

# Dokumentacja Electronic Gradebook

Mateusz Wojaczek, Witoria Klejdysz, Aleksander Wieczorek

## 1 Opis ogólny projektu

Electronic Gradebook to aplikacja typu e-dziennik umożliwiająca zarządzanie ocenami, planami lekcji, obecnością uczniów oraz kontami użytkowników (nauczycieli, uczniów, administratorów). System składa się z aplikacji desktopowej napisanej w JavaFX (frontend) oraz backendu w technologii Spring Boot z dostępem do bazy danych PostgreSQL.

## 2 Funkcjonalności systemu

### 2.1 Panel nauczyciela

- **Przegląd planu lekcji** – nauczyciel może zobaczyć plan swoich zajęć w formie siatki. Plan pobierany jest z backendu i renderowany w komponencie GridPane.
- **Wystawianie ocen** – nauczyciel może kliknąć na daną lekcję, wyświetlić listę uczniów i przydzielić ocenę za pomocą formularza.
- **Przegląd ocen uczniów** – po wybraniu przedmiotu i klasy wyświetlana jest tabela z uczniami i ich ocenami. Dostępne są przyciski edycji i usuwania.
- **Obecności** – widok listy obecności z możliwością oznaczenia obecności/nieobecności dla danego dnia i ucznia.
- **Przycisk WRÓĆ** – umożliwia powrót do panelu głównego nauczyciela z dowolnego widoku podrzednego.

### 2.2 Panel ucznia

- **Plan lekcji** – wyświetlany jest plan w postaci siatki. Lekcje są oznaczone kolorowymi blokami z nazwami przedmiotów.
- **Oceny** – dla każdego przedmiotu uczeń może zobaczyć przydzielone mu oceny w formie tabeli.
- **Obecności** – lista obecności z informacjami o dacie, przedmiocie i statusie obecności.
- **Przycisk WRÓĆ** – umożliwia powrót do panelu głównego ucznia z widoków szczegółowych.

## 2.3 Panel administratora

- **Zarządzanie kontami** – administrator może dodać, edytować lub usunąć użytkowników (uczniów, nauczycieli).
- **Zarządzanie planem lekcji** – edycja planu lekcji dla klas i nauczycieli poprzez interfejs siatki z możliwością dodawania/zmiany przedmiotów i godzin.
- **Zarządzanie przedmiotami i przypisaniami** – administrator przypisuje nauczycieli do przedmiotów i klas.
- **Przycisk ZAPISZ PLAN** – zatwierdza zmiany wprowadzone w siatce planu lekcji i przesyła je do backendu.

## 3 Opis przycisków i ich funkcji

- **WRÓĆ** – powrót do poprzedniego widoku lub głównego panelu danej roli użytkownika.
- **Dodaj ocene** – otwiera formularz dodania nowej oceny; po zatwierdzeniu wysyłane jest żądanie POST do backendu z informacjami o uczniu, ocenie i przedmiocie.
- **Usuń ocene** – usuwa konkretną ocenę; uruchamia żądanie DELETE do backendu, a następnie odświeża widok ocen.
- **Edytuj ocene** – otwiera formularz edycji istniejącej oceny; po zatwierdzeniu wysyła żądanie PUT.
- **Zapisz plan** – przesyła cały aktualny widok siatki planu do backendu i zapisuje go w bazie danych.
- **Pokaż obecność** – ładuje i wyświetla dane obecności dla wybranego ucznia.
- **Zapisz obecność** – umożliwia nauczycielowi zapisanie oznaczonych obecności dla danego dnia.

## 4 Architektura systemu

System opiera się na architekturze klient-serwer. Frontend (JavaFX) komunikuje się z backendem (Spring Boot) przy użyciu REST API.

### 4.1 Frontend (JavaFX)

- Interfejs użytkownika zbudowany przy użyciu FXML.
- Kontrolery Java obsługujące logikę UI i wysyłanie żądań do backendu.
- Widoki FXML definiują układ komponentów graficznych.

## 4.2 Backend (Spring Boot)

- Udostępnione REST API w formie kontrolerów (np. `/api/student`, `/api/admin`).
- Logika biznesowa w serwisach.
- Komunikacja z bazą danych przez Spring Data JPA.
- Modele domenowe i DTO do wymiany danych między warstwami.

## 4.3 Baza danych

- Relacyjna baza danych PostgreSQL.
- Tabele powiązane relacjami: uczniowie, nauczyciele, przedmioty, plan lekcji, oceny, obecności.

## 5 Zasady komunikacji

1. Użytkownik (np. nauczyciel) wybiera akcje w interfejsie JavaFX.
2. Kontroler UI wysyła żądanie HTTP (GET, POST, PUT, DELETE) do odpowiedniego endpointu REST.
3. Backend odbiera żądanie, przetwarza je, i zwraca dane w formacie JSON.
4. Dane są przetwarzane i wyświetlane użytkownikowi.