**Instytut Informatyki  
Wydział Nauk Ścisłych i Technicznych  
Uniwersytet Rzeszowski**

**Przedmiot:**

**Programowanie zespołowe**

**Dokumentacja projektu:**

***System do zarządzania zadaniami  
w firmie informatycznej***

**Wykonał:**

**Zespół projektowy różowy**

**Prowadzący: mgr inż. Adam Szczur**

**Rzeszów 2025**

Spis treści

[1. Zespół projektowy 3](#_Toc161677644)

[2. Specyfikacja projektu 3](#_Toc161677645)

[2.1. Cel projektu 3](#_Toc161677646)

[2.2. Zakres projektu 3](#_Toc161677647)

[2.2.1. Zasoby 3](#_Toc161677648)

[2.2.2. Produkty końcowe projektu 4](#_Toc161677649)

[2.2.3. Harmonogram realizacji projektu 4](#_Toc161677650)

[2.3. Wymagania stawiane aplikacji / systemowi 4](#_Toc161677651)

[2.4. Panele / zakładki systemu, które będą oferowały potrzebne funkcjonalności 4](#_Toc161677652)

[2.5.1. Raporty PDF 5](#_Toc161677653)

[2.5.2. Inne dokumenty 5](#_Toc161677654)

[2.6. Przepływ informacji w środowisku systemu 5](#_Toc161677655)

[2.7. Użytkownicy aplikacji i ich uprawnienia 5](#_Toc161677656)

[2.8. Interesariusze 6](#_Toc161677657)

[3. Diagramy UML 6](#_Toc161677658)

[3.1. Diagram przypadków użycia 6](#_Toc161677659)

[3.2. Diagramy aktywności 6](#_Toc161677660)

[3.3. Diagramy sekwencji 6](#_Toc161677661)

[4. Baza danych 6](#_Toc161677662)

[4.1. Diagram ERD 6](#_Toc161677663)

[4.2. Skrypt do utworzenia struktury bazy danych 6](#_Toc161677664)

[5. Wykorzystane technologie 6](#_Toc161677665)

[6. Interfejs aplikacji / systemu 7](#_Toc161677666)

# Zespół projektowy

Nazwa zespołu: różowi

Skład zespołu projektowego:

|  |  |  |
| --- | --- | --- |
| Imię i nazwisko | Pełniona rola | Zakres czynności / odpowiedzialności |
| Piotr Nowak | Lider/Baza danych | - Zarządzanie projektem, koordynacja prac zespołu - Projektowanie i implementacja bazy danych - Definiowanie schematu bazy danych i relacji - Zapewnienie integralności i bezpieczeństwa danych |
| Piotr Smoła | Backend | - Rozwój logiki w języku Java - Implementacja mechanizmów uwierzytelniania i autoryzacji - Integracja backendu z bazą danych MySQL |
| Łukasz Solecki | Baza danych | - Wsparcie w projektowaniu bazy danych - Pisanie i optymalizacja zapytań SQL - Zarządzanie migracjami bazy danych |
| Dawid Olko | Frontend | - Projektowanie i implementacja interfejsu użytkownika - Połączenie frontend z API backendu - Implementacja walidacji |

# Specyfikacja projektu

## Cel projektu

Głównym celem projektu jest opracowanie w pełni funkcjonalnego systemu do zarządzania zadaniami dla firmy informatycznej. System ten ma usprawnić procesy wewnętrzne firmy, w szczególności zarządzania projektami, zasobami ludzkimi oraz komunikacją między pracownikami. Jego wdrożenie przyniesie wymierne korzyści, które można opisać następująco:

* **Zwiększenie efektywności zarządzania zadaniami**: System umożliwi automatyzację przydzielania zadań, monitorowanie ich postępu oraz generowanie raportów zarządczych, co pozwoli na lepsze wykorzystanie zasobów ludzkich.
* **Usprawnienie komunikacji**: Centralizacja informacji i narzędzia do współpracy zmniejszą ryzyko nieporozumień oraz opóźnień, poprawiając koordynację działań zespołu.
* **Skrócenie czasu realizacji projektów**: Optymalizacja procesów i eliminacja zbędnych przestojów.
* **Minimalizacja błędów operacyjnych**: Automatyzacja rutynowych zadań i walidacja danych.
* **Optymalizacja kosztów operacyjnych**: Lepsze zarządzanie zasobami i automatyzacja procesów administracyjnych.

Ponadto, system będzie realizował cele technologiczne i jakościowe, takie jak:

* **Skalowalność i elastyczność**: Architektura systemu pozwoli na łatwe dodawanie nowych funkcjonalności oraz integrację z istniejącymi systemami informatycznymi firmy.
* **Bezpieczeństwo danych**: System zapewni ochronę danych w tym szyfrowanie haseł użytkowników.

System zostanie oparty na nowoczesnych technologiach, takich jak **Java** i baza danych **MariaDB** z naciskiem na modularność, intuicyjny interfejs użytkownika oraz możliwość generowania zaawansowanych raportów zarządczych. Wszystkie cele są zgodne z długoterminową strategią firmy i wspierają jej rozwój na konkurencyjnym rynku IT.

## Zakres projektu

### Zasoby

* **Zasoby ludzkie**:
  + Zespół programistów: 4 osoby, 10 godzin pracy tygodniowo przez 10 tygodni.
  + Zespół testerów: 4 osoby, 5 godzin pracy tygodniowo przez 2 tygodnie.
  + UI Designer: 5 godzin pracy.
* **Zasoby programowe**:
  + Darmowe narzędzia, takie jak **IntelliJ**, **GitHub**, **Jira** oraz inne wspierające rozwój systemu.

### Produkty końcowe projektu

W wyniku realizacji projektu powstaną:

* **System do zarządzania zadaniami w firmie informatycznej**, w tym:
  + Pakiet instalacyjny.
  + Kod źródłowy aplikacji.
  + Skrypty do utworzenia bazy danych.
* **Dokumentacja techniczna projektu**.
* **Podręcznik użytkownika dla wytworzonego systemu.**

Projekt zakłada również opracowanie harmonogramu realizacji, w tym wykresu Gantta, który określi zadania, daty ich realizacji, zależności między nimi oraz kamienie milowe.

### Harmonogram realizacji projektu

Harmonogram realizacji projektu obejmuje wszystkie zadania związane z projektem wraz z określeniem dat ich realizacji. Poniżej przedstawiono szczegółowy harmonogram uwzględniający kluczowe etapy projektu na podstawie dostępnych danych oraz logicznego rozszerzenia:

|  |  |
| --- | --- |
| Zakres dat | Zadanie |
| 26.02.2025 – 12.03.2025 | Opracowanie wizji projektu i przygotowanie środowiska pracy |
| 12.03.2025 – 26.03.2025 | Zaplanowanie i wykonanie GUI aplikacji |
| 26.03.2025 – 09.04.2025 | Zaprojektowanie schematu bazy danych oraz jej implementacja |
| 09.04.2025 – 30.04.2025 | Testy jednostkowe |
| 30.04.2025 – 14.05.2025 | Stworzenie biblioteki generującej raporty w PDF |
| 14.04.2025 – 28.04.2025 | Wykonanie brakujących elementów |
| 28.04.2025 – 04.06.2025 | Tworzenie dokumentacji technicznej i podręcznika użytkownika |

#### Wykres Gantta

## Wymagania stawiane aplikacji / systemowi

System powinien składać się z kilku kluczowych modułów, z których każdy realizuje określone funkcje:

* **Moduł administracji użytkownikami (role):**
  + Umożliwia administratorom tworzenie, edytowanie i usuwanie kont użytkowników.
  + Wspiera cztery role użytkowników:
    - **Administrator**: Pełna kontrola nad zadaniami, kontami i systemem.
    - **Kierownik**: Zarządza zespołami i projektami, oraz edytuje i przegląda zadania.
    - **Team Lider**: Zarządza zadaniami – może je tworzyć, przypisywać innym i edytować.
    - **Pracownik**: Może przeglądać wszystkie zadania i edytować tylko swoje.
  + Definiuje specyficzne uprawnienia dla każdej roli, zapewniając kontrolę dostępu do funkcji systemu.
* **Moduł raportów:**
  + Umożliwia generowanie raportów na podstawie danych zgromadzonych w systemie, takich jak informacje o zadaniach, projektach czy użytkownikach.
  + Pozwala użytkownikom na określenie parametrów raportów, np. zakresu dat, zespoły lub
  + Generuje raporty w formacie PDF, które są profesjonalnie sformatowane i możliwe do zapisania lub eksportu.
* **Współpraca z bazą danych**
* System musi przechowywać wszystkie dane (użytkownicy, zadania, raporty) w bazie danych.
* Zarządzanie połączeniami z bazą danych powinno być efektywne.
* System musi zapewniać integralność i spójność danych zapisanych w bazie.

## Panele / zakładki systemu, które będą oferowały potrzebne funkcjonalności

**Panel administratora**  
Główne narzędzie administratorów systemu umożliwiające wykonanie wszystkich czynności potrzebnych do zarządzania systemem, w tym:

* Zarządzanie użytkownikami: Dodawanie, edycja, usuwanie użytkowników oraz przydzielanie ról i grup
* **Zarządzanie zespołami:** Tworzenie zespołów, przypisywanie członków i liderów zespołów, powiązanie z projektami
* Zarządzanie projektami: Tworzenie, edycja i usuwanie projektów, przypisywanie kierowników
* Zarządzanie zadaniami: Tworzenie, edycja, usuwanie i przeglądanie wszystkich zadań w systemie
* Aktywności systemowe: Przeglądanie logów systemowych i działań użytkowników
* Raporty: Generowanie raportów dotyczących projektów, zespołów i użytkowników
* Ustawienia systemowe: Konfiguracja parametrów systemu, kopie zapasowe bazy danych

**Panel kierownika**  
Panel umożliwiający kierownikom zarządzanie swoimi zespołami i projektami, oferujący funkcje takie jak:

* Zarządzanie projektami: Przeglądanie i edycja projektów przypisanych do kierownika
* Zarządzanie zespołami: Tworzenie zespołów, przypisywanie członków do zespołów w ramach swoich projektów
* Monitorowanie zadań: Śledzenie postępów realizacji zadań przypisanych do projektów
* Przegląd pracowników: Podgląd członków zespołów przypisanych do projektów
* Raporty: Generowanie statystyk i podsumowań dotyczących zespołów i projektów

*(Kierownik nie może tworzyć ani przypisywać zadań)*

**Panel team lidera**  
Panel dla liderów zespołów do zarządzania zadaniami w obrębie swojego zespołu:

* Moje zadania: Przegląd zadań przypisanych bezpośrednio do lidera
* Zadania zespołu: Przeglądanie wszystkich zadań przypisanych do zespołu
* Zarządzanie zadaniami: Tworzenie nowych zadań, przypisywanie zadań członkom zespołu, edycja statusów
* Zarządzanie pracownikami: Przeglądanie i zarządzanie członkami zespołu
* Raporty zespołu: Generowanie raportów dotyczących członków zespołu i przypisanych zadań

*(Team Lider nie zarządza projektami ani zespołami w całości)*

**Panel użytkownika**  
Panel dla zwykłych użytkowników, umożliwiający im zarządzanie swoimi zadaniami i dostęp do osobistych statystyk:

* Moje zadania: Przegląd listy przydzielonych zadań wraz z ich szczegółami
* Zadania w zespole: Przeglądanie zadań innych członków zespołu
* Aktualizacja zadań: Zmiana statusu własnych zadań
* Ustawienia: Zmiana preferencji konta

**Wspólne komponenty systemu**

**Okno ustawień**  
Okno umożliwiające użytkownikom konfigurację preferencji systemowych:

* Preferencje interfejsu: Wybór motywu (jasny/ciemny)
* Domyślny widok: Wybór zakładki, która będzie pokazywana po zalogowaniu
* Ustawienia konta: Zmiana hasła oraz adresu email, ustawienie podpowiedzi do hasła

**Panel logowania i rejestracji**

* Logowanie: Uwierzytelnianie użytkowników
* Rejestracja: Tworzenie nowych kont użytkowników (wymaga zatwierdzenia przez administratora)
* Odzyskiwanie hasła: Wsparcie dla użytkowników, którzy zapomnieli hasła

**Generowanie raportów**  
Funkcjonalność dostępna dla różnych typów użytkowników z różnym poziomem uprawnień:

* Typy raportów: Dostosowane do roli użytkownika (np. raporty projektowe dla kierowników, raporty zespołów dla liderów)
* Filtry: Dostosowywanie zakresu raportów (okres czasu, zespół, projekt)
* Eksport: Zapisywanie raportów do plików PDF
* Opcje wyświetlania: Wybór informacji zawartych w raporcie (zadania, członkowie, statystyki)

System uwzględnia hierarchię uprawnień, gdzie każdy poziom użytkownika ma dostęp do funkcji odpowiednich dla swojej roli, przy czym administrator ma dostęp do wszystkich funkcji systemu.

* 1. Typy wymaganych dokumentów w projekcie oraz dostęp do nich

### Raporty PDF

#### Rodzaje raportów PDF

**Struktura Zespołów**  
Raport prezentuje szczegółową strukturę wybranych zespołów, ich skład osobowy oraz przypisane zadania.

Dane w raporcie:  
 – Lista wybranych zespołów  
 – Przypisanie zespołów do projektów  
 – Skład osobowy zespołów (członkowie)  
 – Zadania przypisane do zespołów wraz ze statusami

Opcje zarządzania danymi:  
 – Wybór konkretnych zespołów do analizy  
 – Włączanie/wyłączanie informacji o członkach zespołów  
 – Włączanie/wyłączanie informacji o zadaniach zespołów

Zastosowanie: Raport wspiera administratorów i kierowników w zarządzaniu strukturą organizacyjną projektów, zapewniając przejrzysty widok przydziału zasobów ludzkich.  
Dostęp: Administrator, Kierownik, Team Leader

**Użytkownicy Systemu**  
Raport przedstawia informacje o użytkownikach systemu, ich rolach, przynależności do zespołów i grup.

• Dane w raporcie:  
 – Lista użytkowników z podziałem na role  
 – Imię i nazwisko użytkowników  
 – Adresy email  
 – Przynależność do zespołów  
 – Przynależność do grup zawodowych (np. Deweloperzy, Testerzy, Analitycy)

• Opcje zarządzania danymi:  
 – Filtrowanie według ról użytkowników (Administrator, Kierownik, Team Leader, Pracownik)  
 – Filtrowanie według grup zawodowych  
 – Wybór wyświetlania informacji o przynależności do zespołów

• Zastosowanie: Raport pomaga administratorom w zarządzaniu użytkownikami, ich uprawnieniami oraz przynależnością organizacyjną.  
• Dostęp: Administrator

**Przegląd Projektów**  
Raport dostarcza kompleksowy przegląd projektów w systemie, ich statusu, przypisanych zespołów i zadań.

Dane w raporcie:  
 – Nazwa i opis projektów  
 – Daty rozpoczęcia i zakończenia projektów  
 – Kierownicy projektów  
 – Przypisane zespoły  
 – Lista zadań wraz ze statusami  
 – Statystyki ukończenia zadań (procent ukończenia)

Opcje zarządzania danymi:  
 – Wybór konkretnych projektów  
 – Określenie zakresu dat dla projektów  
 – Włączanie/wyłączanie informacji o zadaniach  
 – Włączanie/wyłączanie statystyk projektów

Zastosowanie: Raport jest kluczowy dla kierowników i administratorów, umożliwiając monitorowanie postępu projektów, identyfikację opóźnień oraz planowanie zasobów.  
Dostęp: Administrator, Kierownik

**Zadania Zespołu**  
Raport prezentuje zadania przypisane do zespołu, którym zarządza Team Leader, z informacjami o statusach i przypisanych użytkownikach.

Dane w raporcie:  
 – Lista zadań zespołu  
 – Statusy zadań (Nowe, W toku, Zakończone)  
 – Priorytety zadań  
 – Użytkownicy przypisani do zadań  
 – Statystyki zadań według statusów

Opcje zarządzania danymi:  
 – Wybór konkretnych zespołów (jeśli Team Leader zarządza wieloma zespołami)  
 – Filtrowanie według statusu zadań  
 – Filtrowanie według przypisanego użytkownika  
 – Włączanie/wyłączanie statystyk

Zastosowanie: Raport wspiera Team Leadera w śledzeniu postępu prac zespołu, identyfikacji zadań wymagających uwagi oraz równomiernym rozdzielaniu obciążenia pracą.  
Dostęp: Team Leader

**Członkowie Zespołu**  
Raport szczegółowo opisuje skład osobowy zespołów zarządzanych przez Team Leadera.

Dane w raporcie:  
 – Lista zespołów  
 – Przypisanie zespołów do projektów  
 – Członkowie zespołów wraz z danymi kontaktowymi

Opcje zarządzania danymi:  
 – Wybór konkretnych zespołów  
 – Filtrowanie według grup zawodowych  
 – Filtrowanie według określonych użytkowników

Zastosowanie: Raport pomaga Team Leaderowi w zarządzaniu składem zespołu, planowaniu zadań oraz komunikacji z członkami zespołu.  
Dostęp: Team Leader

**Eksport raportów**  
Wszystkie raporty mogą być eksportowane do formatu PDF i zapisywane na dysku użytkownika. System automatycznie generuje nazwy plików zawierające typ raportu oraz datę i godzinę wygenerowania, co ułatwia organizację dokumentów.

### Inne dokumenty

## Przepływ informacji w środowisku systemu

System zarządzania zadaniami w firmie informatycznej zostanie zaprojektowany jako **system scentralizowany**, oparty na centralnej bazie danych. Decyzja ta opiera się na założeniu, że firma posiada jedną siedzibę lub, w przypadku kilku oddziałów, wszystkie pracują nad wspólnymi projektami i zadaniami, co uzasadnia użycie jednej, współdzielonej bazy danych. Główną składową systemu jest **centralna baza danych**, która przechowuje wszystkie dane dotyczące użytkowników, zadań, projektów oraz raportów. Taki model zapewnia łatwy dostęp do danych, spójność informacji oraz uproszczenie zarządzania i utrzymania systemu.

**Przepływ informacji między modułami aplikacji**

W systemie wyróżniamy kilka kluczowych modułów, między którymi następuje wymiana informacji:

* **Moduł administracji użytkownikami** przekazuje dane o użytkownikach (np. ID, role) do **modułu zarządzania zadaniami**, umożliwiając przypisywanie zadań do konkretnych osób.
* **Moduł zarządzania zadaniami** dostarcza informacje o zadaniach (np. statusy, priorytety, terminy) do **modułu raportów**, co pozwala na generowanie raportów o postępach prac czy obciążeniu pracowników.
* **Moduł raportów** może zwracać wygenerowane raporty do modułu zarządzania zadaniami w celu ich wyświetlenia użytkownikom.

Dzięki temu przepływowi informacji, system działa spójnie i efektywnie, wspierając zarządzanie zadaniami w firmie informatycznej.

## Użytkownicy aplikacji i ich uprawnienia

W systemie zarządzania zadaniami w firmie informatycznej wyróżniamy cztery główne role użytkowników, z których każda posiada specyficzne uprawnienia dostosowane do ich funkcji i odpowiedzialności. Poniżej przedstawiono role oraz ich uprawnienia w czytelnej formie:

**Administrator**  
Rola przeznaczona dla osób odpowiedzialnych za administrację i utrzymanie systemu. Administrator ma pełną kontrolę nad aplikacją.

* Zarządzanie użytkownikami (tworzenie, edycja, usuwanie kont).
* Zarządzanie rolami i uprawnieniami (definiowanie ról i przypisywanie uprawnień).
* Tworzenie, edycja i usuwanie zadań.
* Przeglądanie wszystkich zadań w systemie.
* Przypisywanie zadań dowolnym użytkownikom.
* Generowanie raportów dla całego systemu.
* Konfiguracja ustawień systemowych.

**Kierownik**  
Rola dla osób nadzorujących pracę zespołów, z uprawnieniami skoncentrowanymi na zarządzaniu zespołami i projektami.

* Przeglądanie zadań przypisanych do zespołu.
* Edycja zadań zespołowych.
* Zarządzanie zespołami (dodawanie, modyfikacja, przypisywanie członków).
* Zarządzanie projektami (tworzenie, edycja).
* Generowanie raportów dla zespołu i projektów.

**Team lider**  
Rola dla liderów zespołów odpowiedzialnych za zarządzanie zadaniami w obrębie swojego zespołu.

* Tworzenie zadań w ramach zespołu.
* Przypisywanie zadań członkom zespołu.
* Edycja zadań zespołowych.
* Przeglądanie zadań realizowanych przez zespół.

**Użytkownik**  
Rola dla pracowników realizujących zadania, z podstawowymi uprawnieniami do zarządzania własnymi obowiązkami.

* Przeglądanie własnych zadań.
* Aktualizacja statusu własnych zadań.
* Dodawanie komentarzy i załączników do własnych zadań.
* Przegląd własnych statystyk i raportów osobistych.

Powyższe role i uprawnienia zapewniają efektywne zarządzanie zadaniami w firmie informatycznej, dostosowane do hierarchii i potrzeb organizacji.

## Interesariusze

#### Interesariusze wewnętrzni

* **Zarząd firmy**  
  Osoby decyzyjne, które zatwierdzają projekt, jego cele biznesowe oraz budżet. Ich wsparcie jest kluczowe dla powodzenia projektu, ponieważ określają strategiczne priorytety i zapewniają zasoby.
* **Kierownicy działów**  
  Osoby zarządzające zespołami, które będą korzystać z systemu do przydzielania, monitorowania i raportowania zadań. Są zainteresowani usprawnieniem procesów zarządzania, co zwiększa efektywność ich zespołów.
* **Pracownicy firmy**  
  Użytkownicy końcowi systemu, którzy będą używać go do zarządzania swoimi codziennymi zadaniami. Ich satysfakcja i opinie są istotne dla skutecznego wdrożenia i adopcji systemu.
* **Dział IT**  
  Zespół odpowiedzialny za wdrożenie, utrzymanie i wsparcie techniczne systemu. Odpowiadają za jego integrację z istniejącą infrastrukturą IT, bezpieczeństwo danych i niezawodność działania.
* **Zespół projektowy**  
  Osoby zaangażowane w rozwój i realizację projektu, takie jak programiści, analitycy, testerzy czy projektanci. Ich praca bezpośrednio przekłada się na jakość i funkcjonalność dostarczonego systemu.

#### Interesariusze zewnętrzni

* **Dostawcy oprogramowania**  
  Firmy dostarczające narzędzia, biblioteki lub usługi wykorzystywane w projekcie, np. dostawcy baz danych czy bibliotek programistycznych. Jakość ich produktów i wsparcie techniczne mają wpływ na realizację projektu.
* **Klienci firmy**  
  Osoby lub organizacje, dla których firma realizuje projekty informatyczne. Mogą być zainteresowani systemem w kontekście śledzenia postępów prac nad ich zleceniami, co może wpłynąć na ich zadowolenie i relacje z firmą.
* **Organy regulacyjne**  
  Instytucje nadzorujące przestrzeganie przepisów prawnych, np. dotyczących ochrony danych osobowych (RODO). System musi być zgodny z tymi regulacjami, aby uniknąć problemów prawnych i zapewnić bezpieczeństwo danych.

# Diagramy UML

## Diagram przypadków użycia

Obraz zawierający tekst, diagram, rysowanie, szkic

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

## Diagramy aktywności

Obraz zawierający diagram, tekst, Plan, Rysunek techniczny

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Obraz zawierający tekst, diagram, Plan, Rysunek techniczny

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Obraz zawierający tekst, diagram, Plan, Rysunek techniczny

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

## Diagramy sekwencji

Obraz zawierający tekst, Równolegle, paragon, numer

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Obraz zawierający tekst, diagram, linia, Równolegle

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.Obraz zawierający tekst, diagram, linia, Równolegle

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

# Baza danych

## Diagram ERD

## 

Obraz zawierający tekst, diagram, linia, Równolegle

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

## Skrypt do utworzenia struktury bazy danych

**CREATE** **DATABASE** IF **NOT** **EXISTS** it\_task\_management;

USE it\_task\_management;

**CREATE** **TABLE** notifications (

id INT(11) **NOT** **NULL** AUTO\_INCREMENT,

user\_id INT(11) **NOT** **NULL**,

content TEXT **NOT** **NULL**,

is\_read TINYINT(1) **DEFAULT** 0,

created\_at **TIMESTAMP** **NOT** **NULL** **DEFAULT** **current\_timestamp**(),

**PRIMARY** **KEY** (id),

**KEY** (user\_id)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8mb4;

**CREATE** **TABLE** projects (

id INT(11) **NOT** **NULL** AUTO\_INCREMENT,

project\_name VARCHAR(100) **NOT** **NULL**,

description TEXT **DEFAULT** **NULL**,

start\_date DATE **DEFAULT** **NULL**,

end\_date DATE **DEFAULT** **NULL**,

manager\_id INT(11) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (id),

**KEY** (manager\_id)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8mb4;

**CREATE** **TABLE** reports (

id INT(11) **NOT** **NULL** AUTO\_INCREMENT,

report\_name VARCHAR(100) **NOT** **NULL**,

report\_type VARCHAR(50) **NOT** **NULL**,

report\_scope TEXT **DEFAULT** **NULL**,

created\_by INT(11) **NOT** **NULL**,

created\_at DATETIME **DEFAULT** **NULL**,

exported\_file VARCHAR(255) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (id),

**KEY** (created\_by)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8mb4;

**CREATE** **TABLE** roles (

id INT(11) **NOT** **NULL** AUTO\_INCREMENT,

role\_name VARCHAR(50) **NOT** **NULL**,

permissions TEXT **DEFAULT** **NULL**,

**PRIMARY** **KEY** (id)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8mb4;

**CREATE** **TABLE** settings (

id INT(11) **NOT** **NULL** AUTO\_INCREMENT,

user\_id INT(11) **NOT** **NULL**,

theme VARCHAR(255) **DEFAULT** **NULL**,

default\_view VARCHAR(255) **DEFAULT** **NULL**,

last\_password\_change **TIMESTAMP** **NULL**,

**PRIMARY** **KEY** (id),

**KEY** (user\_id)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8mb4;

**CREATE** **TABLE** tasks (

id INT(11) **NOT** **NULL** AUTO\_INCREMENT,

project\_id INT(11) **NOT** **NULL**,

team\_id INT(11) **NOT** **NULL**,

title VARCHAR(100) **NOT** **NULL**,

description TEXT **DEFAULT** **NULL**,

status VARCHAR(20) **DEFAULT** 'TO\_DO',

priority VARCHAR(20) **DEFAULT** 'MEDIUM',

start\_date DATE **DEFAULT** **NULL**,

end\_date DATE **DEFAULT** **NULL**,

**PRIMARY** **KEY** (id),

**KEY** (project\_id),

**KEY** (team\_id)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8mb4;

**CREATE** **TABLE** task\_activities (

id INT(11) **NOT** **NULL** AUTO\_INCREMENT,

task\_id INT(11) **NOT** **NULL**,

user\_id INT(11) **NOT** **NULL**,

activity\_type VARCHAR(50) **NOT** **NULL**,

description TEXT **DEFAULT** **NULL**,

created\_at **TIMESTAMP** **NOT** **NULL** **DEFAULT** **current\_timestamp**(),

**PRIMARY** **KEY** (id),

**KEY** (task\_id),

**KEY** (user\_id)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8mb4;

**CREATE** **TABLE** task\_assignments (

task\_id INT(11) **NOT** **NULL**,

user\_id INT(11) **NOT** **NULL**,

**PRIMARY** **KEY** (task\_id, user\_id),

**KEY** (user\_id)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8mb4;

**CREATE** **TABLE** teams (

id INT(11) **NOT** **NULL** AUTO\_INCREMENT,

team\_name VARCHAR(100) **NOT** **NULL**,

project\_id INT(11) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (id),

**KEY** (project\_id)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8mb4;

**CREATE** **TABLE** team\_members (

team\_id INT(11) **NOT** **NULL**,

user\_id INT(11) **NOT** **NULL**,

is\_leader TINYINT(1) **DEFAULT** 0,

**PRIMARY** **KEY** (team\_id, user\_id),

**KEY** (user\_id)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8mb4;

**CREATE** **TABLE** groups (

id INT(11) **NOT** **NULL** AUTO\_INCREMENT,

group\_name VARCHAR(50) **NOT** **NULL**,

**PRIMARY** **KEY** (id)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8mb4;

**CREATE** **TABLE** users (

id INT(11) **NOT** **NULL** AUTO\_INCREMENT,

name VARCHAR(50) **NOT** **NULL**,

last\_name VARCHAR(50) **NOT** **NULL**,

email VARCHAR(100) **NOT** **NULL**,

password VARCHAR(255) **NOT** **NULL**,

password\_hint VARCHAR(255) **DEFAULT** **NULL**,

role\_id INT(11) **DEFAULT** **NULL**,

group\_id INT(11) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (id),

**UNIQUE** **KEY** (email),

**KEY** (role\_id)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8mb4;

**ALTER** **TABLE** notifications **ADD** **FOREIGN** **KEY** (user\_id) **REFERENCES** users (id);

**ALTER** **TABLE** projects **ADD** **FOREIGN** **KEY** (manager\_id) **REFERENCES** users (id);

**ALTER** **TABLE** reports **ADD** **FOREIGN** **KEY** (created\_by) **REFERENCES** users (id);

**ALTER** **TABLE** settings **ADD** **FOREIGN** **KEY** (user\_id) **REFERENCES** users (id);

**ALTER** **TABLE** tasks

**ADD** **FOREIGN** **KEY** (project\_id) **REFERENCES** projects (id),

**ADD** **FOREIGN** **KEY** (team\_id) **REFERENCES** teams (id);

**ALTER** **TABLE** task\_activities

**ADD** **FOREIGN** **KEY** (task\_id) **REFERENCES** tasks (id),

**ADD** **FOREIGN** **KEY** (user\_id) **REFERENCES** users (id);

**ALTER** **TABLE** task\_assignments

**ADD** **FOREIGN** **KEY** (task\_id) **REFERENCES** tasks (id),

**ADD** **FOREIGN** **KEY** (user\_id) **REFERENCES** users (id);

**ALTER** **TABLE** teams **ADD** **FOREIGN** **KEY** (project\_id) **REFERENCES** projects (id);

**ALTER** **TABLE** team\_members

**ADD** **FOREIGN** **KEY** (team\_id) **REFERENCES** teams (id),

**ADD** **FOREIGN** **KEY** (user\_id) **REFERENCES** users (id);

**ALTER** **TABLE** users **ADD** **FOREIGN** **KEY** (role\_id) **REFERENCES** roles (id);

# Wykorzystane technologie

**Język Java 21**

* **Opis:** Java to język programowania ogólnego przeznaczenia, ceniony za przenośność, wydajność i bogatą bibliotekę standardową. Wersja 21 wprowadza nowe API, ulepszenia w zakresie bezpieczeństwa oraz optymalizacje kodu. W projekcie posłuży do realizacji logiki biznesowej, tworzenia interfejsu użytkownika (z wykorzystaniem JavaFX) oraz integracji z bazą danych.

**Baza danych MariaDB**

* **Opis:** MySQL to popularny relacyjny system zarządzania bazami danych (RDBMS), który charakteryzuje się wysoką wydajnością, skalowalnością i kompatybilnością z różnymi platformami. W systemie będzie wykorzystywany do przechowywania danych użytkowników, zadań, projektów, raportów oraz konfiguracji. Dzięki wsparciu dla transakcji, indeksowania i replikacji zapewni integralność i dostępność danych.

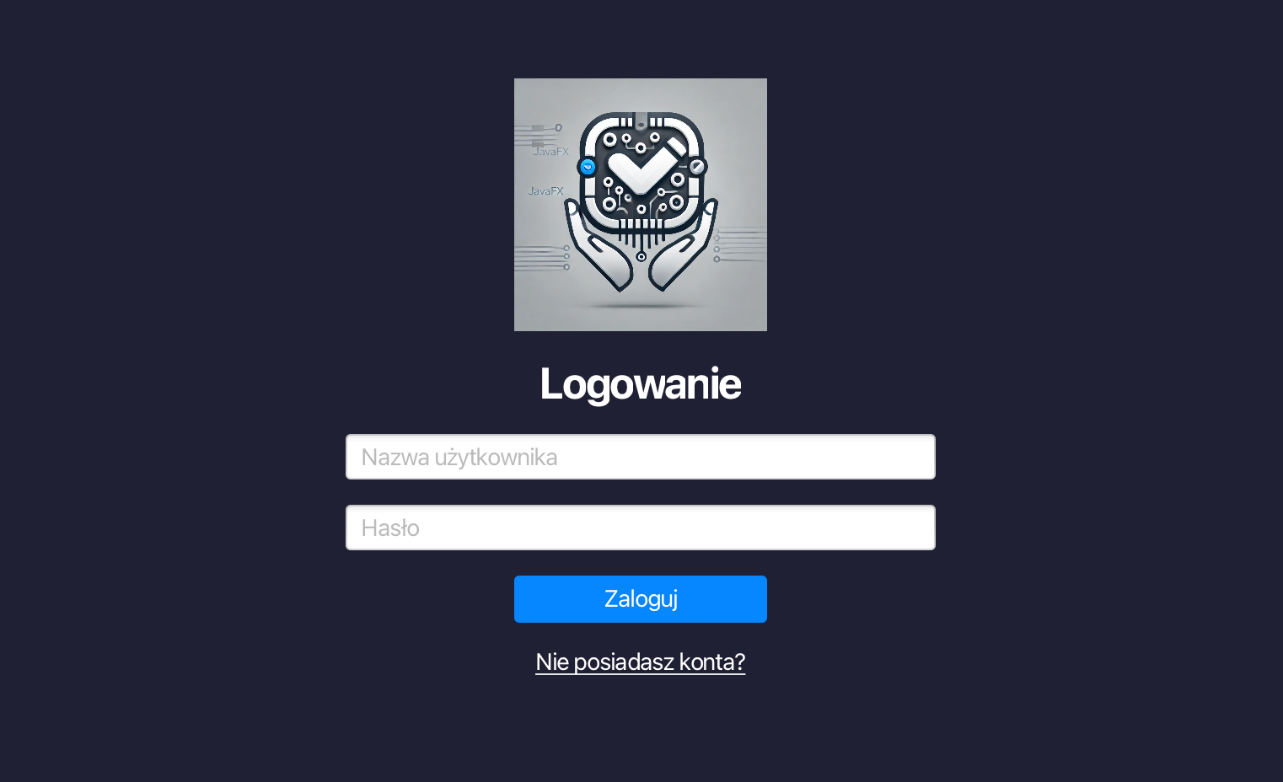
**Inne technologie**

* **JavaFX**
  + **Opis:** JavaFX to biblioteka służąca do tworzenia interfejsów użytkownika w aplikacjach desktopowych. Oferuje nowoczesne narzędzia do projektowania atrakcyjnych i responsywnych interfejsów, co przełoży się na intuicyjną obsługę systemu przez użytkowników.
* **iText**
  + **Opis:** iText to biblioteka open-source w Javie, przeznaczona do generowania i edycji dokumentów PDF. W projekcie zostanie wykorzystana do tworzenia profesjonalnie sformatowanych raportów PDF, które użytkownicy będą mogli generować i eksportować.
* **JUnit**
  + **Opis:** JUnit to framework do testowania jednostkowego w Javie, który pozwala na automatyzację testów, zapewniając poprawność i niezawodność kodu. W systemie posłuży do weryfikacji działania poszczególnych modułów i funkcji.
* **Git**
  + **Opis:** Git to system kontroli wersji, który umożliwia zespołowi programistów współpracę nad kodem, śledzenie zmian i zarządzanie wersjami projektu. Ułatwi organizację pracy zespołowej i rozwój oprogramowania.

# Interfejs aplikacji / systemu

**Ekran logowania**

*Ekran, na którym użytkownik podaje login i hasło w celu uzyskania dostępu do systemu.*



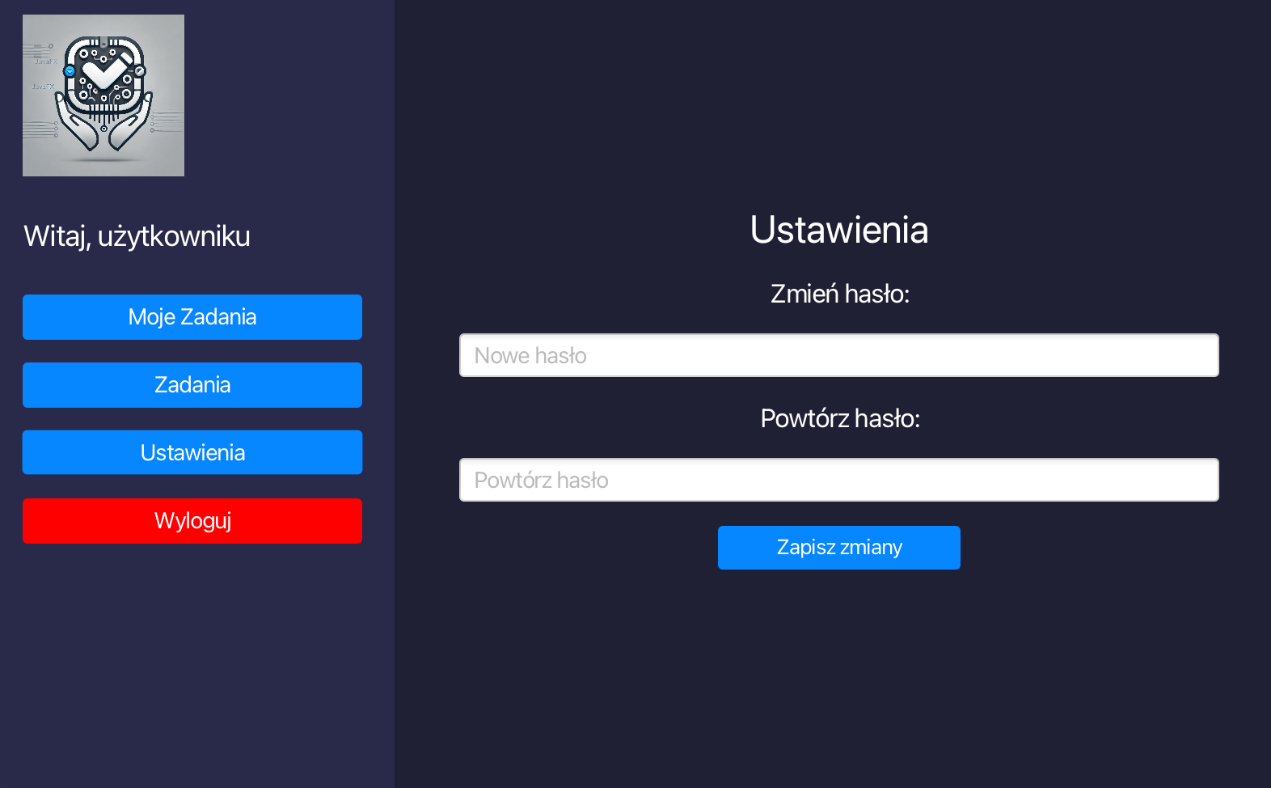
**Ekran rejestracji**

*Formularz do tworzenia nowego konta w systemie.*



**Panel użytkownika**

*Panel z bocznym menu, gdzie pracownik widzi swoje zadania, może je aktualizować i zgłaszać postępy.*



**Menu główne / Dashboard**

*Przykładowy widok głównego menu, dostępnego po zalogowaniu w zależności od roli.*

