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

**Przedmiot:**

**Programowanie zespołowe**

**Dokumentacja projektu:**

***System do zarządzania zadaniami  
w*** ***sklepie materiałowo-ogrodowym „ObiWanKenobi”***

**Wykonał:**

**Zespół projektowy Zieloni**

**Prowadzący: mgr inż. Adam Szczur**

**Rzeszów 2025**

Spis treści

[1. Zespół projektowy 3](#_Toc193187472)

[2. Specyfikacja projektu 3](#_Toc193187473)

[2.1. Cel projektu 3](#_Toc193187474)

[2.2. Zakres projektu 3](#_Toc193187475)

[2.2.1. Zasoby 3](#_Toc193187476)

[2.2.2. Produkty końcowe projektu 4](#_Toc193187477)

[2.2.3. Harmonogram realizacji projektu 4](#_Toc193187478)

[2.3. Wymagania stawiane aplikacji / systemowi 4](#_Toc193187479)

[2.4. Panele / zakładki systemu, które będą oferowały potrzebne funkcjonalności 5](#_Toc193187480)

[2.5.1. Raporty PDF 5](#_Toc193187481)

[2.5.2. Inne dokumenty 5](#_Toc193187482)

[2.6. Przepływ informacji w środowisku systemu 5](#_Toc193187483)

[2.7. Użytkownicy aplikacji i ich uprawnienia 6](#_Toc193187484)

[2.8. Interesariusze 6](#_Toc193187485)

[3. Diagramy UML 6](#_Toc193187486)

[3.1. Diagram przypadków użycia 6](#_Toc193187487)

[3.2. Diagramy aktywności 7](#_Toc193187488)

[3.3. Diagramy sekwencji 11](#_Toc193187489)

[4. Baza danych 15](#_Toc193187490)

[4.1. Diagram ERD 15](#_Toc193187491)

[4.2. Skrypt do utworzenia struktury bazy danych 15](#_Toc193187492)

[5. Wykorzystane technologie 15](#_Toc193187493)

[6. Interfejs aplikacji / systemu 16](#_Toc193187494)

# Zespół projektowy

Nazwa zespołu: Zieloni

Skład zespołu projektowego:

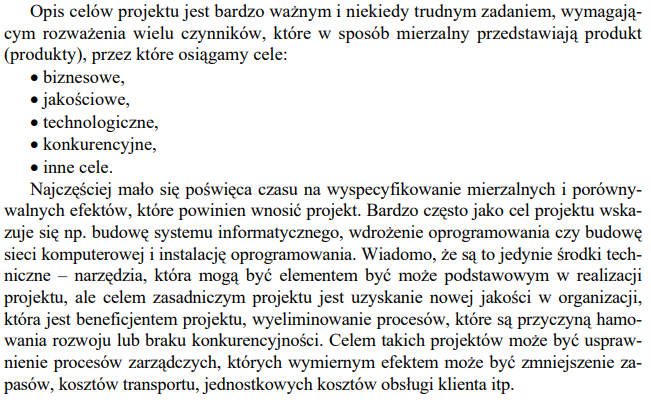
|  |  |  |
| --- | --- | --- |
| Imię i nazwisko | Pełniona rola | Zakres czynności / odpowiedzialności |
| Miłosz Gierus | Project Menager | Projekt i implementacja DB, Logika aplikacji |
| Bartłomiej Florek | Ux/Ui designer, Backend Developer | JavaFx, Logika aplikacji |
| Kacper Długosz | Backend Developer, Ux/Ui designer | Tworzenie zapytań, optymalizacja, Logika aplikacji |
| Ewelina Kaniewska | Tester / analityk biznesowy | JavaFx, Logika Biznesowa, Logika aplikacji |
| Arkadiusz Haznar | Backend Developer / Ux/Ui designer | JavaFx, Integracja z bazą danych, Logika aplikacji |

# Specyfikacja projektu

## Cel projektu

Opis słowny

System "ObiWanKeboi" ma na celu usprawnienie zarządzania zadaniami poprzez eliminację chaosu organizacyjnego, poprawę komunikacji między pracownikami oraz zapewnienie bieżącej kontroli nad postępem realizacji prac. Kluczowe funkcje obejmują centralizację informacji o zadaniach, szybkie przekazywanie poleceń oraz zapewnienie przejrzystości w procesach decyzyjnych.



## Zakres projektu

### Zasoby

Określenie zasobów ludzkich, sprzętowych, programowych, finansowych i innych, np.:

* *Zespół programistów (3 osoby), 10 godzin pracy tygodniowo przez 10 tygodni;*
* *Zespół testerów (2 osoby), 5 godzin pracy tygodniowo przez 2 tygodnie;*
* *UI Designer, 5 godzin pracy;*
* *Darmowe narzędzia do wytworzenia systemu: IntelliJ, MySQL, GitHub, Jira, inne*

### Produkty końcowe projektu

Określenie zakładanych produktów, które powstaną w wyniku realizacji projektu, np.:

* *System do zarządzania zadaniami w ObiWanKenobi*
  + *Pakiet instalacyjny*
  + *Kod źródłowy aplikacji*
  + *Skrypty do utworzenia bazy danych*
  + *Moduł raportowania i analizy danych*
  + *Moduł zarządzania użytkownikami i uprawnieniami*
  + *Testy jednostkowe i integracyjne*
* *Dokumentacja techniczna projektu*
  + *Opis architektury systemu*
  + *Instrukcja wdrożenia i konfiguracji*
  + *Dokumentacja bazy danych*
* *Podręcznik użytkownika dla wytworzonego systemu*
  + *Instrukcja obsługi dla pracowników*
  + *Przewodnik dla kierowników*
  + *Instrukcja administracyjna dla administratorów*
* *Szkolenie dla użytkowników systemu*
  + *Materiały szkoleniowe (prezentacje, instrukcje, ćwiczenia)*
  + *Warsztaty praktyczne dla użytkowników końcowych*
  + *Szkolenie dla administratorów systemu*
* *Dodatkowe elementy wynikające z pracy zespołu projektowego*
  + *Plan zarządzania projektem*
  + *Raporty z testów i ewaluacji systemu*
  + *Repozytorium kodu źródłowego z systemem kontroli wersji*
  + *Dokumentacja procesu wytwarzania oprogramowania*

### Harmonogram realizacji projektu

Wymienienie wszystkich zadań dot. projektu wraz z określeniem dat ich realizacji, np.:

|  |  |
| --- | --- |
| Zakres dat | Zadanie |
| 04.03.2024 – 06.03.2024 | Opracowanie specyfikacji projektu |
| 11.03.2024 – 15.03.2024 | Zaprojektowanie GUI aplikacji |
| … | … |

#### Wykres Gantta

Wykres Gantta powinien zawierać wszystkie zadania z powyższej tabelki. Na wykresie powinny znaleźć się: nazwy zadań, daty ich rozpoczęcia i zakończenia oraz zależności między zadaniami. Dodatkowo można też określić: kamienie milowe, osoby przypisane, postęp realizacji zadań, itp.

Należy wkleić tutaj rysunek diagramu Gantta. Ponadto plik graficzny z diagramem Gantta należy zamieścić w repozytorium.

## Wymagania stawiane aplikacji / systemowi

Aplikacja powinna mieć kilka modułów:

1. Administracja użytkownikami (role),
2. Zarządzanie zadaniami,
3. Moduł raportów,
4. Moduł logowania.
5. Gromadzenie danych.
6. Moduł zgłaszania błędów.

System powinien współpracować z bazą danych.

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

* Panel administratora
  + Główne narzędzie administratora systemu umożliwiające wykonanie wszystkich czynności potrzebnych do zarządzania systemem np. dodawanie, edycja, usuwanie, tworzenie i modyfikacja kont.
* Panel dyrektora
  + Główne narzędzie dyrektora umożliwiające generowania raportów oraz pogląd działań personelu.
* Panel kierownika
  + Główne narzędzie kierownika umożliwiające tworzenie, usuwanie, edycję zadań, oraz zatwierdzenie zakończonych zadań.
* Panel pracowników
  + Podgląd przypisanych zadań, zgłaszanie problemów, oznaczanie statusów zadań oraz wylogowanie.
* Panel gościa
  + Dostęp do panelu logowania.
  1. Typy wymaganych dokumentów w projekcie oraz dostęp do nich

### Raporty PDF

#### Rodzaje raportów PDF

* Raport wszystkich zadań - generowany dla dyrektora i kierowników.
* Raport zadań zakończonych - dostarcza informacji o zamkniętych zadaniach.
* Raport ilości zadań w trakcie wykonywania - pomaga w bieżącym monitorowaniu realizacji prac.
* Raport aktualnie zatrudnionych pracowników.

### Inne dokumenty

## Przepływ informacji w środowisku systemu

System jest scentralizowany i oparty na bazie danych, co pozwala na efektywne zarządzanie danymi oraz ich bezpieczne przechowywanie.

1. Logowanie i zarządzanie użytkownikami

* Gość uzyskuje dostęp do panelu logowania.
* Wprowadzone dane są weryfikowane przez moduł logowania.
* Po poprawnej weryfikacji system przekierowuje użytkownika do odpowiedniego panelu (Administratora, Dyrektora, Kierownika, Pracownika).
* Administrator może zarządzać użytkownikami (dodawanie, edycja, usuwanie kont, przypisywanie ról).

1. Zarządzanie zadaniami

* Kierownik tworzy zadanie i przypisuje je pracownikowi.
* Zadanie jest zapisywane w bazie danych i przypisane do konkretnego użytkownika.
* Pracownik może aktualizować status zadania oraz zgłaszać problemy.
* Kierownik monitoruje postęp i zatwierdza zakończone zadania.

1. Generowanie raportów

* Dyrektor lub kierownik wybiera typ raportu.
* System pobiera dane z bazy i generuje raport w formacie PDF.
* Użytkownik może pobrać interesujący go raport.

1. Zgłaszanie błędów

* Pracownik lub kierownik zgłasza problem w systemie.
* System zapisuje zgłoszenie w bazie danych.
* Administrator może analizować błędy i podejmować działania naprawcze.

1. Komunikacja między GUI, modułami i bazą danych

* GUI (Interfejs Użytkownika) - dostarcza formularze i panele do interakcji użytkownika z systemem.
* Moduły aplikacji - przetwarzają dane oraz komunikują się z bazą danych.
* Baza danych - przechowuje informacje o użytkownikach, zadaniach, raportach oraz zgłoszeniach.

## Użytkownicy aplikacji i ich uprawnienia

* Administrator
  + pełny dostęp do wszystkich funkcji systemu
  + zarządzanie użytkownikami
* Dyrektor
  + wgląd do wszystkich raportów
  + generowanie raportów
* Kierownik
  + tworzenie, edytowanie i przypisywanie zadań w obrębie swojego działu
  + monitorowanie postępu oraz zatwierdzanie zakończonych prac
* Pracownik
  + dostęp do przypisanych zadań
  + modyfikacja statusu swoich zadań
  + zgłaszanie problemów
* Gość
  + logowanie do systemu

## Interesariusze

* Interesariusze wewnętrzni
  + Dyrektor
  + Kierownik
  + Pracownik
* Interesariusze zewnętrzni
  + Administrator
  + Gość

# Diagramy UML

## Diagram przypadków użycia

Opis:

W projektowanej aplikacji wyróżnia się 5 aktorów którzy w zależności od uprawnie posiadają pewnie funkcjonalności, poniżej przedstawiono diagram przypadków użycia z uwzględnieniem przyporządkowania ich do określonych aktorów tj. Gość, Pracownik, Kierownik, Dyrektor, Administrator:

* Gość – jest to niezalogowany użytkownik systemu, który ma ograniczoną funkcjonalność do zalogowania się, co da mu dostęp do pulpitu nawigacyjnego.
* Pracownik – Zalogowany jako pracownik ma dostęp do przeglądania przypisanych zadań, oznaczania ich statusu oraz zgłaszania problemów. Ma również możliwość wylogowania się.
* Kierownik – Zalogowany jako kierownik ma możliwość dodawania nowych zadań, usuwania zadań, zatwierdzenia wykonanych zadań i modyfikowania zadań w co wchodzi zmiana opisu, zmiana nazwy, zmiana wykonawcy zadania. Ma również możliwość wylogowania się.
* Dyrektor – Zalogowany jako dyrektor ma możliwość podglądu działań działów oraz generowania raportów. Ma również możliwość wylogowania się.
* Administrator – Zalogowany jako administrator ma pełny dostęp do wszystkich funkcji systemu, zarządzanie użytkownikami, czyli dodawanie/usuwanie kont, zmiana hasła, nazwy oraz uprawnień istniejącego konta. Ma również możliwość wylogowania się.

Obraz zawierający tekst, diagram, szkic, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 1 - Diagram przypadków użycia

## Diagramy aktywności

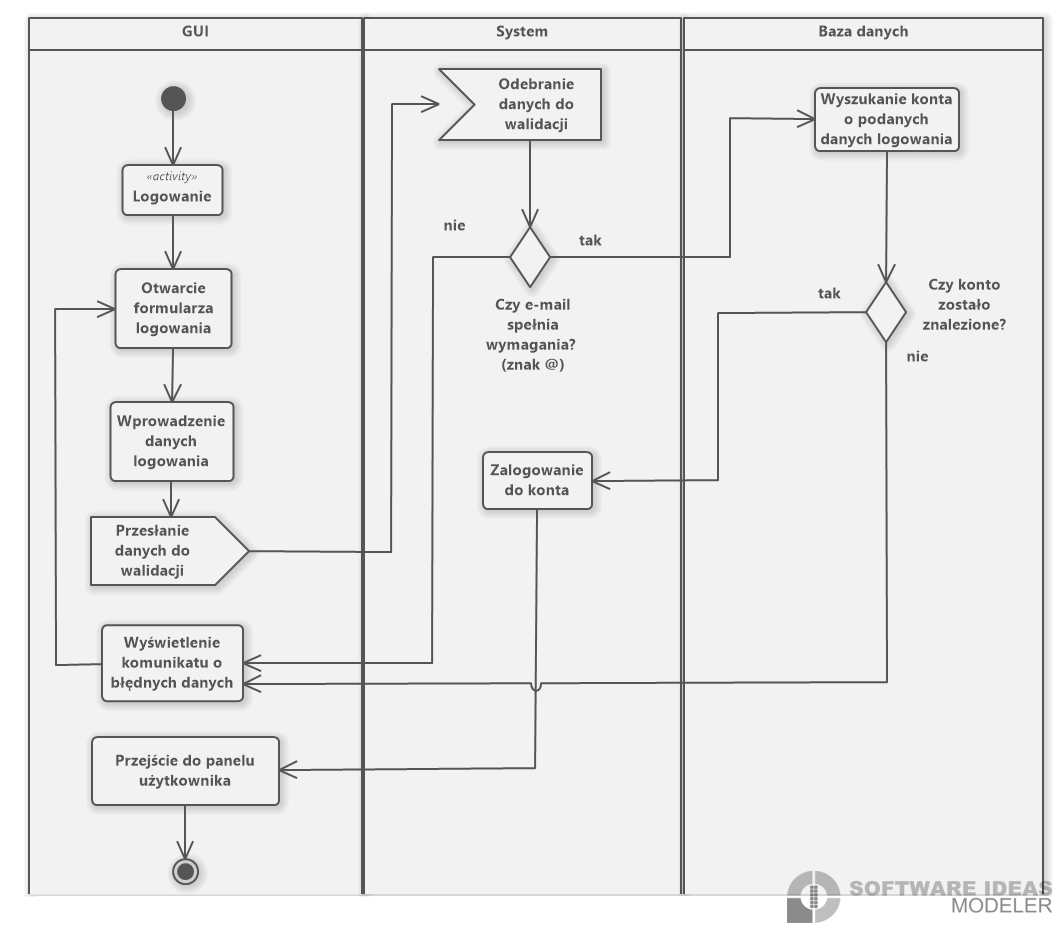
Przypadek użycia 1: Logowanie

* Aktor: Gość
* Cel: Gość chce uzyskać dostęp do swojego konta.
* Warunki początkowe: Gość jest zarejestrowany i zna swoje dane logowania.
* Przebieg główny:

1. Gość otwiera panel logowania.
2. Gość wprowadza adres e-mail i hasło.
3. System weryfikuje poprawność danych logowania (w tym format e-maila).
4. System sprawdza, czy konto istnieje w bazie danych.
5. Jeśli dane są poprawne i konto istnieje, system loguje użytkownika.
6. Gość zostaje przekierowany do panelu użytkownika.

* Scenariusz alternatywny (dla kroku 3 i 4):
* Jeśli e-mail nie spełnia wymagań, system wyświetla komunikat o błędzie.
* Jeśli konto o podanych danych logowania nie istnieje, system wyświetla komunikat o błędnych danych.
* Warunki końcowe: Gość jest zalogowany i ma dostęp do systemu lub widzi komunikat o błędnych danych logowania.

W oparciu o zdefiniowany scenariusz na Rysunku 2 przedstawiono diagram aktywności dla przypadku użycia 1:



Rysunek 2 Diagram aktywności dla przypadku użycia "Logowanie"

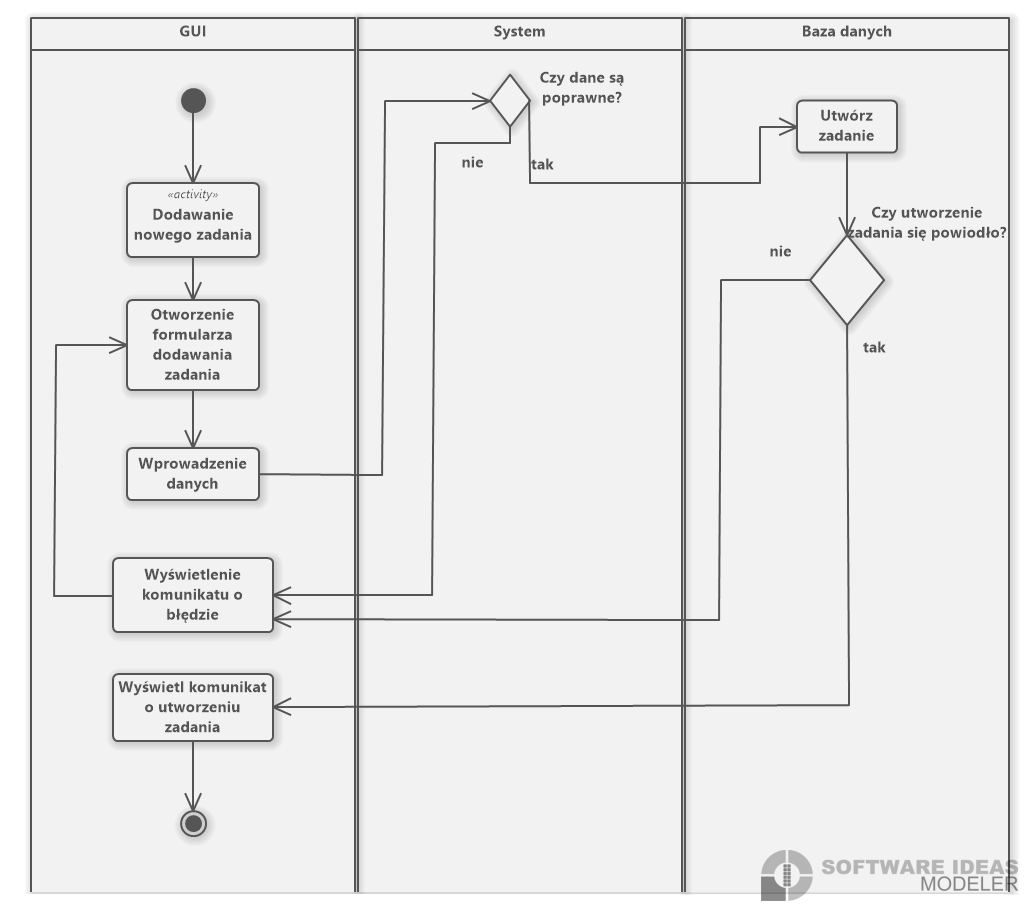
Przypadek użycia 2: Dodawanie nowego zadania

* Aktor: Kierownik
* Cel: Kierownik chce dodać nowe zadanie do systemu.
* Warunki początkowe: Kierownik jest zalogowany i posiada uprawnienia do dodawania zadań.
* Przebieg główny:

1. Kierownik otwiera formularz dodawania zadania.
2. Kierownik wprowadza dane zadania.
3. System sprawdza poprawność wprowadzonych danych.
4. System przesyła dane do bazy danych w celu utworzenia zadania.
5. System wyświetla komunikat o pomyślnym utworzeniu zadania.

* Scenariusz alternatywny (dla kroku 3 i 4):
* Jeśli dane są niepoprawne, system wyświetla komunikat o błędzie.
* Jeśli utworzenie zadania w bazie danych się nie powiodło, system informuje o błędzie.
* Warunki końcowe: Zadanie zostaje dodane do bazy danych lub kierownik widzi komunikat o błędzie.

W oparciu o zdefiniowany scenariusz na Rysunku 3 przedstawiono diagram aktywności dla przypadku użycia 2:



Rysunek 3 Diagram aktywności dla przypadku użycia "Dodawanie nowego zadania"

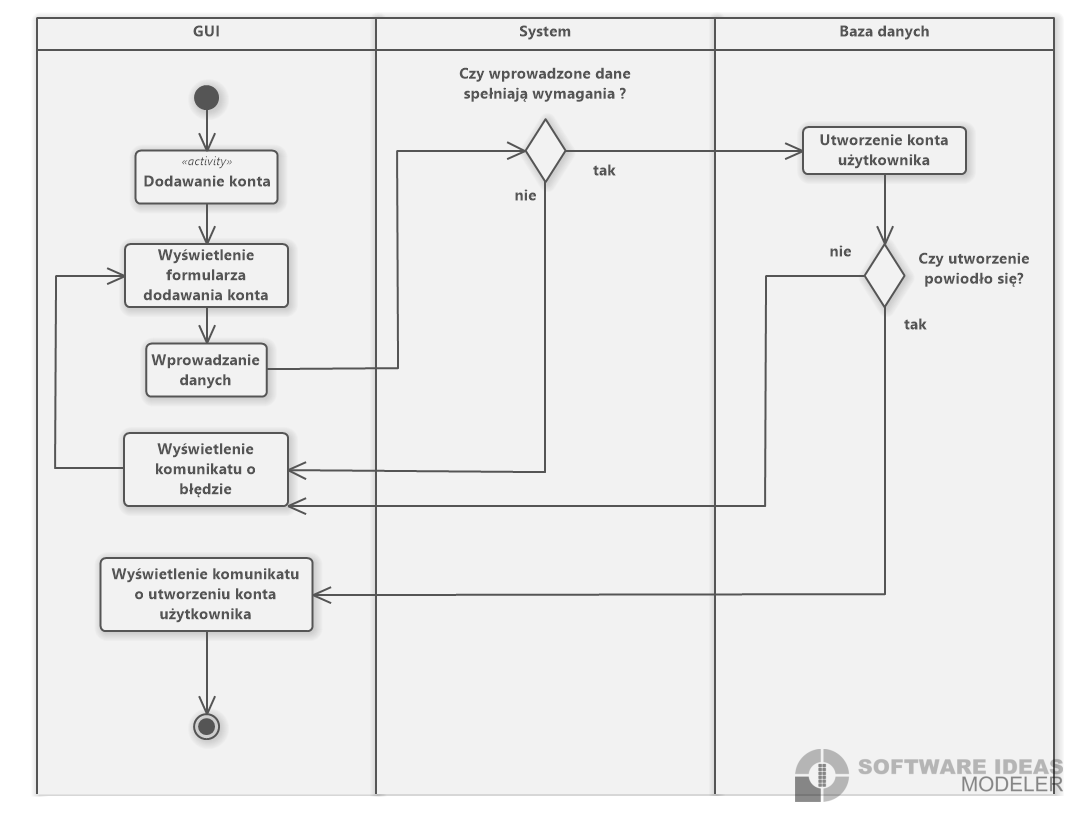
Przypadek użycia 3: Dodawanie nowego konta

* Aktor: Administrator
* Cel: Administrator chce dodać nowe konto użytkownika.
* Warunki początkowe: Administrator jest zalogowany i posiada odpowiednie uprawnienia.
* Przebieg główny:

1. Administrator otwiera formularz dodawania konta.
2. Administrator wprowadza dane nowego użytkownika.
3. System sprawdza, czy dane spełniają wymagania.
4. System przesyła dane do bazy danych w celu utworzenia konta.
5. System wyświetla komunikat o pomyślnym utworzeniu konta.

* Scenariusz alternatywny (dla kroku 3 i 4):
* Jeśli dane nie spełniają wymagań, system wyświetla komunikat o błędzie.
* Jeśli utworzenie konta w bazie danych się nie powiodło, system informuje o błędzie.
* Warunki końcowe: Konto zostaje utworzone w systemie lub administrator widzi komunikat o błędzie.

W oparciu o zdefiniowany scenariusz na Rysunku 4 przedstawiono diagram aktywności dla przypadku użycia 3:



Rysunek 4 Diagram aktywności dla przypadku użycia "Dodawanie nowego konta"

## Diagramy sekwencji

**Proces logowania użytkownika (diagram sekwencji)**

Diagram sekwencji przedstawia proces logowania użytkownika do systemu z uwzględnieniem ścieżek wykonania w przypadku poprawnych i błędnych danych logowania.

Główne elementy diagramu:

1. **Gość** – użytkownik próbujący uzyskać dostęp do systemu.
2. **Interfejs** **użytkownika** – umożliwia wprowadzanie danych logowania oraz przekazuje informacje zwrotne.
3. **System** – odpowiada za walidację oraz autoryzację użytkownika.
4. **Baza danych** – przechowuje dane logowania i realizuje ich weryfikację.

Przebieg interakcji:

1. Wprowadzenie danych logowania

• Gość podaje dane logowania za pomocą interfejsu użytkownika.

• Interfejs przekazuje dane do systemu w celu ich walidacji.

1. Walidacja danych

• System sprawdza poprawność wprowadzonych informacji.

• Jeśli dane są błędne, użytkownik otrzymuje komunikat o nieprawidłowych danych.

1. Weryfikacja w bazie danych (Dwa możliwe scenariusze)

a) Sukces:

• System przesyła zweryfikowane dane do bazy danych.

• Baza potwierdza poprawność informacji.

• Użytkownik otrzymuje komunikat o pomyślnym zalogowaniu.

b) Niepowodzenie:

• Jeśli dane logowania nie są zgodne z zapisami w bazie, zwracany jest błąd.

• Użytkownik otrzymuje informację o błędzie logowania.

Obraz zawierający tekst, zrzut ekranu

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 5 Diagram sekwencji dla przypadku użycia "Logowanie"

**Dodawanie nowego zadania (diagram sekwencji)**

Diagram sekwencji przedstawia proces dodawania nowego zadania do systemu przez kierownika. Pokazuje interakcje pomiędzy użytkownikiem, interfejsem użytkownika, systemem oraz bazą danych.

Główne elementy diagramu:

1. **Kierownik** – użytkownik próbujący dodać nowe zadanie
2. **Interfejs użytkownika** – moduł umożliwiający wprowadzenie informacji o nowym zadaniu
3. **System** – obsługuje proces walidacji nowego zadania
4. **Baza danych** – przechowuje dane na temat zadań

Przebieg interakcji:

1. Kierownik wprowadza informacje na temat nowego zadania w panelu dodawania zadań
2. Interfejs przekazuje dane do systemu, który weryfikuje poprawność wprowadzonych danych
   1. Blok alternatywny w przypadku niepowodzenia weryfikacji danych, zwraca negatywny wynik i kierownik otrzymuje komunikat o błędnych danych
   2. Jeśli dane są poprawne, przechodzi do następnego etapu
3. System próbuje utworzyć nowe zadanie w bazie danych
   1. Blok alternatywny w przypadku jeśli zadanie nie zostanie utworzone poprawnie, system zwraca informacje o niepowodzeniu i kierownik otrzymuje komunikat
   2. Jeśli utworzenie zadania powiedzie się, system zwraca wynik pozytywny a kierownik otrzymuje odpowiedni komunikat

Obraz zawierający tekst, zrzut ekranu

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 6 Diagram sekwencji dla przypadku użycia "Dodawanie nowego zadania"

**Dodanie nowego konta (diagram sekwencji)**

Diagram sekwencji przedstawia proces dodawania nowego użytkownika do systemu przez administratora. Pokazuje interakcje pomiędzy administratorem, interfejsem użytkownika, systemem oraz bazą danych.

Główne elementy diagramu:

1. **Administrator** – inicjator procesu tworzenia konta.
2. **Interfejs użytkownika** – pośredniczy w komunikacji między administratorem a systemem.
3. **System** – odpowiada za walidację danych oraz obsługę procesu tworzenia konta.
4. **Baza danych** – przechowuje dane konta i zwraca odpowiedź o powodzeniu lub niepowodzeniu operacji.

Przebieg interakcji:

1. Rozpoczęcie procesu
   * Administrator wprowadza dane konta poprzez interfejs użytkownika.
   * Interfejs przesyła dane do systemu w celu walidacji.
2. Walidacja danych
   * System sprawdza poprawność wprowadzonych informacji.
   * Jeśli dane są niepoprawne, zwracany jest błąd, a interfejs użytkownika informuje administratora o konieczności poprawienia danych.
3. Tworzenie konta (Dwa możliwe scenariusze)
   1. Sukces:
      * Po poprawnej walidacji system wysyła żądanie utworzenia konta do bazy danych.
      * Baza zwraca pozytywną odpowiedź.
      * System informuje administratora o pomyślnym utworzeniu konta.
   2. Niepowodzenie:
      * Jeśli proces tworzenia konta w bazie się nie powiedzie, system zwraca informację o błędzie.
      * Administrator zostaje poinformowany o problemie i może ponowić próbę

Obraz zawierający zrzut ekranu, tekst

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek 7 Diagram sekwencji dla przypadku użycia "Dodanie nowego konta"

# Baza danych

## Diagram ERD

Wstawić diagram ERD bazy danych systemu. Dodatkowo zamieścić rysunek z diagramem w repozytorium.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

## Skrypt do utworzenia struktury bazy danych

Wkleić treść skryptu SQL tworzącego bazę danych systemu i jej strukturę (bez danych). Dodatkowo w repozytorium należy zamieścić 2 skrypty SQL: skrypt tworzący bazę danych i jej strukturę (bez danych) oraz skrypt tworzący bazę danych ze strukturą i przykładowymi danymi.

CREATE DATABASE Obiwanshop;

USE Obiwanshop;

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

email VARCHAR(255) NOT NULL UNIQUE,

password VARCHAR(255) NOT NULL,

first\_name VARCHAR(255) NOT NULL,

last\_name VARCHAR(255) NOT NULL,

department\_id INT,

role\_id INT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE departments (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL UNIQUE,

manager\_id INT,

FOREIGN KEY (manager\_id) REFERENCES users(id) ON DELETE SET NULL

);

CREATE TABLE roles (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL UNIQUE

);

CREATE TABLE tasks (

id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(255) NOT NULL,

description TEXT,

user\_id INT,

status VARCHAR(255),

deadline TIMESTAMP NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE

);

CREATE TABLE warehouses (

id INT AUTO\_INCREMENT PRIMARY KEY,

department\_id INT UNIQUE,

name VARCHAR(255) NOT NULL,

FOREIGN KEY (department\_id) REFERENCES departments(id) ON DELETE CASCADE

);

CREATE TABLE items (

id INT AUTO\_INCREMENT PRIMARY KEY,

warehouse\_id INT,

name VARCHAR(255) NOT NULL,

quantity INT NOT NULL,

description TEXT,

FOREIGN KEY (warehouse\_id) REFERENCES warehouses(id) ON DELETE CASCADE

);

ALTER TABLE users ADD CONSTRAINT fk\_department FOREIGN KEY (department\_id) REFERENCES departments(id) ON DELETE SET NULL;

ALTER TABLE users ADD CONSTRAINT fk\_role FOREIGN KEY (role\_id) REFERENCES roles(id) ON DELETE SET NULL;

# Wykorzystane technologie

* Java Electron + SpringBoot – IntelliJ (ew. JavaFX)
* MySQL (MariaDB)
* GitHub
* Jira

# Interfejs aplikacji / systemu

Zamieścić screeny GUI systemu wraz z krótkim opisem. Szczegółowy opis będzie w podręczniku użytkownika.

Panel startowy po uruchomieniu aplikacji

Obraz zawierający tekst, zrzut ekranu, projekt graficzny, design

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Panel dodawanie nowego użytkownika

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

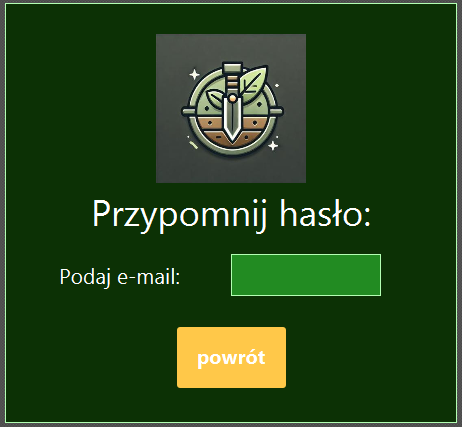
Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Panel logowania

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Panel w przypadku gdy użytkownik zapomni swojego hasła



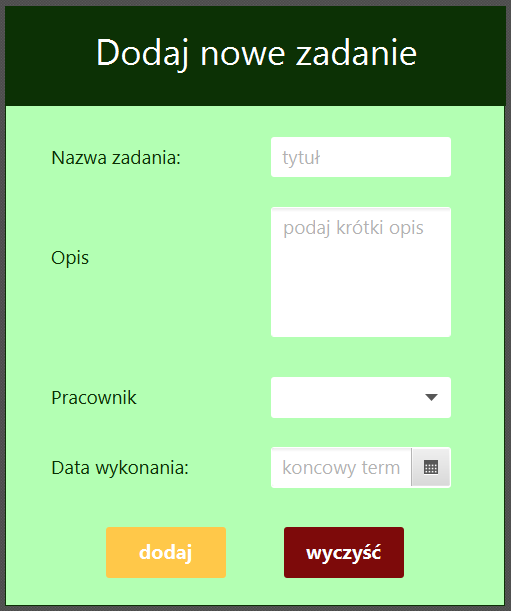
Panel generowania raportów



Panel główny aplikacji



Panel dodawania nowego zadania dla pracowników



Sczegóły zadania

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Zarządzanie zadaniami



Panel dyrektora

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Panel addministratora

Obraz zawierający tekst, elektronika, zrzut ekranu, wyświetlacz

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.