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

Należy pozostawić wszelkie nagłówki tego dokumentu, a umieszczać treść w odpowiednich miejscach zamiast obecnych objaśnień.

Stronę tytułową można sformatować w dowolny sposób, ale należy pozostawić zawartość informacyjną w układzie pokazanym powyżej.

Praca powinna zostać złożona wyłącznie w formacie pdf. Przed wygenerowaniem ostatecznej wersji należy zaktualizować spis treści – wyświetlane dwa poziomy.

Niniejszą informację należy również usunąć z wersji końcowej.

# Spis treści

Spis treści

[1. Spis treści 1](#_Toc164798562)

[2. Odnośniki do innych źródeł 3](#_Toc164798563)

[3. Słownik pojęć 3](#_Toc164798564)

[4. Wprowadzenie 3](#_Toc164798565)

[4.1 Cel dokumentacji 3](#_Toc164798566)

[4.2 Przeznaczenie dokumentacji 4](#_Toc164798567)

[4.3 Opis organizacji lub analiza rynku 4](#_Toc164798568)

[4.4 Analiza SWOT organizacji 5](#_Toc164798569)

[5. Specyfikacja wymagań 5](#_Toc164798570)

[5.1 Charakterystyka ogólna 5](#_Toc164798571)

[5.1.1 Definicja produktu 5](#_Toc164798572)

[5.1.2 Podstawowe założenia 5](#_Toc164798573)

[5.1.3 Cel biznesowy 6](#_Toc164798574)

[5.1.4 Użytkownicy 6](#_Toc164798575)

[5.1.5 Korzyści z systemu 6](#_Toc164798576)

[5.1.6 Ograniczenia projektowe i wdrożeniowe 6](#_Toc164798577)

[5.2 Wymagania funkcjonalne 6](#_Toc164798578)

[5.2.1 Lista wymagań 6](#_Toc164798579)

[5.2.2 Diagramy przypadków użycia 7](#_Toc164798580)

[5.2.3 Szczegółowy opis wymagań 9](#_Toc164798581)

[5.3 Wymagania niefunkcjonalne 14](#_Toc164798582)

[6. Zarządzanie projektem 16](#_Toc164798583)

[6.1 Zasoby ludzkie 16](#_Toc164798584)

[6.2 Harmonogram prac 16](#_Toc164798585)

[6.3 Etapy/kamienie milowe projektu 16](#_Toc164798586)

[7. Zarządzanie ryzykiem 16](#_Toc164798587)

[7.1 Lista czynników ryzyka 16](#_Toc164798588)

[7.2 Ocena ryzyka 16](#_Toc164798589)

[7.3 Plan reakcji na ryzyko 16](#_Toc164798590)

[8. Zarządzanie jakością 17](#_Toc164798591)

[8.1 Scenariusze i przypadki testowe 17](#_Toc164798592)

[9. Projekt techniczny 17](#_Toc164798593)

[9.1 Opis architektury systemu 17](#_Toc164798594)

[9.2 Technologie implementacji systemu 17](#_Toc164798595)

[9.3 Diagramy UML 18](#_Toc164798596)

[9.3.1 Diagram(-y) klas 18](#_Toc164798597)

[9.3.2 Diagram(-y) czynności 18](#_Toc164798598)

[9.3.3 Diagramy sekwencji 18](#_Toc164798599)

[9.3.4 Inne diagramy 18](#_Toc164798600)

[9.4 Charakterystyka zastosowanych wzorców projektowych 18](#_Toc164798601)

[9.5 Projekt bazy danych 19](#_Toc164798602)

[9.5.1 Schemat 19](#_Toc164798603)

[9.5.2 Projekty szczegółowe tabel 19](#_Toc164798604)

[9.6 Projekt interfejsu użytkownika 19](#_Toc164798605)

[9.6.1 Lista głównych elementów interfejsu 19](#_Toc164798606)

[9.6.2 Przejścia między głównymi elementami 19](#_Toc164798607)

[9.6.3 Projekty szczegółowe poszczególnych elementów 19](#_Toc164798608)

[9.7 Procedura wdrożenia 19](#_Toc164798609)

[10. Dokumentacja dla użytkownika 20](#_Toc164798610)

[11. Podsumowanie 20](#_Toc164798611)

[Szczegółowe nakłady projektowe członków zespołu 20](#_Toc164798612)

[12. Inne informacje 20](#_Toc164798613)

# 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.

###### Dodać nazwę systemu

### 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: Brak trenera dla danej grupy treningowej
* Przebieg:

1. Kierownik loguje się do systemu administracyjnego.
2. Kierownik wybiera opcję „Dodaj nowego trenera”.
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: 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 potwierdza decyzję usuwania.
5. 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:
* 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:
* 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:
* 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:
* Przebieg:

1. Kierownik loguje się do panelu administracyjnego.
2. Kierownik wybiera opcję „Dodaj ogłoszenie”.
3. System wyświetla formularz ogłoszenia
4. Kierownik wpisuje treść ogłoszenia oraz określa grupę docelową (np. trenerzy, klienci).
5. 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 posiada przypisaną grupę treningową
    - Przebieg:
      1. Trener loguje się do swojego konta.
      2. Trener przechodzi do sekcji „Ogłoszenia” lub „Aktualności”.
      3. System wyświetla formularz ogłoszenia
      4. Trener tworzy nowe ogłoszenie dotyczące treningów dla swojej grupy.
      5. 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: brak
    - Przebieg
      1. Trener loguje się do swojego konta.
      2. Trener przechodzi do sekcji „Treningi indywidualne”.
      3. System wyświetla harmonogram zajęć trenera
      4. Trener wybiera rodzaj treningu i określa cenę oraz dostępność terminów.
      5. 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: w bazie istnieje przynajmniej jeden klient
    - Przebieg:
      1. Trener loguje się do panelu swojego konta.
      2. Trener przechodzi do sekcji „Grafik treningów”.
      3. System wyświetla harmonogram zajęć trenera
      4. Trener dodaje nowe terminy treningów indywidualnych, uwzględniając swoją dostępność.
      5. 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: klient zgłosił chęć udziału w treningu
    - Przebieg:
      1. Trener otrzymuje powiadomienie o nowej rezerwacji treningu.
      2. Trener loguje się do panelu swojego konta.
      3. Trener przechodzi do sekcji „Rezerwacje”.
      4. System wyświetla prośby rezerwacji zajęć
      5. Trener akceptuje rezerwację klienta dla konkretnego treningu.
      6. 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. Klient loguje się do swojego konta.
      2. Klientprzechodzi do sekcji „Treningi” lub „Trenerzy”.
      3. System wyświetla informacje o treningach lub trenerach
      4. Klientprzegląda dostępne treningi lub informacje o trenerach, takie jak specjalizacje i opinie innych klientów.
      5. 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 nie należy do grupy treningowej
* Przebieg:
  + - 1. Klient loguje się do swojego konta.
      2. Klientprzechodzi do sekcji „Moje grupy treningowe”.
      3. System wyświetla listę grup
      4. Klientwybiera opcję „Dołącz” obok odpowiedniej grupy.
      5. 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 należy do grupy treningowej
* Przebieg:
  + - 1. Klient loguje się do swojego konta.
      2. Klientprzechodzi do sekcji „Moje grupy treningowe”.
      3. System wyświetla grupy do których należy klient
      4. Klientwybiera opcję „Odejdź” obok odpowiedniej grupy.
      5. 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 nie jest zapisany na treningi indywidualne u trenera
* Przebieg:
  + - 1. Klient loguje się do swojego konta.
      2. Klientprzechodzi do sekcji „Treningi indywidualne”.
      3. System wyświetla listę trenerów i ich dostępnych terminów
      4. Klientwybiera trenera i termin treningu.
      5. Klientwysyła prośbę o rezerwację treningu indywidualnego.
      6. 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*

# Projekt techniczny

//TODO

Use casy, logowanie jako nowa funkcja?, w innych podpunktach jako warunek początkowy //Jakub

9.6.1 //Kacper

9.6.2 //Kacper

9.6.3 //Vlad/Jakub

##### 9.3.4 //Kacper

9.1 //Vladislav

9.3.3 //Jakub\*\*\*

## 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.

// TODO: diagramu komponentów 9.3.4, żeby dodać link

1. 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
     + // TODO
2. 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.
3. Komponenty i technologie   
   W sekcji 9.2, szczegółowo opisujemy języki programowania i technologie wykorzystane do realizacji poszczególnych komponentów.
4. // TODO: Wybrane wzorce architektoniczne

## 9.2 Technologie implementacji systemu

tabela z listą wykorzystanych technologii, każda z uzasadnieniem

//tabelka z tablicy

//język programowania (kilka?), platformy, system operacyjny (konkretnie, nie jednym zdaniem), opisywać konkretnie, technicznie

###### Zakres dopuszczalnych systemow dla uzytkownikow koncowych

System baz danych

Dodatkowe urządzenia? Ekrany, kioski, skanery?------

Uzasadnienie do każdej technologii

„jest wydajna, bezpieczna, dobrze udokumentowana” albo „jest to technologia nam znana”

|  |  |  |
| --- | --- | --- |
| 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

każdy diagram ma mieć tytuł oraz ma być na osobnej stronie

diagramy przypadków użycia umieszczone w punkcie , a nie tutaj.

//wkleić od Jakuba

### 9.3.1 Diagram(-y) klas

1 lub więcej

### 9.3.2 Diagram(-y) czynności

###### Co najmniej 1 dla zespołów 2-osobowych, więcej dla liczniejszych

### 9.3.3 Diagramy sekwencji

co najmniej 5, w tym co najmniej 1 przypadek użycia zilustrowany kilkoma diagramami (dla zespołów 2-osobowych, dla liczniejszych więcej)

dla nas 7-8

najlepiej żeby odpowiadały naszym use case’om

### 9.3.4 Inne diagramy

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

###### Mogą wyjść bardzo proste, no problem

Maszyna stanowa – wybrać jakiś obiekt, jego możliwe stany, jakiś charakterystyczny obiekt jeśli się da

Nie dawać diagramu komunikacji z przekształcenia sekwencji

## 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.

## 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

okien, stron, aktywności (Android)

### 9.6.2 Przejścia między głównymi elementami

np. storyboard, schemat blokowy lub inna notacja

START 🡪 funkcjonalności główne🡪STOP

### 9.6.3 Projekty szczegółowe poszczególnych elementów

dla 5-7 głównych elementów (w zespołach 2-osobowych)

visual paradigm -> wireframe

///może być figma jak jesteś masochistą

Wstawiać przykładowe dane do pól wypełnialnych

Wybierać w miarę charakterystyczne

każdy element od nowej strony z następującą minimalną zawartością:

* numer – ID elementu
* nazwa – np. formularz danych produktu
* projekt graficzny – wystarczy schemat w narzędziu graficznym lub zrzut ekranu – z przykładowymi danymi (nie pusty!!!)
* opcjonalnie:
* opis – dodatkowe opcjonalne informacje o przeznaczeniu, obsłudze – jeśli nazwa nie będzie wystarczająco czytelna
* wykorzystane dane – jakie dane z bazy danych są wykorzystywane
* opis działania – tabela pokazująca m.in. co się dzieje po kliknięciu przycisku, wybraniu opcji z menu itp.

## 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 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Inne informacje

przydatne informacje, które nie zostały ujęte we wcześniejszych punktach