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

**Przedmiot:**

**Programowanie zespołowe**

**Dokumentacja projektu:**

***System do zarządzania zadaniami w sklepie ogrodniczo-budowlanym „ObiWanKenobi”***

**Wykonał:**

**Zespół projektowy Zieloni**

**Prowadzący: mgr inż. Adam Szczur**

**Rzeszów 2025**

Spis treści

[1. Zespół projektowy 3](#_Toc198151100)

[2. Specyfikacja projektu 3](#_Toc198151101)

[2.1. Cel projektu 3](#_Toc198151102)

[2.2. Zakres projektu 4](#_Toc198151103)

[2.2.1. Zasoby 4](#_Toc198151104)

[2.2.2. Produkty końcowe projektu 4](#_Toc198151105)

[2.2.3. Harmonogram realizacji projektu 5](#_Toc198151106)

[2.3. Wymagania stawiane aplikacji / systemowi 5](#_Toc198151107)

[2.4. Panele / zakładki systemu, które będą oferowały potrzebne funkcjonalności 6](#_Toc198151108)

[2.5.1. Raporty PDF 6](#_Toc198151109)

[2.5.2. Inne dokumenty 6](#_Toc198151110)

[2.6. Przepływ informacji w środowisku systemu 6](#_Toc198151111)

[2.7. Użytkownicy aplikacji i ich uprawnienia 7](#_Toc198151112)

[2.8. Interesariusze 7](#_Toc198151113)

[3. Diagramy UML 7](#_Toc198151114)

[3.1. Diagram przypadków użycia 7](#_Toc198151115)

[3.2. Diagramy aktywności 8](#_Toc198151116)

[3.3. Diagramy sekwencji 12](#_Toc198151117)

[4. Baza danych 17](#_Toc198151118)

[4.1. Diagram ERD 17](#_Toc198151119)

[4.2. Skrypt do utworzenia struktury bazy danych 17](#_Toc198151120)

[5. Wykorzystane technologie 18](#_Toc198151121)

[6. Interfejs aplikacji / systemu 18](#_Toc198151122)

# Zespół projektowy

Nazwa zespołu: Zieloni

Skład zespołu projektowego:

|  |  |  |
| --- | --- | --- |
| Imię i nazwisko | Pełniona rola | Zakres czynności / odpowiedzialności |
| Miłosz Gierus | Project Manager | Projekt i implementacja DB, Logika aplikacji |
| Bartłomiej Florek | Ux/Ui Designer, Backend Developer, Database Manager, Tester | JavaFx, Logika aplikacji, Integracja z bazą danych |
| Kacper Długosz | Backend Developer, Ux/Ui Designer | Optymalizacja, Logika biznesowa, Logika aplikacji, JavaFx |
| Ewelina Kaniewska | Ux/Ui Designer, Tester, analityk biznesowy | JavaFx, Logika Biznesowa, Logika aplikacji, testowanie |
| Arkadiusz Haznar | Ux/Ui Designer, Backend Developer, Database Manager | JavaFx, Integracja z bazą danych, Logika aplikacji |

# Specyfikacja projektu

## Cel projektu

System „ObiWanKenobi” został zaprojektowany jako nowoczesne narzędzie wspomagające zarządzanie zadaniami oraz organizację pracy w sklepie ogrodniczo - budowlanym. Jego głównym celem jest eliminacja chaosu organizacyjnego poprzez centralizację informacji, usprawnienie przepływu danych i zwiększenie kontroli nad postępem realizacji działań. System umożliwia sprawne przydzielanie zadań, ich monitorowanie i raportowanie na każdym etapie - od momentu zlecenia, aż po zakończenie. Użytkownicy przypisani do różnych ról mają dostęp do dedykowanych funkcji, które odpowiadają ich zakresowi obowiązków. Dzięki temu możliwa jest jasna hierarchia decyzyjna, a także szybkie reagowanie na zmieniające się potrzeby operacyjne. Jednym z kluczowych aspektów systemu jest poprawa komunikacji między członkami zespołu. System umożliwia szybką wymianę informacji, przesyłanie poleceń oraz dostęp do aktualnych danych, co znacząco ogranicza ryzyko nieporozumień i opóźnień. Dodatkowo, moduły raportowania i wizualizacji danych zapewniają bieżący wgląd w sytuację operacyjną firmy - zarówno w zakresie realizacji zadań, jak i stanu magazynowego, efektywności działów czy kosztów pracowniczych. Dyrektor może w łatwy sposób generować raporty, analizować wyniki i podejmować świadome decyzje na podstawie rzetelnych danych. System został zaprojektowany z myślą o skalowalności, ergonomii oraz prostocie obsługi, dzięki czemu może być wykorzystywany zarówno w mniejszych zespołach, jak i w większych strukturach organizacyjnych. „ObiWanKenobi” to narzędzie, które wprowadza ład, zwiększa efektywność operacyjną i wspiera rozwój firmy poprzez lepsze zarządzanie zasobami ludzkimi i informacyjnymi.

## Zakres projektu

### Zasoby

W ramach realizacji projektu systemu przewidziano wykorzystanie następujących zasobów niezbędnych do jego prawidłowego zaprojektowania, implementacji oraz testowania:

* Zespół programistów (3 osoby): Trzyosobowy zespół deweloperski odpowiedzialny za projektowanie architektury systemu, implementację logiki biznesowej oraz integrację z bazą danych. Praca programistyczna została rozplanowana na 10 godzin tygodniowo przez okres 10 tygodni;
* Zespół testerów (1 osoba): Dwóch testerów manualnych odpowiedzialnych za wykrywanie błędów oraz weryfikację poprawności działania systemu w różnych scenariuszach użytkowych. Testowanie odbywa się w końcowej fazie projektu i przewidziane jest na 5 godzin tygodniowo przez 2 tygodnie;
* UI Designer (1 osoba): Jedna osoba odpowiedzialna za opracowanie spójnego i intuicyjnego interfejsu graficznego, uwzględniającego zasady ergonomii oraz nowoczesne standardy projektowania. Praca projektanta ogranicza się do fazy przygotowawczej i obejmuje 5 godzin pracy;
* Do wytworzenia systemu wykorzystano wyłącznie darmowe, ogólnodostępne narzędzia i środowiska takie jak: IntelliJ, MySQL, GitHub, Jira. Komputery osobiste członków zespołu projektowego zostały wyposażone w to niezbędne oprogramowanie. Stałe łącze internetowe do współpracy zdalnej i synchronizacji danych w repozytorium Github.
* Projekt realizowany w ramach pracy zespołowej bez dodatkowego finansowania - wszystkie wykorzystywane narzędzia są bezpłatne, a członkowie zespołu nie pobierają wynagrodzenia (projekt edukacyjno-rozwojowy).

### Produkty końcowe projektu

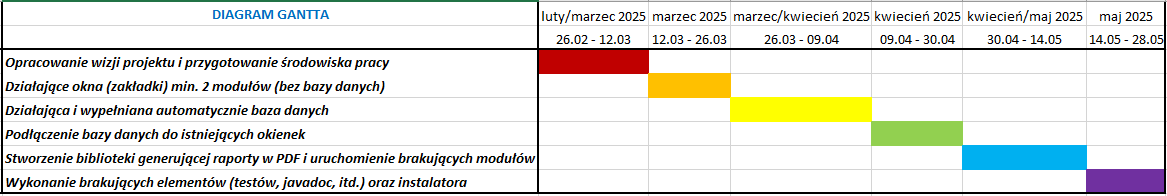
W wyniku realizacji projektu systemu zostaną dostarczone następujące produkty końcowe, obejmujące zarówno komponenty techniczne systemu, jak i pełną dokumentację oraz wsparcie wdrożeniowe i szkoleniowe:

* System do zarządzania zadaniami w „ObiWanKenobi”
  + Pakiet instalacyjny
  + Kod źródłowy aplikacji
  + Skrypty do utworzenia bazy danych
  + Biblioteka do generowania raportów PDF
  + Testy jednostkowe
* Dokumentacja techniczna projektu
* Podręcznik użytkownika dla wytworzonego systemu
* Szkolenie dla użytkowników systemu
* Plan zarządzania projektem
* Repozytorium kodu źródłowego z systemem kontroli wersji
* Dokumentacja procesu wytwarzania oprogramowania

### Harmonogram realizacji projektu

|  |  |
| --- | --- |
| Zakres dat | Zadanie |
| 26.02.2025 - 12.03.2025  12.03.2025 - 26.03.2025  26.03.2025 - 09.04.2025  09.04.2025 - 30.04.2025  30.04.2025 - 14.05.2025  14.05.2025 - 28.05.2025 | Opracowanie wizji projektu i przygotowanie środowiska pracy  Działające okna (zakładki) min. 2 modułów (bez bazy danych)  Działająca i wypełniana automatycznie baza danych  Podłączenie bazy danych do istniejących okienek  Stworzenie biblioteki generującej raporty w PDF i uruchomienie brakujących modułów  Wykonanie brakujących elementów (testów, javadoc, itd.) oraz instalatora |

#### Wykres Gantta



Rysunek . Diagram Gantta

## Wymagania stawiane aplikacji / systemowi

* Aplikacja powinna mieć kilka modułów:
* Moduł administracji użytkownikami (role)
* Moduł zarządzania zadaniami
* Moduł raportów
* Moduł logowania
* System powinien umożliwiać generowanie raportów PDF
* 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 użytkownikami systemu: dodawanie, edycja i usuwanie kont użytkowników.
* Panel dyrektora
  + Główne narzędzie dyrektora umożliwiające generowania raportów oraz podgląd stanów magazynowych na danym dziale.
* Panel kierownika
  + Główne narzędzie kierownika umożliwiające dodawanie, edycję, usuwanie zadań i zatwierdzenie oczekujących zadań wykonanych przez pracowników oraz podgląd stanu magazynu dla swojego działu.
* Panel pracowników
  + Główne narzędzie pracowników umożliwiające podgląd przypisanych zadań danemu pracownikowi wraz ze szczegółowymi danymi i oznaczanie statusów realizacji zadań.
* Panel gościa
  + Główne narzędzie gości umożliwiające dostęp tylko do panelu logowania.
* Panel raportów
* Dostęp do panelu generowania raportów w pdf ma tylko dyrektor.
  1. Typy wymaganych dokumentów w projekcie oraz dostęp do nich

### Raporty PDF

#### Rodzaje raportów PDF

* Raport użytkowników - generowany dla dyrektora, zawiera dane osobowe użytkowników, przypisane im działy oraz adres e-mail. Daje możliwość filtrowania po mieście, dziale, wynagrodzeniu i ilości wykonanych zadań. Ułatwia kontrolę nad strukturą organizacyjną.
* Raport zadań - generowany dla dyrektora, zawiera tytuł, opis, termin realizacji, status, priorytet i przydzielenie dla danej osoby zadania. Daje możliwość filtrowania po statusie zadania, dacie od, dacie do, priorytecie zadania, dziale i przydzielonym pracowniku do tego zadania. Ułatwia kontrolę nad zadaniami.
* Raport magazynu - generowany dla dyrektora, zawiera nazwę magazynu, kierownika magazynu, nazwę przedmiotu i ilość. Daje możliwość filtrowania po kategorii, ilości minimalnej i ilości maksymalnej. Ułatwia kontrolę nad magazynem.

### 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 pracownika.
* Pracownik może aktualizować status zadania.
* Kierownik monitoruje postęp i zatwierdza oczekujące zadania.

1. Generowanie raportów:

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

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, magazynach i przedmiotach.

## Użytkownicy aplikacji i ich uprawnienia

* Administrator
  + zarządzanie użytkownikami i rolami
* Dyrektor
  + generowanie raportów w pdf
* Kierownik
  + dodawanie, edytowanie, usuwanie zadań w obrębie swojego działu
  + monitorowanie postępu oraz zatwierdzanie oczekujących do zatwierdzenia zadań
* Pracownik
  + dostęp do przypisanych zadań
  + modyfikacja statusu swoich zadań
* Gość
  + tylko dostęp do panelu logowania do systemu

## Interesariusze

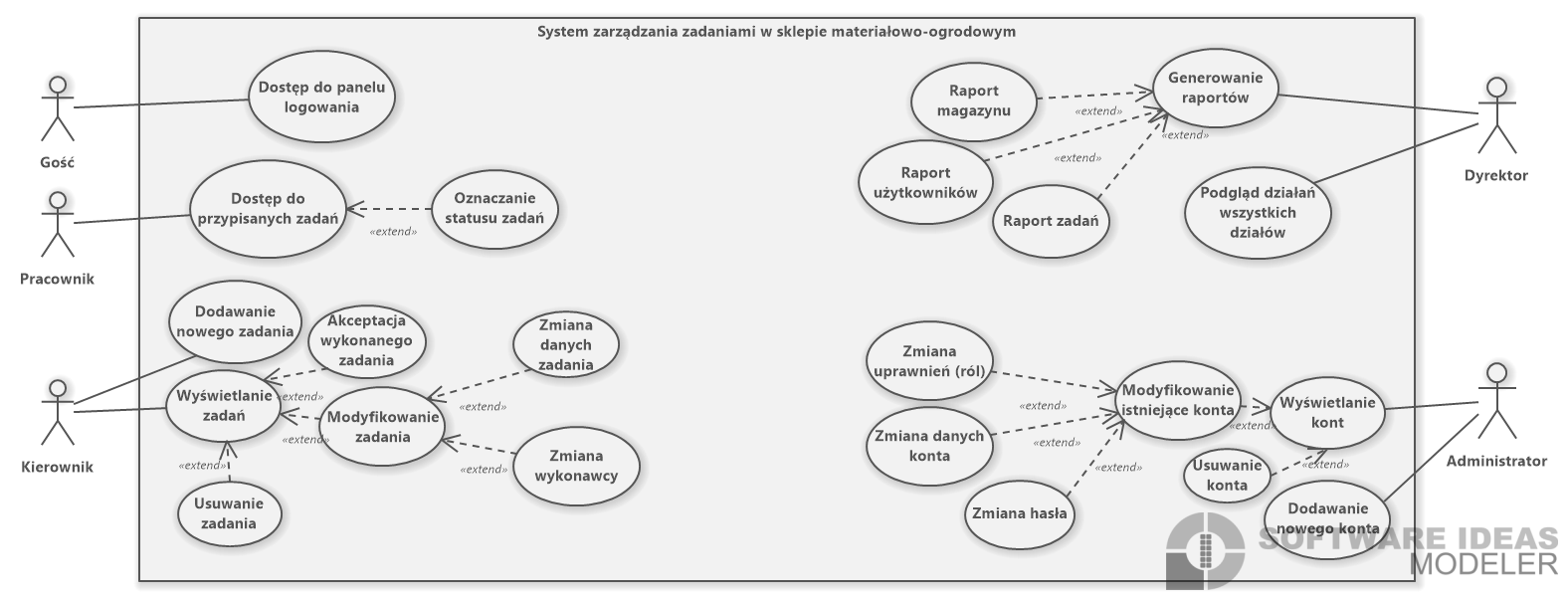
* Interesariusze wewnętrzni
  + Dyrektor
  + Kierownik
  + Pracownik
* Interesariusze zewnętrzni
  + Administrator
  + Gość

# Diagramy UML

## Diagram przypadków użycia

W projektowanej aplikacji wyróżnia się 5 aktorów którzy w zależności od uprawnień posiadają pewne 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.



Rysunek . Diagram przypadków użycia

## Diagramy aktywności

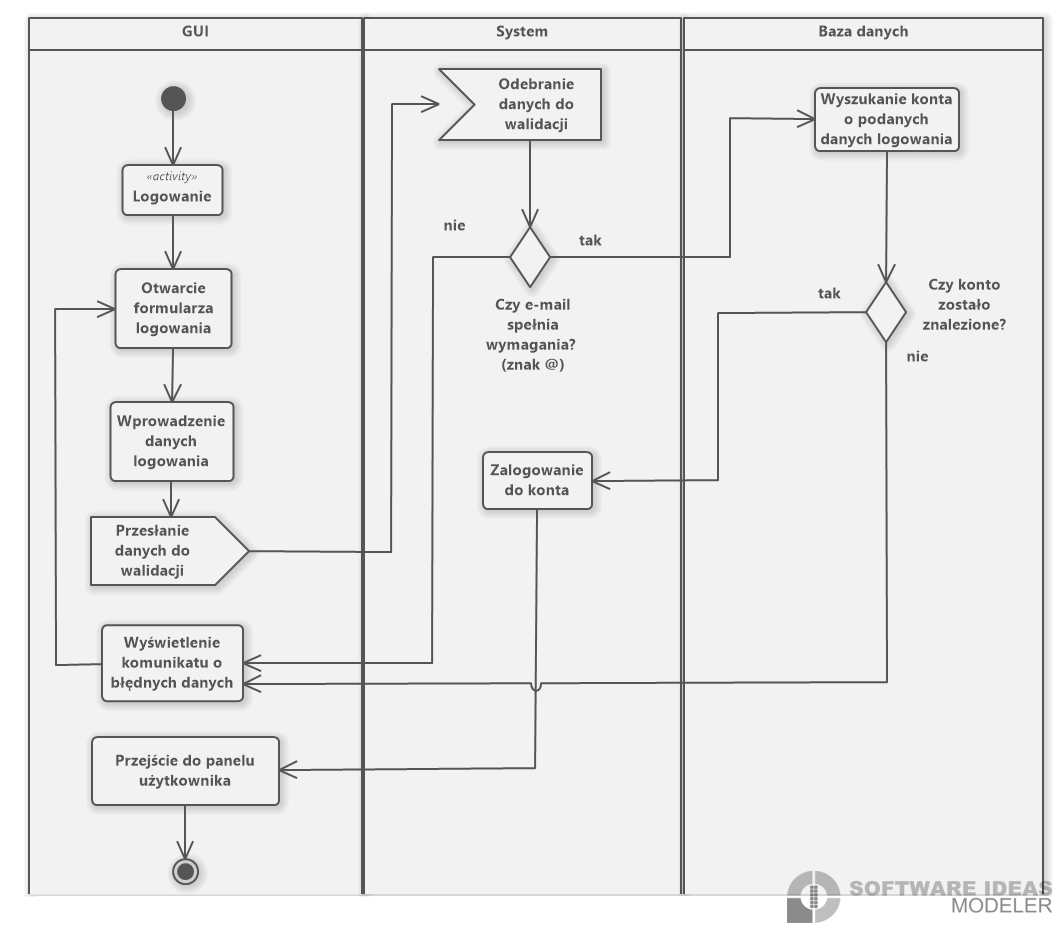
Przypadek użycia 1: Logowanie

* 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 . Diagram aktywności dla przypadku użycia "Logowanie"

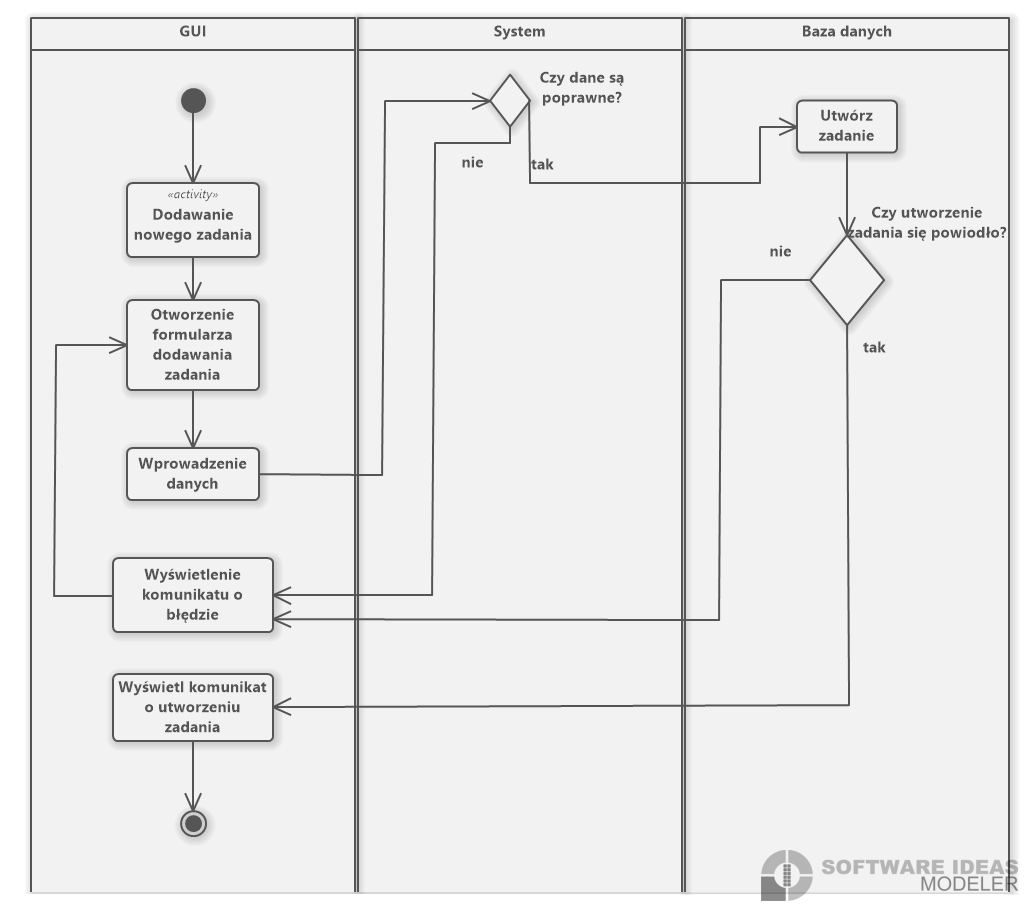
Przypadek użycia 2: Dodawanie nowego zadania

* 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 . Diagram aktywności dla przypadku użycia "Dodawanie nowego zadania"

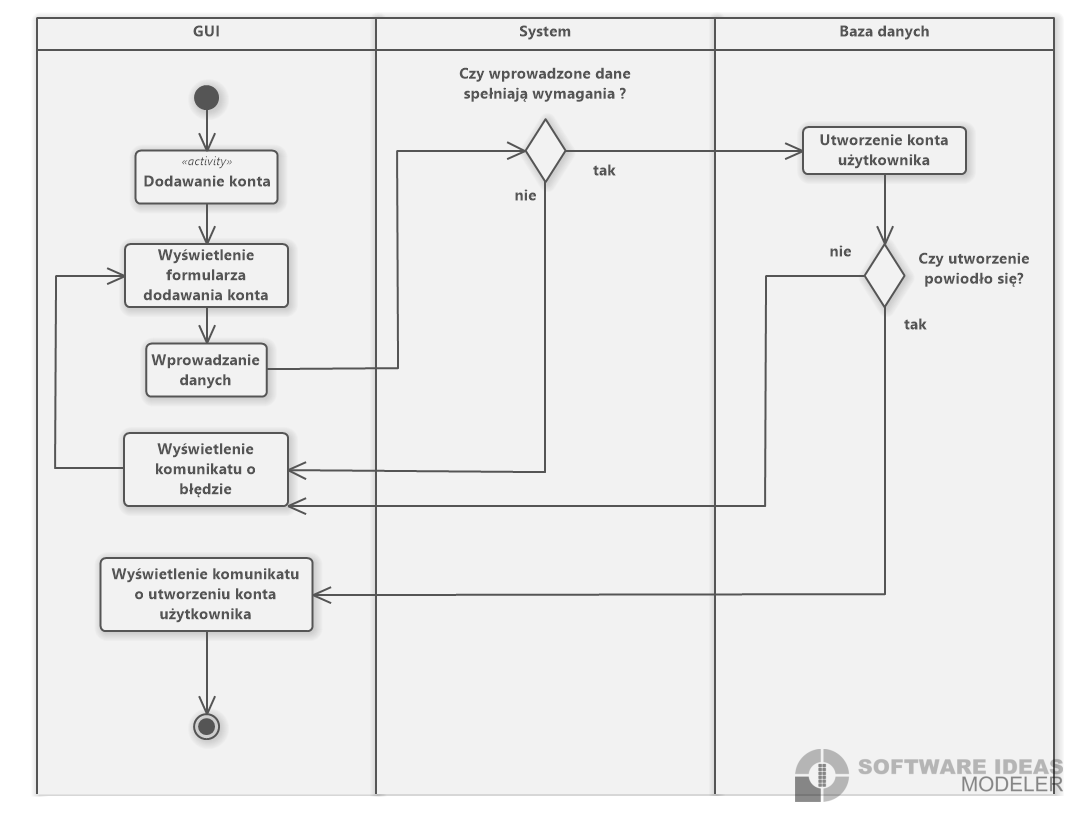
Przypadek użycia 3: Dodawanie nowego konta

* 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 . 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 . 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 . 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.

1. Walidacja danych:

* System sprawdza poprawność wprowadzonych informacji.
* Jeśli dane są niepoprawne, zwracany jest błąd, a interfejs użytkownika informuje administratora o konieczności poprawienia danych.

1. 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.
  1. Niepowodzenie:
* Jeśli proces tworzenia konta w bazie się nie powiedzie, system zwraca informację o błędzie.
* Administrator zostaje poinformowany o problemie i może ponowić próbę.

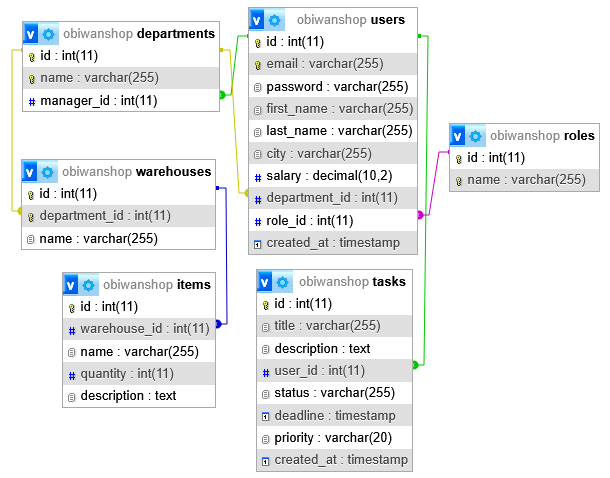
Obraz zawierający zrzut ekranu, tekst

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

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

# Baza danych

## Diagram ERD



## Skrypt do utworzenia struktury bazy danych

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,

city VARCHAR(255) NOT NULL,

salary DECIMAL(10, 2) 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,

priority VARCHAR(20),

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

* IntelliJ (JavaFX)
* MySQL
* GitHub
* Jira

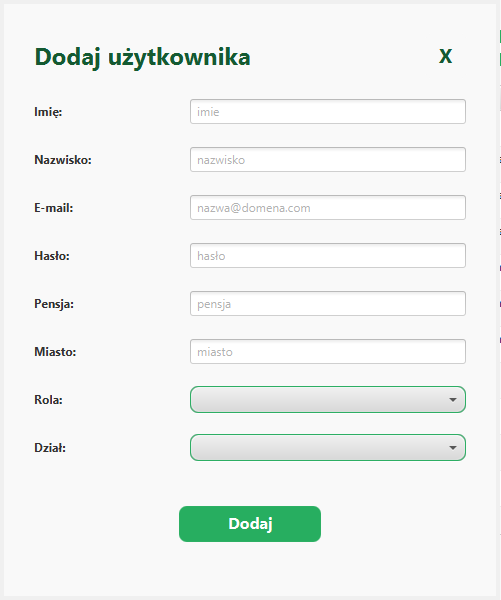
# Interfejs aplikacji / systemu

Na poniżsym rysunku został przedstawiony panel, który pojawia się po uruchomieniu aplikacji. Aby przejść do formularza logowania należy nacisnąć przycisk „start”.



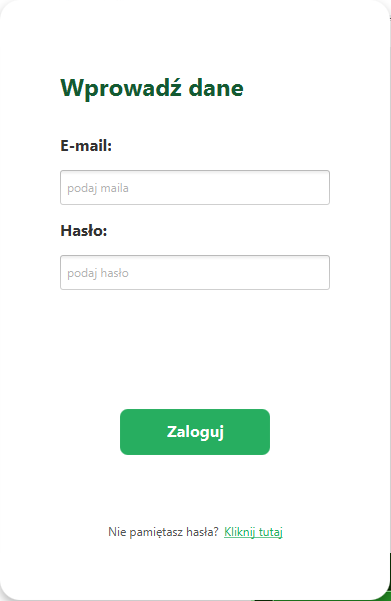
Rysunek Panel startowy

Na poniższym rysunku przedstawiono panel, za pomocą którego możliwe jest dodanie nowego użytkownika. Do tego panelu dostęp ma tylko administrator. Aby dodać nowego użytkownika należy podać jego dane i wcisnąć przycisk „dodaj”.



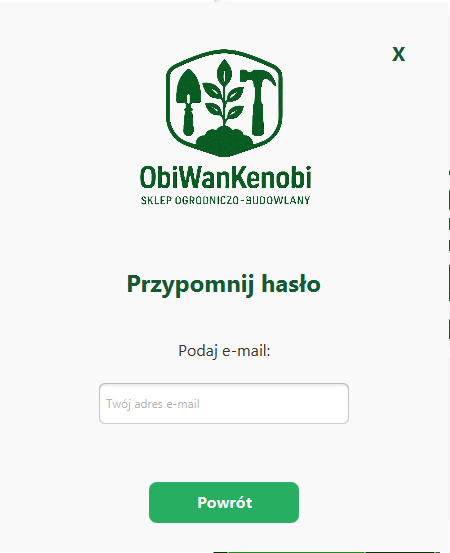
Rysunek Panel dodawania nowego użytkownika

Na poniższym rysunku przedstawiono panel z formularzem logowania do aplikacji. Aby pomyślnie przejść proces logowania należy posiadać konto i znać swoje dane logowania. Z poziomu tego panelu dostępna jest również funkcjonalność resetowania hasła.



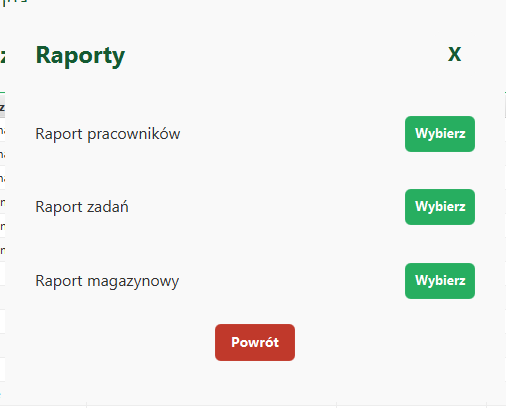
Rysunek Panel logowania

Na poniższym rysunku przedstawiono panel przypominania hasła. Aby odzyskać swoje hasło użytkownik musi podać adres e-mail, który jest powiązany z kontem w aplikacji. Aby wrócić do formularza logowania należy nacisnąć przycisk „powrót”.



Rysunek Panel przypominania hasła

Na poniższym rysunku przedstawiono panel generowania raportów. Tylko dyrektor ma dostęp do tego panelu. Użytkownik ma do wyboru 3 rodzaje raportów. Aby przejść do filtrów konkretnego raportu należy nacisnąć odpowiedni przycisk „Wybierz”.



Rysunek Panel generowania raportów

Na poniższych rysunkach przedstawiono panele filtrowania raportów. Tylko dyrektor ma dostęp do tych paneli. Przycisk „wyczyść obsługuje funkcję czyszczenia formularza. Aby wygenerować raport należy ustawić odpowiednie filtry i nacisnąć przycisk „generuj”.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek Panel filtrowania raportu użytkowników

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

Rysunek Panel filtrowania raportu zadań

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

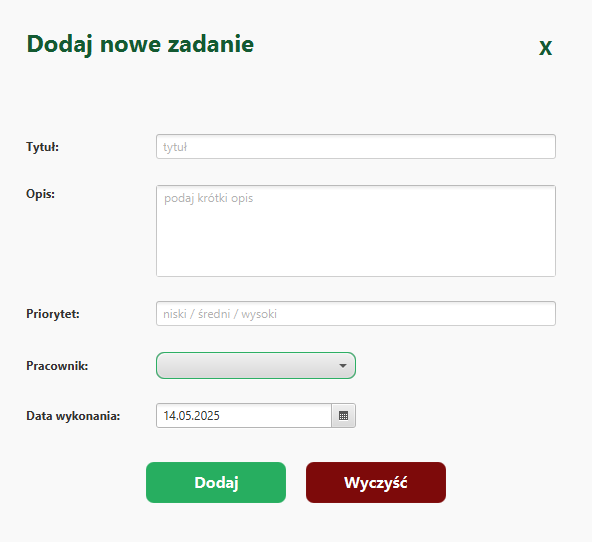
Rysunek Panel filtrowania raportu stanu magazynowego

Na poniższym rysunku przedstawiono panel główny pracownika. Panel pojawia się po zalogowaniu. Znajdują się w nim nadchodzące zadania do wykonania oraz przycisk wyloguj. Aby zobaczyć więcej informacji o zadaniu należy nacisnąć przycisk „szczegóły”. Przycisk „wyloguj” wylogowuje użytkownika z aplikacji.



Rysunek Panel główny użytkownika

Na poniższym rysunku przedstawiono panel dodawania nowego zadania dla kierownika. Tylko kierownik ma dostęp do tego panelu. Aby dodać nowe zadanie należy uzupełnić formularz informacjami o zadaniu i nacisnąć przycisk „dodaj”. Formularz udostępnia również funkcjonalność czyszczenia swoich pól.



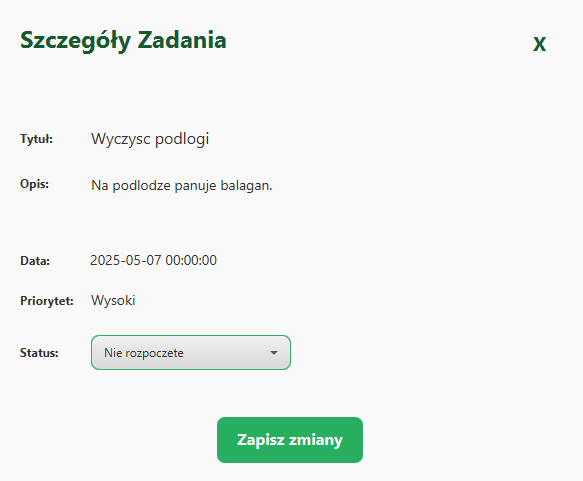
Rysunek Panel dodawania nowego zadania

Na poniższym rysunku przedstawiono panel edytowania istniejącego zadania dla kierownika. Tylko kierownik ma dostęp do tego panelu. Aby edytować zadanie należy uzupełnić formularz informacjami o zadaniu i nacisnąć przycisk „zapisz”. Formularz udostępnia również funkcjonalność czyszczenia swoich pól.Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.

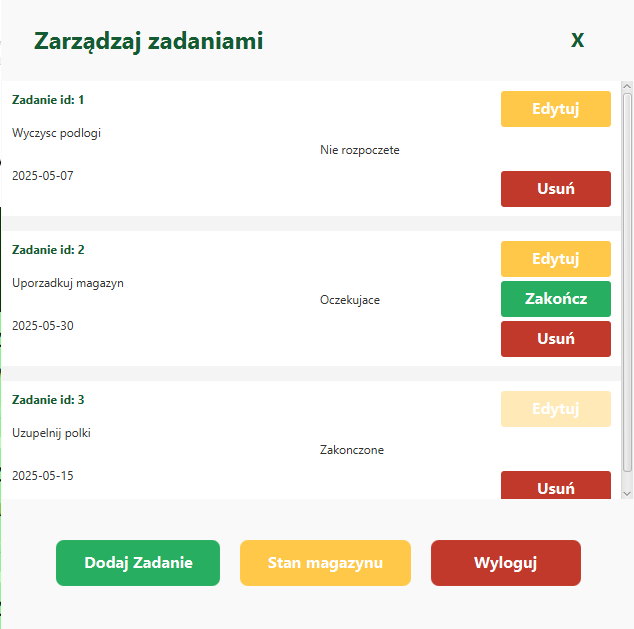
Rysunek Panel edytowania zadania

Na poniższym rysunku przedstawiono panel ze szczegółami zadania. Panel pojawia się w momencie, kiedy użytkownik z poziomu panelu z zadaniami naciśnie odpowiedni przycisk. W panelu widoczne są szczegółowe informacje o wybranym zadaniu oraz dwa przyciski, które użytkownik może wcisnąć, kiedy zakończył zadanie lub napotkał problem w trakcie wykonywania.



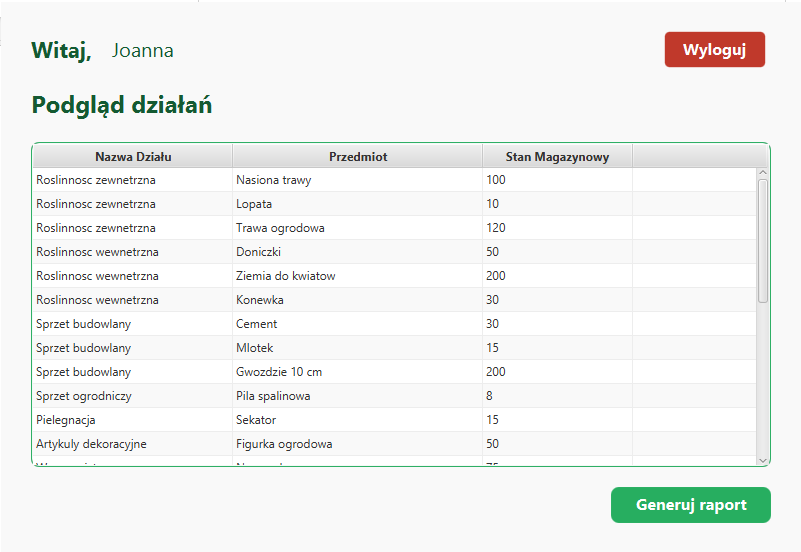
Rysunek Panel szczegółów zadania

Na poniższym rysunku przedstawiono panel zarządzania zadaniami. Dostęp do tego panelu ma tylko kierownik. Z jego poziomu może zatwierdzić wykonanie danego zadania przez pracownika, edytować lub usunąć zadanie. Oprócz tego może też dodać nowe zadanie, sprawdzić stan magazynu oraz wylogować się.



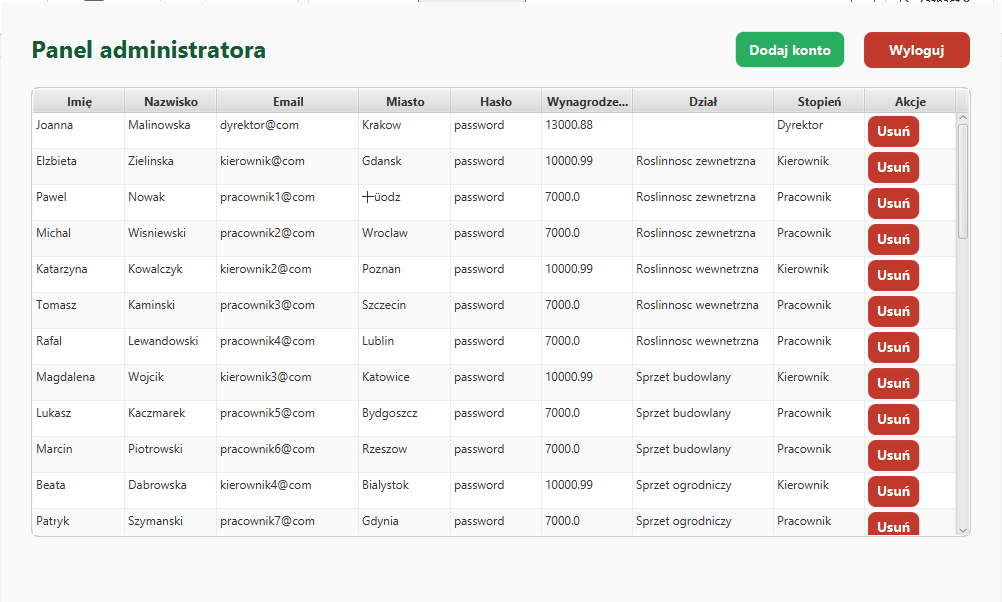
Rysunek Panel zarządzania zadaniami

Na poniższym rysunku przedstawiono panel dyrektora. Tylko dyrektor ma do niego dostęp. Panel pojawia się po zalogowaniu odpowiedniego użytkownika. Z jego poziomu można dostać się do panelu pozwalającego generować raporty oraz przeglądać stany magazynowe przedmiotów.



Rysunek Panel dyrektora

Na poniższym rysunku został przedstawiony panel administratora. Pojawia się on po zalogowaniu przez administratora i pozwala na przejście do panelu dodania nowego konta oraz wylogowanie. W tym panelu administrator może przeglądać informacje o uzytkownikach sklepu.



Rysunek Panel administratora