**Instytut Informatyki  
Kolegium Nauk Przyrodniczych  
Uniwersytet Rzeszowski**

**Przedmiot:**

**Programowanie zespołowe**

**Dokumentacja projektu:**

***System do zarządzania zadaniami  
w szkole***

**Wykonał:**

**Zespół projektowy ZIELONI**

**Prowadzący: mgr inż. Adam Szczur**

**Rzeszów 2024**

Spis treści

[1. Zespół projektowy 3](#_Toc168905556)

[2. Specyfikacja projektu 3](#_Toc168905557)

[2.1. Cel projektu 3](#_Toc168905558)

[2.2. Zakres projektu 3](#_Toc168905559)

[2.2.1. Zasoby 3](#_Toc168905560)

[2.2.2. Produkty końcowe projektu 4](#_Toc168905561)

[2.2.3. Harmonogram realizacji projektu 4](#_Toc168905562)

[2.3. Wymagania stawiane aplikacji / systemowi 4](#_Toc168905563)

[2.4. Panele / zakładki systemu, które będą oferowały potrzebne funkcjonalności 5](#_Toc168905564)

[2.5.1. Raporty PDF 5](#_Toc168905565)

[2.5.2. Inne dokumenty 5](#_Toc168905566)

[2.6. Przepływ informacji w środowisku systemu 6](#_Toc168905567)

[2.7. Użytkownicy aplikacji i ich uprawnienia 6](#_Toc168905568)

[2.8. Interesariusze 7](#_Toc168905569)

[3. Diagramy UML 7](#_Toc168905570)

[3.1. Diagram przypadków użycia 7](#_Toc168905571)

[3.2. Diagramy aktywności 8](#_Toc168905572)

[3.3. Diagramy sekwencji 11](#_Toc168905573)

[4. Baza danych 13](#_Toc168905574)

[4.1. Diagram ERD 13](#_Toc168905575)

[4.2. Skrypt do utworzenia struktury bazy danych 14](#_Toc168905576)

[5. Wykorzystane technologie 17](#_Toc168905577)

[6. Interfejs aplikacji / systemu 18](#_Toc168905578)

# Zespół projektowy

Nazwa zespołu: ZIELONI

Skład zespołu projektowego:

|  |  |  |
| --- | --- | --- |
| **Imię i nazwisko** | **Pełniona rola** | **Zakres czynności / odpowiedzialności** |
| Wojciech Olejko | Project Manager | Zarządzanie zadaniami, programowanie. |
| Dawid Madej | Programista Frontend | Projektowanie i wdrażanie interfejsu oprogramowania. |
| Filip Kaczmarczyk | Programista Backend | Programowanie, testowanie oprogramowania. |
| Krzysztof Kaczka | Programista Backend | Programowanie, zarządzanie i integracja z bazą danych. |
| Tomasz Nowak | Programista Fullstack | Programowanie, integracja interfejsu z backend’em. |

# Specyfikacja projektu

## Cel projektu

Stworzenie systemu zarządzania zadaniami w szkole mającego na celu usprawnienie procesu przypisywania, monitorowania i oceniania zadań dla uczniów oraz umożliwienie skutecznego śledzenia postępów uczniów przez nauczycieli, dyrektorów oraz ich rodziców.

## Zakres projektu

### Zasoby

* Programista Frontend, 5 godzin pracy tygodniowo przez 6 tygodni
* Zespół dwóch programistów Backend, 8 godzin pracy tygodniowo przez 10 tygodni
* Programista Fullstack, 4 godziny pracy tygodniowo przez 10 tygodni
* Darmowe narzędzia do wytworzenia systemu
  + MySQL
  + GitHub
  + Jira
  + Figma
  + JetBrains WebStorm (licencja studencka)
* Zasoby finansowe
  + Ja nie płace za naukę ~ Dariusz Jabłoński (Giftpol.pl)

### Produkty końcowe projektu

* System do zarządzania zadaniami w szkole
  + Pakiet instalacyjny
  + Kod źródłowy aplikacji
  + Skrypty do utworzenia bazy danych
  + Graficzny interfejs do zarządzania systemem
* Dokumentacja techniczna projektu
* Szkolenie dla użytkowników systemu
* Wdrożenie systemu

### Harmonogram realizacji projektu

|  |  |
| --- | --- |
| **ZAKRES DAT** | **ZADANIE** |
| 01.03.2024 – 03.03.2024 | Opracowanie specyfikacji systemu |
| 04.03.2024 – 25.03.2024 | Stworzenie dokumentacji technicznej |
| 18.03.2024 – 27.03.2024 | Zaprojektowanie interfejsu graficznego |
| 28.03.2024 – 08.04.2024 | Zaimplementowanie interfejsu graficznego |
| 08.04.2024 – 15.04.2024 | Zaprojektowanie bazy danych |
| 16.04.2024 – 22.04.2024 | Zaimplementowanie bazy danych |
| 23.04.2024 – 30.04.2024 | Integracja interfejsu z bazą danych |
| 01.05.2024 – 06.05.2024 | Testowanie aplikacji |
| 07.05.2024 – 15.05.2024 | Stworzenie biblioteki do generowania raportów |
| 20.05.2024 – 25.05.2024 | Stworzenie pakietów instalacyjnych |
| 26.05.2024 – 03.06.2024 | Końcowe poprawki |

#### Wykres Gantta

A screenshot of a computer

Description automatically generated

Rysunek . Wykres Gantta

## Wymagania stawiane aplikacji / systemowi

* System powinien mieć kilka modułów
  + Moduł administracji użytkownikami (role)
  + Moduł raportów
  + Moduł zarządzania zadaniami
  + Moduł konfiguracji
  + Moduł logowania/rejestracji
  + Moduł powiadomień
* System powinien umożliwiać generowanie raportów PDF
* System umożliwia nauczycielom tworzenie, przypisywanie i monitorowanie zadań dla swoich klas, takich jak zadania domowe, projekty
* System powinien współpracować z bazą danych
* System powinien posiadać intuicyjny i przyjazny dla użytkownika interfejs
* System powinien mieć wbudowane funkcje bezpieczeństwa

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

* Panel dyrekcji (administratora)
  + Pozwala administratorom szkoły zarządzać kontami użytkowników, przypisywać im role (np. uczniowie, nauczyciele, dyrektorzy) oraz zarządzać uprawnieniami dostępu.
* Panel nauczyciela
  + Umożliwia nauczycielom tworzenie, przypisywanie i monitorowanie zadań dla swoich klas, takich jak zadania domowe, projekty.
* Panel ucznia
  + Zapewnia uczniom dostęp do przypisanych im zadań, możliwość przesyłania rozwiązań oraz przeglądanie ocen i komentarzy od nauczycieli. Uczniowie mogą również śledzić swoje postępy, planować swoje zadania.
* Panel rodzica
  + Zapewnia rodzicom dostęp do zadań przypisanych ich dzieciom, możliwość przeglądania ocen i komentarzy od nauczycieli. Rodzice mogą również śledzić postępy swoich dzieci.
* Zakładka raportów
  + W tej zakładce nauczyciele i dyrektorzy mogą generować i przeglądać raporty dotyczące postępów uczniów. Raporty te mogą być wykorzystywane do analizy i doskonalenia procesów dydaktycznych oraz do personalizacji procesu nauczania dla poszczególnych uczniów.
  + Rodzice również mogą generować raporty dotyczące postępów swoich dzieci.
* Okno ustawień
  + Pozwala wszystkim użytkownikom systemu na personalizację ich doświadczenia z aplikacją. Mogą tutaj zmieniać swoje dane osobowe, hasła, preferencje dotyczące powiadomień oraz inne ustawienia związane z ich kontem i interakcją z systemem.
  1. Typy wymaganych dokumentów w projekcie oraz dostęp do nich

### Raporty PDF

#### Rodzaje raportów PDF

* Raport dla nauczyciela o uczniach i ich postępach w nauce
* Raporty dla dyrekcji o nauczycielach o ich postępach w pracy + o uczniach tak jak w przypadku nauczycieli
* Raport dla rodzica o postępach w nauce swoich dzieci.

### Inne dokumenty

* Brak

## Przepływ informacji w środowisku systemu

System opiera się o scentralizowaną bazę danych znajdującą się na serwerze dostawcy oprogramowania. Dla każdej placówki korzystającej z systemu, zostaje utworzona osobna instancja bazy, którą placówka może zarządzać za pośrednictwem aplikacji. Komunikacja między modułami w systemie została przedstawiona na Rysunku 2.

A diagram of a system

Description automatically generated

Rysunek . Komunikacja między modułami

## Użytkownicy aplikacji i ich uprawnienia

* Dyrekcja
  + Generowanie raportów dotyczących postępów nauczycieli oraz uczniów.
  + Zarządzanie zadaniami dla nauczycieli.
* Nauczyciele
  + Zarządzanie zadaniami dla swoich uczniów, w tym tworzenie, przypisywanie, edytowanie i ocenianie zadań.
  + Wgląd w postępy uczniów oraz raporty dotyczące ich osiągnięć.
* Uczniowie
  + Przeglądanie i wykonywanie przypisanych im zadań.
  + Wysyłanie rozwiązań zadań.
  + Przeglądanie własnych ocen i postępów w nauce.
* Rodzic
  + Przeglądanie zadań przypisanych dzieciom.
  + Przeglądanie ocen i postępów w nauce dziecka.

## Interesariusze

* Interesariusze wewnętrzni
  + Dyrekcja
  + Nauczyciele
* Interesariusze zewnętrzni
  + Uczniowie
  + Rodzice
  + Organizacje edukacyjne lub inspektoraty

# Diagramy UML

## Diagram przypadków użycia

A diagram of a system

Description automatically generated

Rysunek . Diagram przypadków użycia

## Diagramy aktywności

A screenshot of a computer

Description automatically generated

Rysunek . Diagram aktywności dla PU Generowanie raportu

A diagram of a flowchart

Description automatically generated

Rysunek . Diagram aktywności dla PU Tworzenie zadań

A diagram of a software project

Description automatically generated

Rysunek . Diagram aktywności dla PU Wykonywanie zadania

## Diagramy sekwencji

A diagram of a diagram

Description automatically generated

Rysunek . Diagram sekwencji dla PU Generowanie raportu

A screenshot of a diagram

Description automatically generated

Rysunek . Diagram sekwencji dla PU Tworzenie zadania

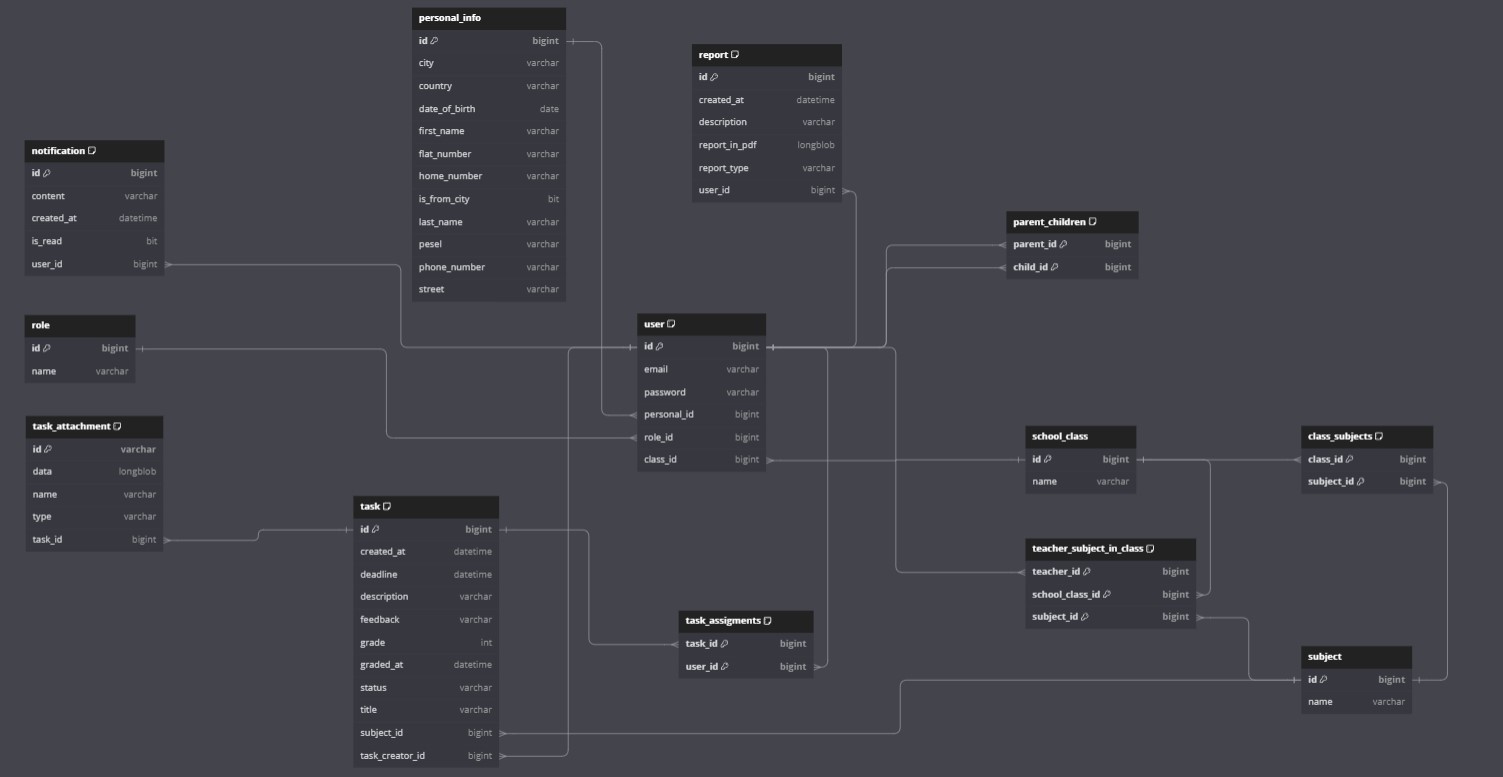
A diagram of a computer program

Description automatically generated

Rysunek . Diagram sekwencji dla PU Wykonywanie zadania

# Baza danych

## Diagram ERD



Rysunek Diagram ERD

## Skrypt do utworzenia struktury bazy danych

SET SQL\_MODE="NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time\_zone="+00:00";

CREATE TABLE `class\_subjects` (`class\_id` bigint(20) NOT NULL, `subject\_id` bigint(20) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `notification` (`id` bigint(20) NOT NULL, `content` varchar(255) DEFAULT NULL, `created\_at` datetime(6) DEFAULT NULL, `is\_read` bit(1) DEFAULT NULL, `user\_id` bigint(20) DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `parent\_children` (`parent\_id` bigint(20) NOT NULL, `child\_id` bigint(20) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `personal\_info` (`id` bigint(20) NOT NULL, `city` varchar(255) DEFAULT NULL, `country` varchar(255) DEFAULT NULL, `date\_of\_birth` date DEFAULT NULL, `first\_name` varchar(255) DEFAULT NULL, `flat\_number` varchar(255) DEFAULT NULL, `home\_number` varchar(255) DEFAULT NULL, `is\_from\_city` bit(1) DEFAULT NULL, `last\_name` varchar(255) DEFAULT NULL, `pesel` varchar(255) DEFAULT NULL, `phone\_number` varchar(255) DEFAULT NULL, `street` varchar(255) DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `report` (`id` bigint(20) NOT NULL, `created\_at` datetime(6) DEFAULT NULL, `description` varchar(255) DEFAULT NULL, `report\_in\_pdf` longblob DEFAULT NULL, `report\_type` varchar(255) DEFAULT NULL, `user\_id` bigint(20) DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `role` (`id` bigint(20) NOT NULL, `name` varchar(255) DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `school\_class` (`id` bigint(20) NOT NULL, `name` varchar(255) DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `subject` (`id` bigint(20) NOT NULL, `name` varchar(255) DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `task` (`id` bigint(20) NOT NULL, `created\_at` datetime(6) DEFAULT NULL, `deadline` datetime(6) DEFAULT NULL, `description` varchar(255) DEFAULT NULL, `feedback` varchar(255) DEFAULT NULL, `grade` int(11) DEFAULT NULL, `graded\_at` datetime(6) DEFAULT NULL, `status` varchar(255) DEFAULT NULL, `title` varchar(255) DEFAULT NULL, `subject\_id` bigint(20) DEFAULT NULL, `task\_creator\_id` bigint(20) DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `task\_assigments` (`task\_id` bigint(20) NOT NULL, `user\_id` bigint(20) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `task\_attachment` (`id` varchar(255) NOT NULL, `data` longblob DEFAULT NULL, `name` varchar(255) DEFAULT NULL, `type` varchar(255) DEFAULT NULL, `task\_id` bigint(20) DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `teacher\_subject\_in\_class` (`teacher\_id` bigint(20) NOT NULL, `school\_class\_id` bigint(20) NOT NULL, `subject\_id` bigint(20) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `user` (`id` bigint(20) NOT NULL, `email` varchar(255) DEFAULT NULL, `password` varchar(255) DEFAULT NULL, `personal\_id` bigint(20) DEFAULT NULL, `role\_id` bigint(20) DEFAULT NULL, `class\_id` bigint(20) DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

ALTER TABLE `class\_subjects` ADD PRIMARY KEY (`class\_id`,`subject\_id`), ADD KEY `FKlfbmt51w06n4kcm7iatjq7m5r` (`subject\_id`);

ALTER TABLE `notification` ADD PRIMARY KEY (`id`), ADD KEY `FKb0yvoep4h4k92ipon31wmdf7e` (`user\_id`);

ALTER TABLE `parent\_children` ADD PRIMARY KEY (`parent\_id`,`child\_id`), ADD KEY `FKmuaapgbnjgx6cxdsu7ndpsw0r` (`child\_id`);

ALTER TABLE `personal\_info` ADD PRIMARY KEY (`id`);

ALTER TABLE `report` ADD PRIMARY KEY (`id`), ADD KEY `FKj62onw73yx1qnmd57tcaa9q3a` (`user\_id`);

ALTER TABLE `role` ADD PRIMARY KEY (`id`);

ALTER TABLE `school\_class` ADD PRIMARY KEY (`id`);

ALTER TABLE `subject` ADD PRIMARY KEY (`id`);

ALTER TABLE `task` ADD PRIMARY KEY (`id`), ADD KEY `FK5k22wv8pvap89p7wpo0ghs95g` (`subject\_id`), ADD KEY `FKm1e3ujho15jj9tsq4trxpha76` (`task\_creator\_id`);

ALTER TABLE `task\_assigments` ADD PRIMARY KEY (`task\_id`,`user\_id`), ADD KEY `FK1jvh56p85qi5qh7ut08cwxhad` (`user\_id`);

ALTER TABLE `task\_attachment` ADD PRIMARY KEY (`id`), ADD KEY `FKkhw6fprv9kv6uio43mem40px6` (`task\_id`);

ALTER TABLE `teacher\_subject\_in\_class` ADD PRIMARY KEY (`school\_class\_id`,`subject\_id`,`teacher\_id`), ADD KEY `FK4ycy3d1gup8pnsadlavtmxcf7` (`subject\_id`), ADD KEY `FKq0x6nuhd9ky2b28589c2u32pv` (`teacher\_id`);

ALTER TABLE `user` ADD PRIMARY KEY (`id`), ADD KEY `FK68xkm4u0qf2bmg9qmukdnclh2` (`personal\_id`), ADD KEY `FKn82ha3ccdebhokx3a8fgdqeyy` (`role\_id`), ADD KEY `FKl59angloqtshc0dbj3mnvybt7` (`class\_id`);

ALTER TABLE `notification` MODIFY `id` bigint(20) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `personal\_info` MODIFY `id` bigint(20) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `report` MODIFY `id` bigint(20) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `role` MODIFY `id` bigint(20) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `school\_class` MODIFY `id` bigint(20) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `subject` MODIFY `id` bigint(20) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `task` MODIFY `id` bigint(20) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `user` MODIFY `id` bigint(20) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `class\_subjects` ADD CONSTRAINT `FK4cemxm83n2j30cgfox390gch5` FOREIGN KEY (`class\_id`) REFERENCES `school\_class` (`id`), ADD CONSTRAINT `FKlfbmt51w06n4kcm7iatjq7m5r` FOREIGN KEY (`subject\_id`) REFERENCES `subject` (`id`);

ALTER TABLE `notification` ADD CONSTRAINT `FKb0yvoep4h4k92ipon31wmdf7e` FOREIGN KEY (`user\_id`) REFERENCES `user` (`id`);

ALTER TABLE `parent\_children` ADD CONSTRAINT `FKmuaapgbnjgx6cxdsu7ndpsw0r` FOREIGN KEY (`child\_id`) REFERENCES `user` (`id`), ADD CONSTRAINT `FKmw9ra9he829lpiu7tj0tqn3yo` FOREIGN KEY (`parent\_id`) REFERENCES `user` (`id`);

ALTER TABLE `report` ADD CONSTRAINT `FKj62onw73yx1qnmd57tcaa9q3a` FOREIGN KEY (`user\_id`) REFERENCES `user` (`id`);

ALTER TABLE `task` ADD CONSTRAINT `FK5k22wv8pvap89p7wpo0ghs95g` FOREIGN KEY (`subject\_id`) REFERENCES `subject` (`id`), ADD CONSTRAINT `FKm1e3ujho15jj9tsq4trxpha76` FOREIGN KEY (`task\_creator\_id`) REFERENCES `user` (`id`);

ALTER TABLE `task\_assigments` ADD CONSTRAINT `FK1jvh56p85qi5qh7ut08cwxhad` FOREIGN KEY (`user\_id`) REFERENCES `user` (`id`), ADD CONSTRAINT `FKnpjl9y4qbr7q652fqve9lmmrh` FOREIGN KEY (`task\_id`) REFERENCES `task` (`id`);

ALTER TABLE `task\_attachment` ADD CONSTRAINT `FKkhw6fprv9kv6uio43mem40px6` FOREIGN KEY (`task\_id`) REFERENCES `task` (`id`);

ALTER TABLE `teacher\_subject\_in\_class` ADD CONSTRAINT `FK4ycy3d1gup8pnsadlavtmxcf7` FOREIGN KEY (`subject\_id`) REFERENCES `subject` (`id`), ADD CONSTRAINT `FKj19qnrj2sqo9bymwusnkecqux` FOREIGN KEY (`school\_class\_id`) REFERENCES `school\_class` (`id`), ADD CONSTRAINT `FKq0x6nuhd9ky2b28589c2u32pv` FOREIGN KEY (`teacher\_id`) REFERENCES `user` (`id`);

ALTER TABLE `user` ADD CONSTRAINT `FK68xkm4u0qf2bmg9qmukdnclh2` FOREIGN KEY (`personal\_id`) REFERENCES `personal\_info` (`id`), ADD CONSTRAINT `FKl59angloqtshc0dbj3mnvybt7` FOREIGN KEY (`class\_id`) REFERENCES `school\_class` (`id`), ADD CONSTRAINT `FKn82ha3ccdebhokx3a8fgdqeyy` FOREIGN KEY (`role\_id`) REFERENCES `role` (`id`);

COMMIT;

# Wykorzystane technologie

* Język Java 17
  + Java 17 to stabilna wersja języka programowania Java, która oferuje wiele nowych funkcji i usprawnień, takich jak pattern matching i ulepszenia w zarządzaniu pamięcią. Jest to wersja LTS (Long-Term Support), co zapewnia długoterminowe wsparcie i stabilność.
* Spring Boot
  + Spring Boot to framework Java, który upraszcza tworzenie samodzielnych, produkcyjnych aplikacji oparte na Spring. Dzięki predefiniowanym ustawieniom i automatycznej konfiguracji, programiści mogą szybko rozpocząć pracę nad nowymi projektami.
* Baza danych MySQL
  + MySQL to system zarządzania relacyjnymi bazami danych, który zapewnia szybki dostęp do danych oraz możliwość obsługi dużych ilości informacji. Jest to jedno z najczęściej używanych rozwiązań bazodanowych na świecie.
* React
  + React to biblioteka JavaScript stworzona przez Facebooka, która umożliwia budowanie interaktywnych interfejsów użytkownika. Dzięki komponentowej architekturze, React ułatwia tworzenie złożonych aplikacji webowych.
* Vite
  + Vite to narzędzie do budowania aplikacji front-end, które oferuje błyskawiczne ładowanie modułów i natychmiastowe odświeżanie stron. Zostało zaprojektowane z myślą o współczesnych frameworkach JavaScript, takich jak React i Vue.
* Electron
  + Vite to narzędzie do budowania aplikacji front-end, które oferuje błyskawiczne ładowanie modułów i natychmiastowe odświeżanie stron. Zostało zaprojektowane z myślą o współczesnych frameworkach JavaScript, takich jak React i Vue.

# Interfejs aplikacji / systemu

Rysunek 11 przedstawia ekran ładowania aplikacji, gdy ta nie nawiązała jeszcze połączenia z bazą danych.

A purple background with white text

Description automatically generated

Rysunek . Ekran ładowania aplikacji

Na Rysunku 12 został przedstawiony ekran logowania do aplikacji.



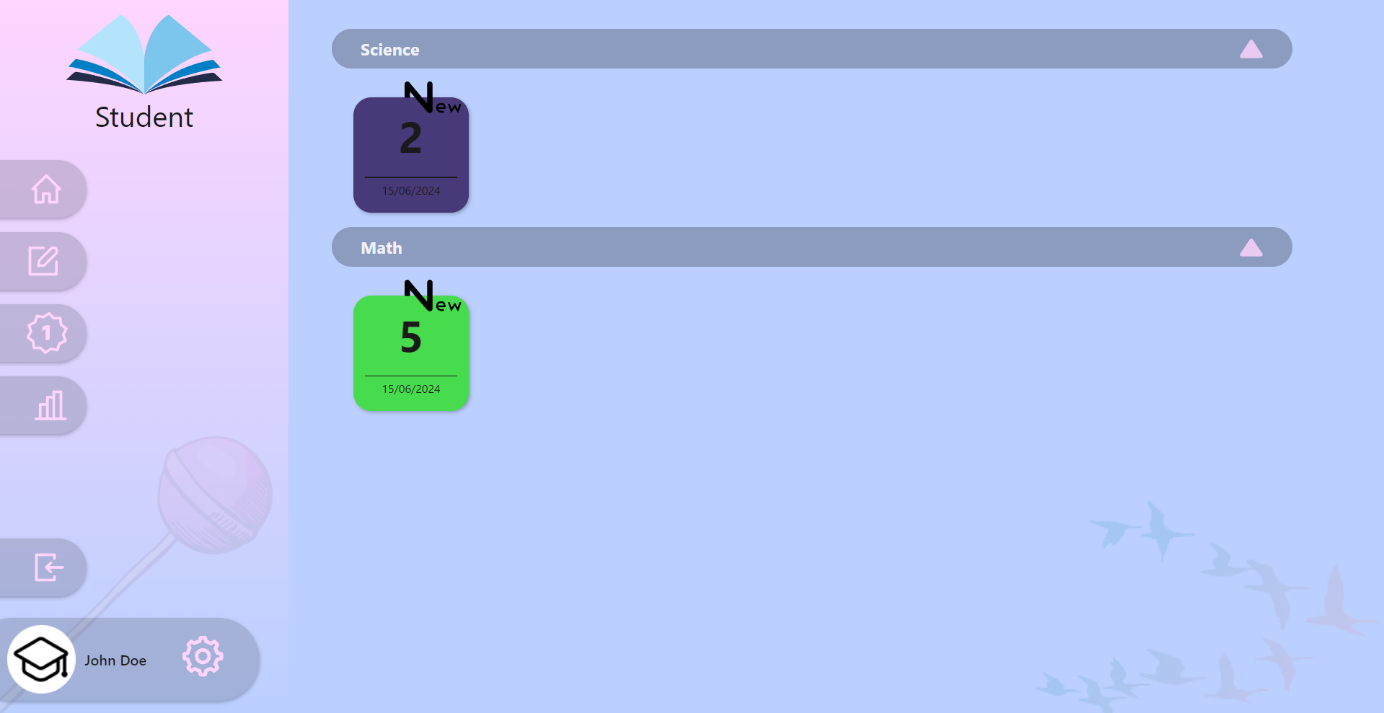
Rysunek . Logowanie

Użytkownik po zalogowaniu trafia do ekranu głównego. Na Rysunku 13 został przedstawiony ekran główny dla ucznia, który przedstawia ostatnie powiadomienia z aplikacji, ostatnią otrzymaną ocenę oraz zadanie z najbliższym terminem wykonania.



Rysunek . Ekran główny ucznia

Rysunek 14 przedstawia ekran ocen ucznia, w którym uczeń może sprawdzić swoje oceny za wykonane zadania z danych przedmiotów.



Rysunek . Ekran ocen ucznia

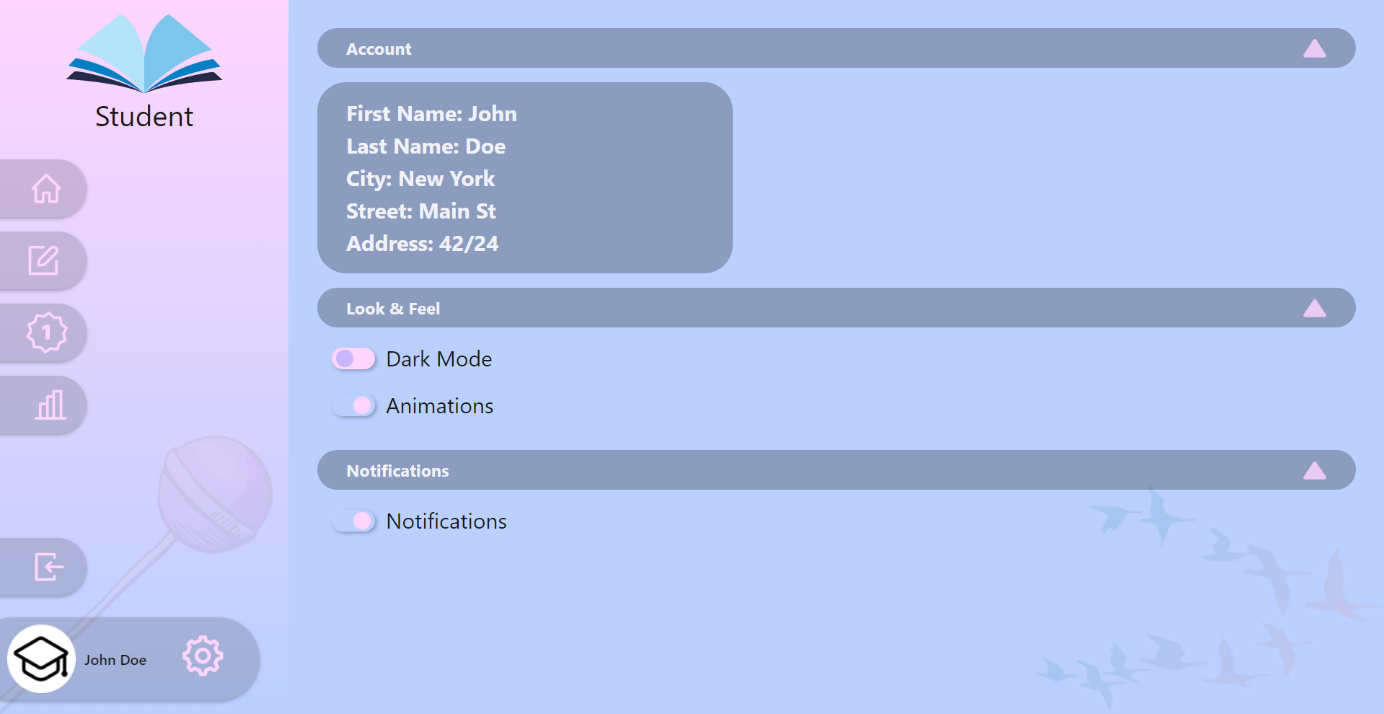
Na Rysunku 15 przedstawiony jest ekran zadań ucznia, w którym uczeń może sprawdzić przypisane mu zadania, przeciągać do odpowiednich sekcji oraz sprawdzać ostatnio ocenione zadania.

A screenshot of a computer

Description automatically generated

Rysunek . Ekran zadań ucznia

Ekran ustawień przedstawiony na Rysunku 16 przedstawia podstawowe dane użytkownika, umożliwia zmianę trybu wyświetlania (jasny, ciemny), wyłączenie animacji oraz powiadomień.



Rysunek . Ekran ustawień

Na Rysunku 17 znajduje się zakładka eksportu danych w której uczeń może wygenerować raport na temat swoich ocen z wybranych przedmiotów.

A screenshot of a video game

Description automatically generated

Rysunek . Ekran eksportu danych

Rysunek 18 przedstawia zakładkę zarządzania użytkownikami która jest do dyspozycji administratora. Składa się ona z 4 zakładek w których administrator może dodawać, usuwać bądź zmieniać dane kolejno nauczycieli, uczniów, przedmiotów w klasach, rodziców.

A screenshot of a computer

Description automatically generated

Rysunek . Ekran zarządzania użytkownikami

Administrator posiada również zakładkę eksportu przedstawioną na Rysunku 19, która umożliwia mu wygenerowanie raportu na temat wybranego nauczyciela w wybranym przedziale czasowym. Może również wygenerować raport na temat uczniów szkoły (School Principal report).

A screenshot of a computer

Description automatically generated

Rysunek . Ekran eksportu administratora

Rysunek 20 przedstawia ekran rodzica, który to może wybrać jedną ze swoich pociech i sprawdzić jej postępy w nauce.

A screenshot of a computer

Description automatically generated

Rysunek . Ekran eksportu na koncie rodzica