniaczy) z zaplecza organizacji, oczekuje się jednak, że klienci będą wyposażeni we własny sprzęt. Organizacja oferuje 5 różnych kursów, z czego 3 są kierowane do osób początkujących w 3 grupach wiekowych (do 13lat, 13-17, 17+) , a pozostałe dwa są dla osób bardziej doświadczonych (grupa nr.4) oraz przygotowujących się do walk i zawodów (grupa nr. 5).

Grupa nr 4 jest dostępna dla starszych klientów (17+), którzy przekraczają umiejętnościami zakres zajęć grupy nr 3. O przejściu decydują wspólnie klient oraz trener grupy nr 3. Grupa nr 5 jest również dla osób z najstarszej grupy (17+) i o przejściu do niej decydują wspólnie klient zainteresowany zawodami oraz trener grupy nr 4. Każda grupa składa się maksymalnie z 30 osób. Organizacja ma w swoim zespole 5 trenerów, każdy zajmuje się wyłącznie swoją grupą. W wyjątkowych przypadkach trenerzy zamieniają się grupami.

Treningi indywidulne są możliwe do zakupienia osobiście u trenerów oraz za pośrednictwem strony internetowej (niżej opisanego systemu). Godziny oraz cena takich treningów są ustalane indywidualnie, istnieje możliwość, by kilku trenerów robiło treningi indywidualne jednocześnie na sali treningowej organizacji. Trenerzy osobiście wprowadzają informacje o takich treningach w grafik na stronie organizacji, te treningi, które zostały zarezerwowane internetowo, muszą być zatwierdzone przez trenera. Płatności są realizowane post-factum lub z wyprzedzeniem, zależnie od potrzeb klienta, strona nie wymaga płatności do ustalenia terminu zajęć, jest to realizowanie na zasadzie wystawionego rachunku.

Treningi grupowe nie są realizowane na zasadzie kursu, tematyka zajęć jest przerabiana cyklicznie, okazjonalnie modyfikowana na rzecz poszerzania kompetencji bardziej doświadczonych klientów.

Główne usługi:

-Treningi grupowe (organizowane według stałego porządku)

-Zajęcia indywidualne (ustalane osobiście z trenerami)

## 4.4 Analiza SWOT organizacji

|  |  |
| --- | --- |
| **Mocne strony** | **Słabe strony** |
| Wyszkolona kadra trenerów  Dobrze wyposażony lokal  Silne poczucie społeczności  Sukcesy sportowe | Brak różnorodności oferty  Ograniczone środki finansowe  Niska świadomość marki |
| **Szanse** | **Zagrożenia** |
| Rozwój programów dla różnych grup wiekowych (junior/senior)  Współpraca z lokalnymi szkołami  Organizacja wydarzeń i turniejów | Konkurencja  Zmiany regulacyjne  Zmiany trendów zdrowotnych i sportowych |

# Specyfikacja wymagań

## Charakterystyka ogólna

### Definicja produktu

Platforma „kswlowkick.pl” umożliwiająca rejestrację członków, zarządzanie harmonogramem treningów, udostępnianie materiałów szkoleniowych, organizację wydarzeń oraz możliwość komunikacji i interakcji pomiędzy członkami klubu.

### Podstawowe założenia

System umożliwia:

1. Dla Kierownika

-Dodawanie i usuwanie trenerów

-Dodawanie i usuwanie kursów grupowych

2. Dla Trenerów

-Dodawanie i usuwanie kursów indywidualnych

-Kontakt z klientami indywidualnymi

3. Dla klientów

-Przegląd i zapisanie się do dostępnych kursów i ofert treningowych

-Kontakt z pracownikami organizacji

-Opłacenie treningów

### Cel biznesowy

Organizacja działania KSW. Stworzenie niezbędnej infrastruktury cyfrowej do obsługi potrzeb firmy. Usprawnienie transakcji, przedstawienie oferty, ułatwiona komunikacja pomiędzy członkami organizacji.

### Użytkownicy

1. Kierownik

2. Trenerzy

3. Klienci

### Korzyści z systemu

|  |  |  |  |
| --- | --- | --- | --- |
| Korzyści \Użytkownik | Kierownik | Trener | Klient |
| 1. Możliwość zamieszczania ofert treningów |  | + | + |
| 1. Usprawnienie transakcji | + |  | + |
| 1. Ułatwienie komunikacji |  | + | + |
| 1. Obsługa zapisów na treningi |  | + | + |
| 1. Kontrola i nadzór działania organizacji | + |  |  |

### Ograniczenia projektowe i wdrożeniowe

Zabezpieczenia danych osobowych zgodne z wymogami RODO - w celu ochrony prywatności członków klubu.

## Wymagania funkcjonalne

### 5.2.1 Lista wymagań

lista numerowana – czyli lista przypadków użycia lub bardziej ogólnie sformułowane wymagania, np. wymagania użytkownika

1. Kierownik może dodać trenerów
2. Kierownik może usunąć trenerów
3. Kierownik może zablokować konkretnemu klientowi możliwość zapisywania się na kursy
4. Kierownik może zmienić cenę treningów grupowych
5. Kierownik może zmienić godziny zajęć grupowych
6. Kierownik może zamieszczać ogłoszenia dotyczące treningów grupowych, zawodów oraz nowości dotyczących organizacji
7. Trener może zamieszczać ogłoszenia dotyczące treningów w jego grupie
8. Trener może określać na stronie cenę i dostępność treningów indywidualnych
9. Trener może dodawać do grafika terminy treningów indywidualnych
10. Trener może zaakceptować uczestnictwo klienta w treningach grupowych i indywidualnych
11. Klient może sprawdzić informacje dotyczące dostępnych treningów i trenerów
12. Klient może zgłosić chęć dołączenia do (lub odejścia od) grupy treningowej
13. Klient może zgłosić chęć uczestnictwa w treningu indywidualnym
14. Klient może założyć konto i zmienić w nim informacje

### 5.2.2 Diagramy przypadków użycia

Obraz zawierający tekst, diagram, zrzut ekranu, linia

Opis wygenerowany automatycznie

###### Diagram 01 - Kierownik i trener

Obraz zawierający tekst, zrzut ekranu, diagram, linia

Opis wygenerowany automatycznie

###### Diagram 02 – Klient

### 5.2.3 Szczegółowy opis wymagań

* *K01 -* Kierownik może dodać trenerów
* (5) Kontrola i nadzór działania organizacji
* Kierownik, trener
* Scenariusz główny:
* Warunki początkowe: Kierownik jest zalogowany do systemu administracyjnego, brak trenera dla danej grupy treningowej
* Przebieg:

1. Kierownik wybiera opcję „Dodaj nowego trenera”.
2. System wyświetla formularz.
3. Kierownik wprowadza dane osobowe nowego trenera oraz jego specjalizacje.
4. System zapisuje informacje o nowym trenerze w bazie danych.

* Efekty: Dodanie nowego trenera do listy trenerów i grafiku
* Częstotliwość: 1/5 (bardzo rzadko)
* Istotność: 4/5

* *K02 -* Kierownik może usunąć trenerów
* (5) Kontrola i nadzór działania organizacji
* Kierownik, trener
* Scenariusz główny:
* Warunki początkowe: Kierownik jest zalogowany do systemu administracyjnego, trener kończy współpracę z organizacją
* Przebieg:

1. Kierownik wybiera opcję „Zarządzaj trenerami” z panelu administracyjnego.
2. System wyświetla listę trenerów.
3. Kierownik wybiera konkretnego trenera, którego chce usunąć.
4. Kierownik wybiera funkcję usuwania.
5. Kierownik potwierdza decyzję usuwania.
6. System usuwa wszystkie powiązane z trenerem dane i wyłącza jego konto.

* Efekty: Usunięcie trenera z listy trenerów i grafiku
* Częstotliwość: 1/5
* Istotność: 4/5
* *K03 -* Kierownik może zablokować konkretnemu klientowi możliwość zapisywania się na kursy
* (5) Kontrola i nadzór działania organizacji
* Kierownik, klient
* Scenariusz główny:
* Warunki początkowe: Kierownik jest zalogowany do systemu administracyjnego
* Przebieg:

1. Kierownik wyszukuje klienta w panelu administracyjnym.
2. System wyświetla listę klientów spełniających warunki wyszukiwania
3. Kierownik wybiera opcję „Zablokuj możliwość zapisywania się na kursy”.
4. System blokuje możliwość zapisywania się na kursy dla danego klienta.
5. Klient otrzymuje powiadomienie o blokadzie.

* Efekty: Zablokowanie klienta
* Częstotliwość: 1/5
* Istotność: 2/5
* *K04 -* Kierownik może zmienić cenę treningów grupowych
* Możliwość zamieszczania ofert treningów
* Kierownik
* Scenariusz główny:
* Warunki początkowe: Kierownik jest zalogowany do systemu administracyjnego
* Przebieg:

1. Kierownik przechodzi do sekcji „Ustawienia cen” w panelu administracyjnym.
2. Kierownik znajduje opcję „Ceny treningów grupowych”.
3. System wyświetla listę kursów
4. Kierownik wprowadza nowe ceny dla poszczególnych rodzajów treningów grupowych.
5. System aktualizuje ceny w interfejsie użytkownika.

* Efekty: Nowa cena figuruje w cenniku
* Częstotliwość: 2/5
* Istotność: 4/5
* *K05 -* Kierownik może zmienić godziny treningów grupowych
* Możliwość zamieszczania ofert treningów
* Kierownik
* Scenariusz główny:
* Warunki początkowe: Kierownik jest zalogowany do systemu administracyjnego
* Przebieg:

1. Kierownik wybiera funkcjonalność „Zmiana godzin treningów grupowych”
2. System wyświetla listę treningów grupowych
3. Kierownik wybiera konkretny dzień i grupę zajęć.
4. Kierownik zmienia godziny rozpoczęcia i zakończenia zajęć.
5. System aktualizuje harmonogram zajęć i informuje trenerów oraz klientów o zmianach.

* Efekty: Nowe godziny figurują w harmonogramie
* Częstotliwość: 2/5
* Istotność: 4/5
* *K06 -* Kierownik może zamieszczać ogłoszenia dotyczące treningów grupowych, zawodów oraz nowości dotyczących organizacji.
* Możliwość zamieszczania ofert treningów
* Kierownik
* Scenariusz główny:
* Warunki początkowe: Kierownik jest zalogowany do systemu administracyjnego
* Przebieg:

1. Kierownik wybiera opcję „Dodaj ogłoszenie”.
2. System wyświetla formularz ogłoszenia
3. Kierownik wpisuje treść ogłoszenia oraz określa grupę docelową (np. trenerzy, klienci).
4. System wyświetla ogłoszenie na stronie głównej lub w odpowiedniej sekcji dla wybranej grupy.

* Efekty: Nowy post jest widoczny dla użytkowników
* Częstotliwość: 4/5
* Istotność: 3/5
* *T01 -* Trener może zamieszczać ogłoszenia dotyczące treningów w jego grupie
  + (3) Ułatwienie komunikacji
  + Trener
  + Scenariusz główny:
    - Warunki początkowe: Trener jest zalogowany na swoje konto, trener posiada przypisaną grupę treningową
    - Przebieg:
      1. Trener przechodzi do sekcji „Ogłoszenia” lub „Aktualności”.
      2. System wyświetla formularz ogłoszenia
      3. Trener tworzy nowe ogłoszenie dotyczące treningów dla swojej grupy.
      4. System wyświetla ogłoszenie na stronie głównej grupy lub w sekcji dedykowanej ogłoszeniom trenera.
    - Efekty: nowe ogłoszenie jest utworzone i widoczne dla wszystkich
    - Wymagania niefunkcjonalne: 2.1
    - Częstotliwość: 2/5
    - Istotność: 3/5
* *T02 -* Trener może określać na stronie cenę i dostępność treningów indywidualnych
  + (1) Możliwość zamieszczania ofert treningów
  + Trener
  + Scenariusz główny:
    - Warunki początkowe: Trener jest zalogowany na swoje konto
    - Przebieg
      1. Trener przechodzi do sekcji „Treningi indywidualne”.
      2. System wyświetla harmonogram zajęć trenera
      3. Trener wybiera rodzaj treningu i określa cenę oraz dostępność terminów.
      4. System aktualizuje informacje na stronie trenera oraz w systemie rezerwacji.
    - Efekty: oferta treningów indywidualnych jest zaktualizowana i zmiany są widoczne dla wszystkich.
    - Wymagania niefunkcjonalne: 2.1
    - Częstotliwość: 2/5
    - Istotność: 5/5
* *T03 -* Trener może dodawać do grafika terminy treningów indywidualnych
  + (3) Ułatwienie komunikacji
  + Trener
  + Scenariusz główny:
    - Warunki początkowe: Trener jest zalogowany na swoje konto , istnieje przynajmniej jeden klient
    - Przebieg:
      1. Trener przechodzi do sekcji „Grafik treningów”.
      2. System wyświetla harmonogram zajęć trenera
      3. Trener dodaje nowe terminy treningów indywidualnych, uwzględniając swoją dostępność.
      4. System aktualizuje grafik treningów na stronie trenera oraz w systemie rezerwacji.
    - Efekty: oferta treningów indywidualnych jest zaktualizowana i zmiany są widoczne dla wszystkich.
    - Wymagania niefunkcjonalne: 2.2
    - Częstotliwość: 3/5
    - Istotność: 5/5
* *T04 -* Trener może zaakceptować uczestnictwo klienta w treningach grupowych i indywidualnych
  + (4) Obsługa zapisów na treningi
  + Trener, klient
  + Scenariusz główny:
    - Warunki początkowe: Trener jest zalogowany na swoje konto , klient zgłosił chęć udziału w treningu
    - Przebieg:
      1. Trener otrzymuje powiadomienie o nowej rezerwacji treningu.
      2. Trener przechodzi do sekcji „Rezerwacje”.
      3. System wyświetla prośby rezerwacji zajęć
      4. Trener akceptuje rezerwację klienta dla konkretnego treningu.
      5. System potwierdza uczestnictwo klienta i aktualizuje stan rezerwacji.
    - Efekty: Klient jest informowany o decyzji trenera (zapisany/niezapisany)
    - Wymagania niefunkcjonalne: 2.1, 2.2
    - Częstotliwość: 3/5
    - Istotność: 5/5
* *C01* – Klientmoże wyświetlić informacje dotyczące dostępnych treningów i trenerów
* (3) Ułatwienie komunikacji
* Klient
* Scenariusz główny:
* Warunki początkowe: N/A
* Przebieg:
  + - 1. Klientprzechodzi do sekcji „Treningi” lub „Trenerzy”.
      2. System wyświetla informacje o treningach lub trenerach
      3. Klientprzegląda dostępne treningi lub informacje o trenerach, takie jak specjalizacje i opinie innych klientów.
      4. System prezentuje klientowi aktualne informacje na podstawie bazy danych.
* Efekty: Wyświetlenie informacji dotyczących dostępnych treningów i trenerów
* Wymagania niefunkcjonalne: 3.1, 3.3, 3.4
* Częstotliwość: 3/5
* Istotność 3/5
* *C02* – Klientmoże zgłosić chęć dołączenia do grupy treningowej
* (4) Obsługa zapisów na treningi
* Klient
* Scenariusz główny:
* Warunki początkowe: Klient jest zalogowany na swoje konto, klient nie należy do grupy treningowej
* Przebieg:
  + - 1. Klientprzechodzi do sekcji „Moje grupy treningowe”.
      2. System wyświetla listę grup
      3. Klientwybiera opcję „Dołącz” obok odpowiedniej grupy.
      4. Trener otrzymuje prośbę o dołączenie.
* Efekty: Wysłanie wniosku o dodanie do grupy
* Wymagania niefunkcjonalne: 1.1, 2.1
* Częstotliwość: 2/5
* Istotność 4/5
* *C03* – Klientmoże zgłosić chęć odejścia od grupy treningowej
* (4) Obsługa zapisów na treningi
* Klient
* Scenariusz główny:
* Warunki początkowe: Klient jest zalogowany na swoje konto , klient należy do grupy treningowej
* Przebieg:
  + - 1. Klientprzechodzi do sekcji „Moje grupy treningowe”.
      2. System wyświetla grupy do których należy klient
      3. Klientwybiera opcję „Odejdź” obok odpowiedniej grupy.
      4. System aktualizuje status uczestnictwa klienta w grupie treningowej.
* Efekty: Wysłanie wniosku o usunięcie z grupy
* Wymagania niefunkcjonalne: 1.1, 2.1
* Częstotliwość: 1/5
* Istotność 4/5
* *C04* – Klient może zgłosić chęć uczestnictwa w treningu indywidualnym
* (4) Obsługa zapisów na treningi
* Klient
* Scenariusz główny:
* Warunki początkowe: Klient jest zalogowany na swoje konto , klient nie jest zapisany na treningi indywidualne u trenera
* Przebieg:
  + - 1. Klientprzechodzi do sekcji „Treningi indywidualne”.
      2. System wyświetla listę trenerów i ich dostępnych terminów
      3. Klientwybiera trenera i termin treningu.
      4. Klientwysyła prośbę o rezerwację treningu indywidualnego.
      5. System informuje trenera o zgłoszeniu klienta i oczekuje na potwierdzenie.
* Efekty: Wysłanie wniosku o zapisanie klienta na treningi indywidualne
* Wymagania niefunkcjonalne: 1.1, 2.1
* Częstotliwość: 1/5
* Istotność 2/5
* *C05* – Klient może założyć i edytować konto
* (2) Usprawnienie transakcji, (3) Ułatwienie komunikacji, (4) Obsługa zapisów na treningi
* Klient
* Scenariusz główny:
* Warunki początkowe: Nie istnieje konto z identycznymi danymi
* Przebieg:
  + - 1. Klient przechodzi do strony rejestracji na platformie.
      2. System wyświetla formularz rejestracyjny
      3. Klientwypełnia formularz rejestracyjny, podając wymagane dane osobowe.
      4. Klientpotwierdza rejestrację poprzez link aktywacyjny wysłany na podany adres e-mail.
      5. Po zalogowaniu się do konta, klient może edytować swoje dane osobowe w sekcji „Ustawienia konta”.
      6. System zapisuje zmiany i aktualizuje informacje w bazie danych.
* Efekty: Założenie konta klienta
* Wymagania niefunkcjonalne: 1.1, 2.1
* Częstotliwość: 2/5
* Istotność 5/5

## 5.3 Wymagania niefunkcjonalne

W odniesieniu do całego systemu, modułów lub innych składowych systemu

1. Wydajność – w odniesieniu do konkretnych sytuacji – funkcji systemu
2. Bezpieczeństwo – utrata, zniszczenie danych, zniszczenie innego systemu przez nasz – wraz z działaniami zapobiegawczymi i ograniczającymi skutki
3. Zabezpieczenia
4. Inne cechy jakości – najlepiej ilościowo, żeby można było zweryfikować (zmierzyć) – adaptowalność, dostępność, poprawność, elastyczność, łatwość konserwacji, przenośność, awaryjność, testowalność, użyteczność
5. Wydajność
   1. System musi obsługiwać jednocześnie co najmniej 1 administratora, 10 trenerów I 100 klientów.
6. Dostępność:
   1. System powinien oferować jednolite ścieżki interakcji, które prowadzą użytkowników przez wszystkie moduły systemowe. Każdy etap interakcji z systemem, od nawigacji po wykonanie zadań, powinien być wspomagany przez instrukcje prowadzące użytkownika
   2. System powinien oferować kontekstowe wskazówki pomocnicze przy każdej interakcji z kalendarzem.
7. Aktualność:
   1. Wszystkie wprowadzone zmiany powinni być widoczne dla wszystkich użytkowników w ciągu maksymalnie 15 sekund przy obciążeniu systemu odpowiadającym szczytowemu użytkowaniu.
8. Dokładność:
   1. System musi zapewniać interakcję z czasem z dokładnością do jednej minuty.
   2. System musi zapewniać interakcję z pieniędzmi z dokładnością do jednego grosza.
9. Bezpieczeństwo:
   1. System powinien oferować automatyczne backupy danych co 24 godzin.
   2. System powinien umożliwiać przywrócenie dowolnego stanu informacji z ostatniego roku.
   3. System powinien zapewniać mechanizm odwzorowania wszystkich zmian na bieząco (replikacja danych).
   4. System powinien być zaprojektowany do szybkiego przywrócenia funkcjonalności w przypadku awarii, z gwarantowanym czasem przywrócenia działania nie przekraczającym 2 godzin.
10. Integracja:
    1. System musi być zdolny do integracji z co najmniej jednym zewnętrznym systemem płatności w celu obsługi transakcji finansowych
    2. System musi umożliwiać eksport danych do zewnętrznych systemów kalendarza z użyciem Google Calendar

# Zarządzanie projektem

## 6.1 Zasoby ludzkie

(rzeczywiste lub hipotetyczne) – przy realizacji projektu

Należy założyć, że projekt byłby realizowany w całości jako projekt komercyjny a nie tylko częściowo w ramach zajęć na uczelni

## 6.2 Harmonogram prac

Etapy mogą się składać z zadań.

Wskazać czasy trwania poszczególnych etapów i zadań – wykres Gantta.

obejmuje również harmonogram wdrożenia projektu – np. szkolenie, rozruch, konfiguracja, serwis – może obejmować różne wydania (tj. o różnej funkcjonalności – personal, professional, enterprise) i wersje (1.0, 1.5, itd.)

## 6.3 Etapy/kamienie milowe projektu

dla głównych etapów projektu

W grupach prowadzonych przez ŁR ten cały ten rozdział jest opcjonalny – dla chętnych. Nie jest omawiany na wykładzie!

Studenci powinni skonsultować szczegółowe wymagania w tym zakresie z nauczycielem prowadzącym zajęcia w danej grupie.

# Zarządzanie ryzykiem

## 7.1 Lista czynników ryzyka

Wypełniona lista kontrolna

## 7.2 Ocena ryzyka

prawdopodobieństwo i wpływ

## 7.3 Plan reakcji na ryzyko

Działania w odniesieniu do poszczególnych ryzyk.

Mogą być wg różnych strategii, tj. kilka strategii dla pojedynczego czynnika ryzyka

Rozdział obowiązkowy w zespołach co najmniej 3-osobowych, w mniejszych – do uzgodnienia z prowadzącym zajęcia.

# Zarządzanie jakością

## 8.1 Scenariusze i przypadki testowe

głównie testowanie funkcjonalności, ale może być też testowanie wymagań niefunkcjonalnych/zgodności; każdy scenariusz od nowej strony, musi zawierać co najmniej następujące informacje (sugerowany układ tabelaryczny, np. wg szablonu podanego w osobnym pliku lub na wykładzie):

* numer – jako ID
* nazwa scenariusza – co test w nim testowane (max kilka wyrazów)
* kategoria – poziom/kategoria testów
* opis – dodatkowe opcjonalne informacje, które nie zmieściły się w nazwie
* tester - konkretna osoba lub klient/pracownik,
* termin – kiedy testowanie ma być przeprowadzane,
* narzędzia wspomagające – jeśli jakieś są używane przy danym scenariuszu
* przebieg działań – tabela z trzema kolumnami: lp. oraz opisującymi działania testera i systemu
* założenia, środowisko, warunki wstępne, dane wejściowe – przygotowanie przed uruchomieniem testów
* zestaw danych testowych – najlepiej w formie tabelarycznej – jakie konkretnie dane mają być użyte przez testera i zwrócone przez system w poszczególnych krokach przebiegu działań
* *przebieg lub zestaw danych testowych musi zawierać jawną informację o warunku zaliczenia testu*

*5 scenariuszy dla 2osobowych, max 2 mogą być niefunkcjonalnymi*

*Przykład prezentacji ostatnich dwóch punktów:*

|  |  |  |
| --- | --- | --- |
| *L.p* | *Działanie testera* | *Działanie systemu* |
| *1.* | *Wybiera opcje* |  |
| *2.* |  | *Wyświetla formularz* |
| *3.* | *Podaje dane* |  |
| *4.* | *Zatwierdza dane* |  |

|  |  |
| --- | --- |
| *Zestaw danych 1* |  |
| *Krok 3* | *Długopis, cena 25* |
| *Krok 5* | *„Dodano produkt”* |
|  | *W bazie dodano produkt długopis* |
| *Zestaw danych 2* |  |
| *3* | *Długopis, cena -5* |
| *5* | *„Nieprawidłowa cena”* |

Wstawiamy zestawy danych, które przetestują system, niestandardowe, np. ujemna ilość lub cena

Mogą być operacje wyjścia również

* PT001
* Funkcjonalność utworzenia nowego konta
* Funkcjonalny
* opis
* tester – kierownik?
* termin – ?
* Manualny
* przebieg działań:

|  |  |  |
| --- | --- | --- |
| *L. p* | *Działanie testera* | *Działanie systemu* |
| 1 | Wchodzi na stronę |  |
| 2 | Wybiera opcję tworzenia nowego konta |  |
| 3 |  | Wyświetla folmularz |
| 4 | Podaje dane: korotkie hasło |  |
| 5 |  | Wyświetla komunikat |
| 6 | Podaje dane: brak imienia |  |
| 7 |  | Wyświetla komunikat |
| 8 | Podaje dane: poprawne |  |
| 9 |  | Tworzy nowe konto i wyświetla komunikat |
| 10 |  | Loguje użytkownika na konto |
| 11 | Wchodzi na stronę w innej instancji przeglądarki |  |
| 12 | Wybiera opcję tworzenia nowego konta |  |
| 13 |  | Wyświetla folmularz |
| 14 | Podaje dane: poprawne (duplikat) |  |
| 15 |  | Wyświetla komunikat |
| 16 | W zalogowanym oknie wybiera opcję wyświetlenia danych konta |  |
| 17 |  | Wyświetla dane konta |
| 18 | Sprawdza poprawność danych konta |  |

* zestaw danych testowych

|  |  |
| --- | --- |
| *Zestaw danych 1: krótkie hasło* | |
| Krok 4 | Imię: “Alex”, Nazwisko: “Kowalski”, email: “[alkowal@gmail.com](mailto:alkowal@gmail.com)”, hasło: “123” |
| Krok 5 | "Hasło jest zbyt krótkie!" |
| Zestaw danych 2: brak imienia | |
| Krok 6 | Imię: “”, Nazwisko: “Kowalski”, email: “[alkowal@gmail.com](mailto:alkowal@gmail.com)”, hasło: “Pa$sWoRd!” |
| Krok 7 | "Pole imię nie może być puste!" |
| Zestaw danych 3: poprawny | |
| Krok 8 | Imię: “Alex”, Nazwisko: “Kowalski”, email: “[alkowal@gmail.com](mailto:alkowal@gmail.com)”, hasło: “Pa$sWoRd!” |
| Krok 9 | “Konto zostało pomyślne utworzone.” |
| Krok 14 | Imię: “Alex”, Nazwisko: “Kowalski”, email: “[alkowal@gmail.com](mailto:alkowal@gmail.com)”, hasło: “Pa$sWoRd!” |
| Krok 15 | "Konto z tym adresem email już istnieje!" |

* Założenia:   
  - tester ma dwie instancji przeglądarki (różne lub incognito)  
  - brak istniejącego konta z emailem testowym
* PT002
* Proces dolączenia i odłączenia do i od grupy
* Funkcjonalny
* opis – dodatkowe opcjonalne informacje, które nie zmieściły się w nazwie
* tester - konkretna osoba lub klient/pracownik,
* termin – kiedy testowanie ma być przeprowadzane,
* Manualny
* przebieg działań

|  |  |  |
| --- | --- | --- |
| *L. p* | *Działanie testera* | *Działanie systemu* |
| 1 | Wchodzi na stronę w 1. przeglądarce |  |
| 2 |  | Wyświetla stronę |
| 3 | Loguje się na konto klienta |  |
| 4 |  | Potwierdza logowanie |
| 5 | Wybiera panel klienta |  |
| 6 |  | Wyświetla panel klienta |
| 7 | Wybiera seksję grup |  |
| 8 |  | Wyświetla listę dostępnych grup |
| 9 | Wysyła prośbę o dołączenie do grupy 1 |  |
| 10 |  | Wyświetla komunikat |
| 11 | Wchodzi na stronę w 2. przeglądarce |  |
| 12 |  | Wyświetla stronę |
| 13 | Loguje się na konto administratora |  |
| 14 |  | Potwierdza logowanie |
| 15 | Wybiera panel administratora |  |
| 16 |  | Wyświetla panel administratora |
| 17 | Wybiera sekcję prósb o dolączenie do grup |  |
| 18 |  | Wyświetla sekcję prósb o dolączenie do grup |
| 19 | Odrzuca prośbę wysłaną przez testowane konto klienta |  |
| 20 |  | Wysyła powiadomienie dla klienta |
| 21 | Sprawdza powiadomienia klienta |  |
| 22 |  | Wyświetla wszystkie powiadomienia |
| 23 | Wysyła prośbę o dołączenie do grupy 2 |  |
| 24 |  | Wyświetla komunikat |
| 25 | Na koncie administratora akceptuje prośbę wysłaną przez testowane konto klienta |  |
| 26 |  | Wysyła powiadomienie dla klienta |
| 27 | Sprawdza powiadomienia klienta |  |
| 28 |  | Wyświetla wszystkie powiadomienia |
| 29 | Odświeza stronę |  |
| 30 |  | Wyświetla grupę, do której się dołączył klient |
| 31 | Opuszcza grupę 2 |  |
| 32 |  | Wyświetla listę grup z możliwością ponownego dołączenia |

* zestaw danych testowych

|  |  |
| --- | --- |
| *Zestaw danych 1* | |
| Krok 10 | “Pomyślnie zgłoszono chęć dołączenia do grupy 1.” |
| Krok 20 | "Twoja prośba o dołączenie do grupy 1 została odrzucona." |
| Krok 24 | "Pomyślnie zgłoszono chęć dołączenia do grupy 2." |
| Krok 26 | "Twoja prośba o dołączenie do grupy 2 została zaakceptowana." |

* Założenia:  
  - tester ma dwie instancji przeglądarki (różne lub incognito)  
  - istnieją grupy 1 i 2  
  - testowane konto nie należy do żadnej grupy
* PT003
* Proces zmiany hasła
* Funkcjonalny
* opis
* tester
* termin – kiedy testowanie ma być przeprowadzane,
* Manualny
* przebieg działań – tabela z trzema kolumnami: lp. oraz opisującymi działania testera i systemu

|  |  |  |
| --- | --- | --- |
| *L. p* | *Działanie testera* | *Działanie systemu* |
| 1 | Loguje się na konto klienta |  |
| 2 |  | Potwierdza logowanie |
| 3 | Wybiera panel klienta |  |
| 4 |  | Wyświetla panel klienta |
| 5 | Wybiera opcję zmiany hasła |  |
| 6 |  | Wyswietla formularz zmiany hasła |
| 7 | Wprowadza obecne, nowe hasło i potwierdzenie nowego hasła |  |
| 8 |  | Wyświetla komunikat |
| 9 | Wprowadza obecne, nowe hasło i potwierdzenie nowego hasła |  |
| 10 |  | Wyświetla komunikat |
| 11 | Wprowadza obecne, nowe hasło i potwierdzenie nowego hasła |  |
| 12 |  | Wyświetla komunikat |
| 13 | Wprowadza obecne, nowe hasło i potwierdzenie nowego hasła |  |
| 14 |  | Zmienia hasło I wyświetla komunikat |
| 15 | Wylogowuje się |  |
| 16 |  | Wyświetla formularz logowania |
| 17 | Loguje się przy użyciu nowego hasła |  |
| 18 |  | Potwierdza logowanie |

* zestaw danych testowych – najlepiej w formie tabelarycznej – jakie konkretnie dane mają być użyte przez testera i zwrócone przez system w poszczególnych krokach przebiegu działań

|  |  |
| --- | --- |
| *Zestaw danych 1: do logowania* | |
| Krok 1 | email: “[alkowal@gmail.com](mailto:alkowal@gmail.com)”, “hasło”: “masło” |
| Zestaw danych 2: nieprawidłowe obecne hasło | |
| Krok 7 | obecne: “3271317”, nowe: “12345”, nowe\_potwierdzenie: “12345” |
| Krok 8 | “Nieprawidłowe obecne hasło!” |
| *Zestaw danych 3: krótkie hasło* | |
| Krok 9 | obecne: “masło”, nowe: “123”, nowe\_potwiedzenie: “123” |
| Krok 10 | "Hasło jest zbyt krótkie!" |
| Zestaw danych 4: niepasujące potwiedzenie | |
| Krok 11 | obecne: “masło”, nowe: “PaSsWoRd!”, nowe\_potwiedzenie: “Pa$sWoRd!” |
| Krok 12 | "Hasła nie pasują do siebie!" |
| Zestaw danych 5: poprawny | |
| Krok 13 | obecne: “masło”, nowe: “Pa$sWoRd!”, nowe\_potwiedzenie: “Pa$sWoRd!” |
| Krok 14 | "Hasło zostało pomyślnie zmienione." |
| Zestaw danych 6: nowy do logowania | |
| Krok 17 | email: “[alkowal@gmail.com](mailto:alkowal@gmail.com)”, “hasło”: “Pa$sWoRd!” |

* *Założenia:  
  - dane dostępu do konta testowego są prawidłowe*
* PT004
* Proces zamieszczenie ogłoszenia przez trenera
* Funkcjonalny
* opis
* tester
* termin
* Manualny
* przebieg działań

|  |  |  |
| --- | --- | --- |
| *L. p* | *Działanie testera* | *Działanie systemu* |
| 1 | Loguje się na konto trenera |  |
| 2 |  | Potwierdza logowanie |
| 3 | Wybiera panel trenera |  |
| 4 |  | Wyświetla panel trenera |
| 5 | Wybiera sekcję “Zarządzanie blogiem grupy” |  |
| 6 |  | Wyświetla sekcję |
| 7 | Wybiera opcję “Dodaj nowe ogłoszenie” |  |
| 8 |  | Wyświetla formularz dodawania ogłoszenia |
| 9 | Wprowadza dane ogłoszenia i próbuje je opublikować |  |
| 10 |  | Wyświetla komunikat |
| 11 | Wprowadza dane ogłoszenia i próbuje je opublikować |  |
| 12 |  | Wyświetla komunikat |
| 13 | Wprowadza dane ogłoszenia i próbuje je opublikować |  |
| 14 |  | Dodaje ogłoszenie I wyświetla komunikat |
| 15 | Ponownie wybiera blog grupy |  |
| 16 |  | Wyświetla blog grupy |
| 17 | Weryfikuje, czy ogłoszenie jest widoczne |  |

* zestaw danych testowych – najlepiej w formie tabelarycznej – jakie konkretnie dane mają być użyte przez testera i zwrócone przez system w poszczególnych krokach przebiegu działań

|  |  |
| --- | --- |
| *Zestaw danych 1: brak tytułu* | |
| Krok 9 | tytuł: “”, treść: “Lorem ipsum” |
| Krok 10 | "Pole tytuł nie może być puste!" |
| Zestaw danych 2: brak treści | |
| Krok 11 | tytuł: “test”, treść: “” |
| Krok 12 | "Pole treść nie może być puste!" |
| Zestaw danych 3: poprawny | |
| Krok 13 | tytuł: “test”, treść: “Lorem ipsum” |
| Krok 14 | "Ogłoszenie zostało pomyślnie dodane." |

* Założenia:   
  - testowane konto trenera ma swoją grupę
* PT005
* Działalność blokowania możliwości zapisywania się na kursy
* Funkcjonalny
* opis
* tester
* termin
* Manualny
* przebieg działań – tabela z trzema kolumnami: lp. oraz opisującymi działania testera i systemu

|  |  |  |
| --- | --- | --- |
| *L. p* | *Działanie testera* | *Działanie systemu* |
| 1 | Loguje się na konto klienta |  |
| 2 |  | Potwierdza logowanie |
| 3 | Wybiera panel klienta |  |
| 4 |  | Wyświetla panel klienta |
| 5 | Wybiera sekcję “Moje grupy” |  |
| 6 |  | Wyświetla “Moje grupy” |
| 7 | Weryfikuje, że klient należy do grupy |  |
| 8 | Loguje się na konto administratora (inna przeglądarka) |  |
| 9 |  | Potwierdza logowanie |
| 10 | Wybiera panel administratora |  |
| 11 |  | Wyświetla panel administratora |
| 12 | Wybiera sekcję “Zarządzaj klientami” |  |
| 13 |  | Wyświetla listę klientów |
| 14 | Wprowadza nazwisko klienta w polu wyszukiwania |  |
| 15 |  | Wyświetla listę klientów według nazwiska |
| 16 | Wybiera klienta z listy i wybiera opcję “Zablokuj” |  |
| 17 |  | Wyświetla komunikat |
| 18 | Potwierdza |  |
| 19 |  | Wyświetla komunikat |
| 20 | Ponownie sprawdza sekcję “Moje grupy” na koncie klienta |  |
| 21 |  | Wyświetla “Moje grupy” |
| 22 | Sprawdza, czy klient nie należy do żadnej z grup i nie może do nich dołączyć. |  |
| 23 |  | Wyświetla komunikat |

* zestaw danych testowych

|  |  |
| --- | --- |
| *Zestaw danych 1* | |
| Krok 14 | Kowalski |
| Krok 17 | “Czy na pewno chcesz zablokować tego klienta?” |
| Krok 19 | "Klient został zablokowany." |
| Krok 23 | "Konto zostało zablokowane. Nie masz dostępu do grup." |

* Założenia:  
  - tester ma dwie instancji przeglądarki (różne lub incognito)  
  - testowane konto należy do grupy i nie jest zablokowane
* PT006
* Proces zmiany ceny treningu grupowego
* Funkcjonalny
* opis
* tester
* termin
* Manualny
* przebieg działań

|  |  |  |
| --- | --- | --- |
| *L. p* | *Działanie testera* | *Działanie systemu* |
| 1 | Loguje się na konto kierownika |  |
| 2 |  | Potwierdza logowanie |
| 3 | Wybiera panel administratora |  |
| 4 |  | Wyświetla panel administratora |
| 5 | Wybiera sekcję “Zarządzanie treningami grupowymi” |  |
| 6 |  | Wyświetla sekcję |
| 7 | Dla grupy 1 wybiera “Zmienić cenę” |  |
| 8 |  | Wyświetla formularz zmiany ceny |
| 9 | Wprowadza nową cenę dla treningu grupowego |  |
| 10 |  | Wyświetla komunikat |
| 11 | Wprowadza nową cenę dla treningu grupowego |  |
| 12 |  | Wyświetla komunikat |
| 13 | Wprowadza nową cenę dla treningu grupowego |  |
| 14 |  | Aktualizuje cenę I wyświetla komunikat |
| 15 | Wraca do zarządzania treningami grupowymi |  |
| 16 |  | Wyświetla sekcję |
| 17 | Sprawdza poprawność zaktualizowania ceny |  |

* zestaw danych testowych

|  |  |
| --- | --- |
| *Zestaw danych 1: brak wartości* | |
| Krok 9 | cena: “” |
| Krok 10 | "Pole cena nie może być puste!" |
| *Zestaw danych 2: wartość ujemna* | |
| Krok 11 | cena: “-30” |
| Krok 12 | "Cena nie może być ujemna" |
| *Zestaw danych 3: poprawny* | |
| Krok 13 | cena: “30” |
| Krok 14 | “Cena została pomyślnie zmieniona." |

* Założenia:  
  - istnieje przynajmniej jedna grupa
* PT007
* Wydajność systemu podczas jednoczesnego logowania 100 użytkowników
* Niefunkcjonalny
* opis
* tester
* termin
* narzędzia wspomagające: JMeter
* przebieg działań

|  |  |  |
| --- | --- | --- |
| *L. p* | *Działanie testera* | *Działanie systemu* |
| 1 | Przygotowanie narzędzia do testowania |  |
| 2 | Skonfigurowanie scenariusza testowego dla 100 użytkowników |  |
| 3 | Uruchomienie testu |  |
| 4 |  | Przetwarza żądania logowania |
| 5 | Monitorowanie czasu odpowiedzi systemu |  |
| 6 |  | Rejestruje wszystkie blędy I awarie |
| 7 | Zakonczenie testu i analiza wyników |  |

* zestaw danych testowych: brak
* Założenia:
* PT008
* nazwa scenariusza – co test w nim testowane (max kilka wyrazów)
* kategoria – poziom/kategoria testów
* opis – dodatkowe opcjonalne informacje, które nie zmieściły się w nazwie
* tester - konkretna osoba lub klient/pracownik,
* termin – kiedy testowanie ma być przeprowadzane,
* narzędzia wspomagające – jeśli jakieś są używane przy danym scenariuszu
* przebieg działań – tabela z trzema kolumnami: lp. oraz opisującymi działania testera i systemu

|  |  |  |
| --- | --- | --- |
| *L. p* | *Działanie testera* | *Działanie systemu* |
|  |  |  |
|  |  |  |
|  |  |  |

* zestaw danych testowych – najlepiej w formie tabelarycznej – jakie konkretnie dane mają być użyte przez testera i zwrócone przez system w poszczególnych krokach przebiegu działań

|  |  |
| --- | --- |
| *Zestaw danych 1* | |
|  |  |
|  |  |
|  |  |
|  |  |

* założenia, środowisko, warunki wstępne, dane wejściowe – przygotowanie przed uruchomieniem testów

# Projekt techniczny

## 9.1 Opis architektury systemu

z ew. rysunkami pomocniczymi

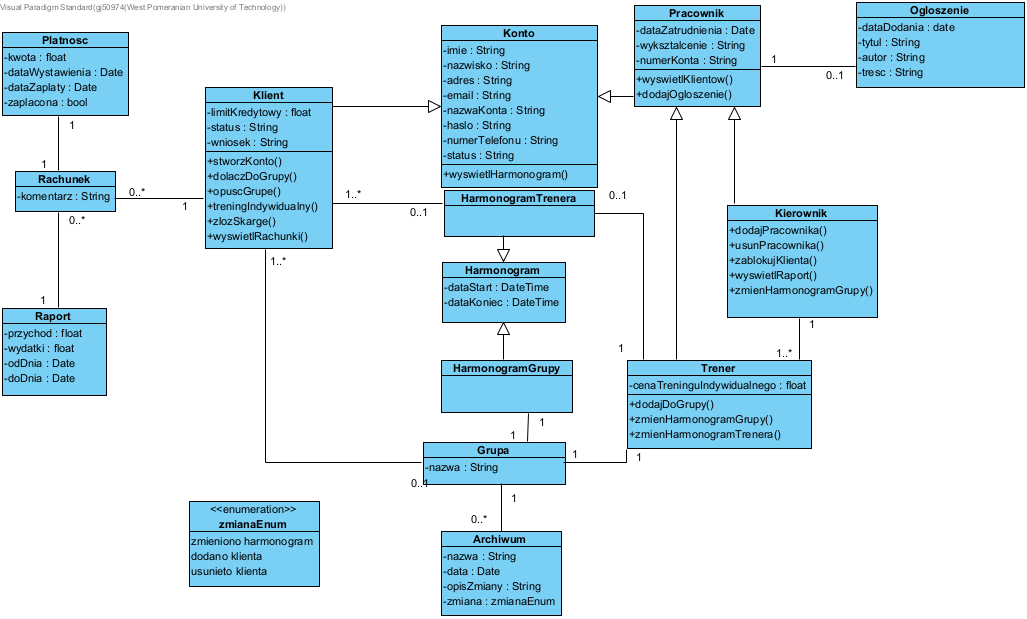
1. Ogólny schemat architektury oprogramowania  
   System został zaprojektowany z wykorzystaniem podejścia modularnego, co umożliwia dekompozycję na niezależne podsystemy i komponenty. Każdy podsystem oferuje orkeślone usługi. Zapewnia to wysoką spoistość i niskie sprzężenie. Poniżej przedstawiono podsystemu oraz usługi i komponenty, które one zapewniają:
   1. Podsystem zarządzania użytkownikami:
      * usługi związane z obsługą danych użytkowników, takie jak tworzenie profilu, autentykacja, autoryzacja.
      * komponenty: profil użytkownika, system logowania, zarządzanie uprawnieniami.
   2. Podsystem zarządzania treningami:
      * usługi dotyczące zarządzania treningami grupowymi i indywidualnymi, planowanie zajęć.
      * komponenty: harmonogram grupy, harmonogram indywidualny, kalendarz.
   3. Podsystem przetwarzania płatności:
      * usługi przetwarzania transakcji finansowych, weryfikacji płatności, generowania potwierdzeń.
      * komponenty: inicjator płatności, weryfikator płatnośći, generator rachunków.
2. Dekompozycja  
   Każdy komponent koncentruje się na ścisłym zbiorze zadań, co zapewnia wysoką spoistość. Komponenty komunikują się z innymi komponentami lub podsystemami poprzez zdefiniowane interfejsy, utrzymując niskie sprzężenie. Oto przykłady zadań dla niektórych komponentów:
   * + Profil użytkownika: zarządza informacjami osobowymi, zapewnia funkcjonalność ich aktualizacji oraz zarządzania kontem użytkownika.
     + Harmonogram grupy: odpowiada za ustalanie terminów zajęć grupowych, zarządzanie rezerwacjami i odwołaniami.
     + Inicjator płatności: inicjuje proces płatności, komunikuje się z zewnętrznymi systemami płatności elektronicznej.
3. Usługi  
   Każdy komponent dostarcza zbior powiązanych usług podporządkowanych realizacji wspólnego celu. Usługi te są ścisłe powiązane z głównymi zadaniami komponentu, co zapewnia, że komponent jest samowystarczalny i skupiony na określonym obszarze funkcjonalnym, co z kolei przekłada się na spoistość modułu i całego systemu. Oto przykłady zadań dla niektórych komponentów:
   * + Usługa tworzenia profilu:  
       - operacja: utworzProfil()  
       - parametry: imie, nazwisko, adres, email, numerTelefonu, hasło  
       - wynik: idProfilu
4. Interfejsy  
   Każdy komponent może oferować kilka interfejsów, które definiują zbiór operacji dostępnych dla różnych części systemu. Dzięki interfejsom, każdy komponent ma możliwość utrzymywać wysoką spoistość, skupiając się na ściśle powiązanych funkcjach, jednocześnie minimalizując sprzężenie z innymi częściami systemu przez klarowne definicje interakcji. To zwiększa bezpieczeństwo, elastyczność systemu oraz ułatwia zarządzanie i rozwój komponentów.  
   Na przykład, komponent zarządzania użytkownikami może używać interfejsu ILogger do logowania, co umożliwia elastyczność w zarządzaniu sposobami logowania i izoluje szczegóły implementacyjne modułu logowania od modułu zarządzania użytkownikami. Dzięki zastosowaniu interfejsu ILogger, moduł zarządzania użytkownikami może łatwo zmienić metodę logowania bez wpływu na swoją główną funkcjonalność, co zapewnia bezpieczeństwo i niezależność rozwoju obu modułów.
5. Komponenty i technologie   
   W sekcji 9.2, szczegółowo opisujemy języki programowania i technologie wykorzystane do realizacji poszczególnych komponentów.

## 9.2 Technologie implementacji systemu

|  |  |  |
| --- | --- | --- |
| Lp. | Technologia | Korzyści |
| 1. | Język programowania C++ | Bogata dokumentacja, język szeroko znany wśród developerów. Może być użyty do implementacji specjalistycznych funkcji, algorytmów lub modułów, które wymagają niskopoziomowej kontroli nad pamięcią lub wydajnością oraz do prototypu struktury systemu. |
| 2. | Baza danych MySQL | Oferuje ona niezawodność, skalowalność, i wydajność, a także zapewnia wsparcie dla zaawansowanych operacji bazodanowych, takich jak złączenia i indeksowanie. |
| 3. | HTML/CSS/JavaScript | Trzy podstawowe technologie webowe, niezbędne do budowy interfejsu użytkownika. HTML odpowiada za strukturę dokumentu, CSS za jego wygląd, a JavaScript za interaktywność i dynamiczne funkcjonalności. |
| 4. | JWT (JSON Web Tokens) | JWT to standard autoryzacji i uwierzytelniania, który pozwala na bezpieczną wymianę informacji między klientem a serwerem. Jest to przydatne narzędzie do zarządzania sesjami użytkowników i zapewnienia bezpieczeństwa aplikacji. |
| 5. | RESTful API | Stworzenie RESTful API pozwoli na komunikację między frontendem a backendem oraz umożliwi integrację z innymi systemami lub usługami. Django i Flask oferują narzędzia do tworzenia API. |

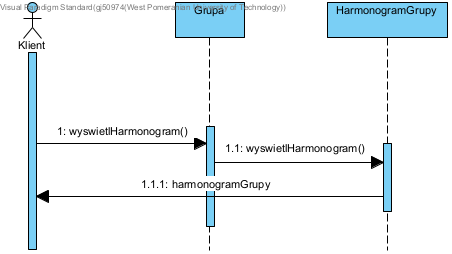
## 9.3 Diagramy UML

### 9.3.1 Diagram(-y) klas

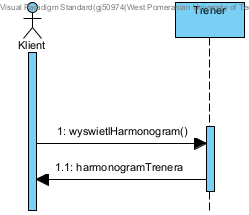


### 9.3.2 Diagram(-y) czynności

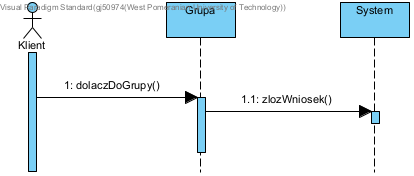
### 9.3.3 Diagramy sekwencji



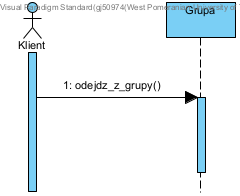
*1. Wyświetl dostępność grupy*

**

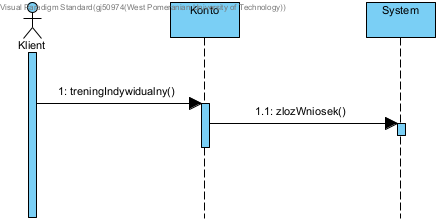
*2. Wyświetl dostępność trenera*

**

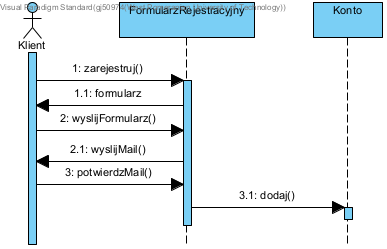
*3. Zgłoś chęć dołączenia do grupy*

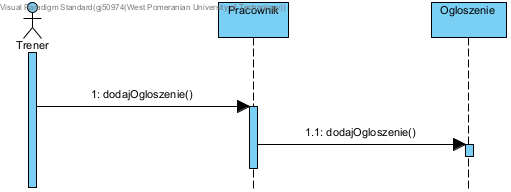
**

*4. Opuść grupę*

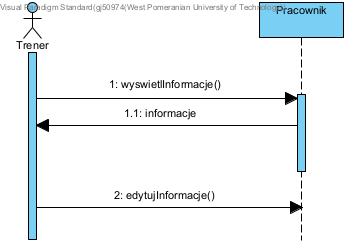
**

*5. Zgłoś chęć uczestnictwa w treningu indywidualnym*

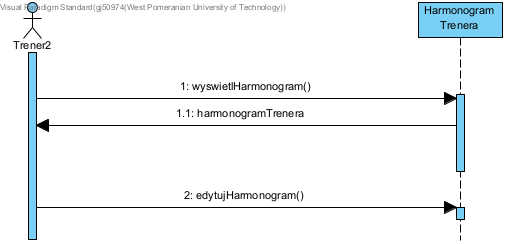
*6. Załóż konto*

**

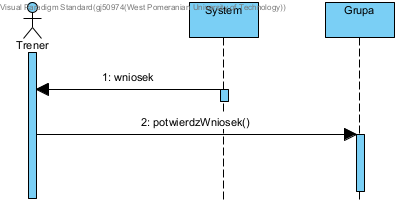
*7. Dodaj ogłoszenie*

**

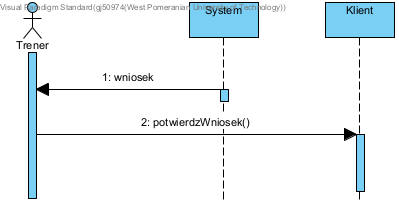
*8. Zmień informacje trenera(np. cenę treningów indywidualnych)*

**

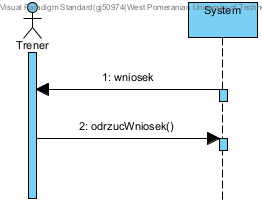
*9. Zmień dostępność treningów indywidualnych*

**

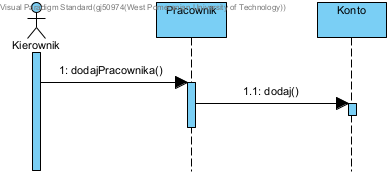
*10. Potwierdź wniosek o dodanie do grupy*

**

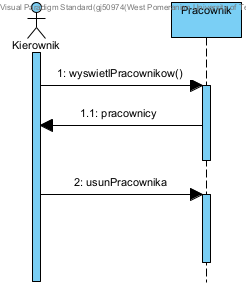
*11. Potwierdź wniosek o uczestniczenie w treningu indywidualnym*

**

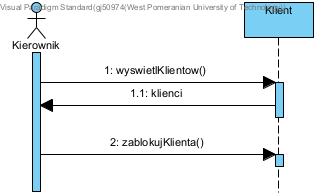
*12. Odrzuć wniosek o dołączenie do grupy lub uczestniczenie w treningu indywidualnym*

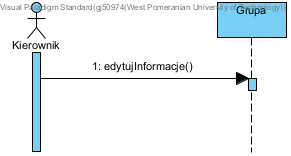
**

*13. Dodaj konto pracownika*

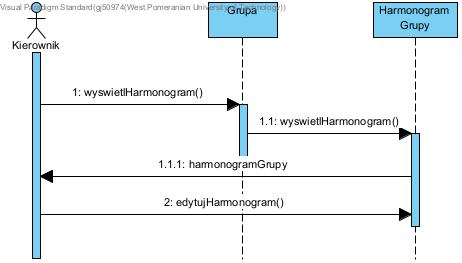
**

*14. Usuń konto pracownika*

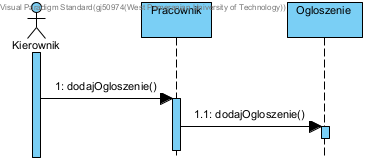
*15. Zablokuj konto klienta*

**

*16. Zmień detale grupy*

**

*17. Zmień dostępność grupy*

**

*18. Dodaj ogłoszenie*

### 9.3.4 Inne diagramy

co najmniej trzy – komponentów, rozmieszczenia, maszyny stanowej itp.

Obraz zawierający tekst, zrzut ekranu, diagram, linia

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, diagram, Czcionka

Opis wygenerowany automatycznie

## 9.4 Charakterystyka zastosowanych wzorców projektowych

informacja opisowa wspomagana diagramami (odsyłaczami do diagramów UML); jeśli wykorzystano wzorce projektowe, to należy wykazać dwa z nich

uwaga – wzorce projektowe nie są omawiane na wykładach!

Punkt opcjonalny!!!!!!!!

## 9.5 Projekt bazy danych

### 9.5.1 Schemat

w trzeciej formie normalnej; jeśli w innej to umieć uzasadnić wybór

### 9.5.2 Projekty szczegółowe tabel

w zależności, czy następujące elementy są widoczne na schemacie b.d.: nazwa tabeli, nazwy pól, typ danych, wartości NULL, klucz główny, klucz obcy –

- jeśli TAK: i nie ma potrzeby pokazania dodatkowych elementów b.d., to ten punkt może być pusty,

- jeśli NIE: to podać te elementy, których nie widać na schemacie.

dodatkowymi elementami mogą być np. triggery, procedury, funkcje, indeksy, użytkownicy, role.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nazwa Tabeli** | **Kolumna** | **Typ Danych** | **NULL** | **Klucz Główny** | **Klucz Obcy** |
| Konto | Id | Int | Nie | Tak | Nie |
| Nazwisko | Varchar | Nie | Nie | Nie |
| Adres | Varchar | Nie | Nie | Nie |
| NumerTelefonu | Varchar | Tak | Nie | Nie |
| KierownikID | Int | Nie | Nie | Tak |
| KlientID | Int | Nie | Nie | Tak |
| Klient | Id | Int | Nie | Tak | Nie |
| LimitKredytowy | Decimal | Nie | Nie | Nie |
| Pracownik | Id | Int | Nie | Tak | Nie |
| DataZatrudnienia | Date | Nie | Nie | Nie |
| Wyksztalcenie | Varchar | Tak | Nie | Nie |
| NumerKonta | Varchar | Nie | Nie | Nie |
| Wyplata | Decimal | Nie | Nie | Nie |
| Premia | Decimal | Tak | Nie | Nie |
| Harmonogram | Id | Int | Nie | Tak | Nie |
| Data | Date | Nie | Nie | Nie |
| Dlugosc | Time | Nie | Nie | Nie |
| Itd…. |  |  |  |  |  |

## 9.6 Projekt interfejsu użytkownika

co najmniej dla głównej funkcjonalności programu – w razie wątpliwości, uzgodnić z prowadzącym zajęcia

### 9.6.1 Lista głównych elementów interfejsu

1. Widok główny (start)

2. Zaloguj

3. Załóż konto

4. Informacje o trenerach

5. Informacje o treningach

6. Wpisy i ogłoszenia

7. Informacje kontaktowe

8. Widok funkcjonalności Kierownika

9. Widok funkcjonalności Trenera

10. Widok funkcjonalności klienta

11. Harmonogram

12. Widok generowania raportu

### 9.6.2 Przejścia między głównymi elementami

np. storyboard, schemat blokowy lub inna notacja

Obraz zawierający tekst, diagram, linia, zrzut ekranu

Opis wygenerowany automatycznie

### 9.6.3 Projekty szczegółowe poszczególnych elementów

Obraz zawierający tekst, zrzut ekranu, design

Opis wygenerowany automatycznieObraz zawierający zrzut ekranu, tekst, Prostokąt

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Prostokąt, diagram

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, wyświetlacz, numer

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, dokument

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, diagram, Równolegle

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, wyświetlacz, diagram

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, wyświetlacz, diagram

Opis wygenerowany automatycznie

## 9.7 Procedura wdrożenia

jeśli informacje w harmonogramie nie są wystarczające (a zapewne nie są)

# Dokumentacja dla użytkownika

Opcjonalnie – dla chętnych

Na podstawie projektu docelowej aplikacji, a nie zaimplementowanego prototypu architektury

4-6 stron z obrazkami (np. zrzuty ekranowe, polecenia do wpisania na konsoli, itp.)

* pisana językiem odpowiednim do grupy odbiorców – czyli najczęściej nie do informatyków
* może to być przebieg krok po kroku obsługi jednej głównej funkcji systemu, kilku mniejszych, instrukcja instalacji lub innej pomocniczej czynności.

# Podsumowanie

## Szczegółowe nakłady projektowe członków zespołu

tabela (kolumny to osoby, wiersze to działania) pokazująca, kto ile czasu poświęcił na projekt oraz procentowy udział każdej osoby w danym zadaniu oraz wiersz podsumowania – procentowy udział każdej osoby w skali całego projektu

|  |  |  |  |
| --- | --- | --- | --- |
| Działanie | Jakub | Vladislav | Kacper |
| 1. Wprowadzenie |  |  | 1 |
| 5.1 Charakterystyka ogólna |  |  | 1 |
| 5.2.1 Lista wymagań | 0.2 | 0.2 | 0.6 |
| 5.2.2 Diagramy przypadków użycia | 1 |  |  |
| 5.2.3 Szczegółowy opis wymagań | 0.(3) | 0.(3) | 0.(3) |
| 5.3 Wymagania niefunkcjonalne |  | 1 |  |
| 9.1 Opis architektury systemu |  | 1 |  |
| 9.2 Technologie implementacji systemu |  |  | 1 |
| 9.3.1 Diagramy klas | 1 |  |  |
| 9.3.2 Diagramy czynności |  |  |  |
| 9.3.3 Diagramy sekwencji | 1 |  |  |
| 9.3.4 Inne diagramy |  |  | 1 |
| 9.6.1 Lista głównych elementów interfejsu |  |  | 1 |
| 9.6.2 Przejścia między głównymi elementami |  |  | 1 |
| 9.6.3 Projekty szczegółowe poszczególnych elementów | 0.5 | 0.5 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

//TODO

Dodać być może wagę do każdego punktu, niektóre są dużo bardziej pracochłonne niż inne

# Inne informacje

przydatne informacje, które nie zostały ujęte we wcześniejszych punktach

//TODO

9.5.2 Projekty szczegółowe tabel

8.1

Konto nie ma hasla

Nie ma reprezentacji checi w diagramie klas

Niespojnosci co do realizacji platnosci

Dodatek I premia zbedne w diagramie klas

Przynaleznosc klienta do grupy w diagramie klas

Tytuły w diagramach sekwencji (I innych uzupełnić)

Trener jako obiekt a nie aktor gdy jest po srodku

Scenariusze alternatywne gdy klient juz jest w jakiejs grupie na przyklad

Ogłoszenie na diagramie klas dodać

Scenariusze przypadkow uzycia do przejrzenia ponownego

Punkt “Inne diagramy” do poprawki

Storyboard tych wireframów wzbogacić o te ktorych nie zwizualizowalismy a maja byc obecne

Zdefiniować konkretnie jakie raporty chcemy generować

Nie usuwać kont, a je dezaktywować I zawrzeć to w archiwum

Klasa zajęcia do dodania

Google Calendar, wzmiankę zrealizować albo usunąć