

UNIwersYTET RZESZOWSKI
WYDZIAŁ NAUK ŚCISŁYCH I TECHNICZNYCH
INSTYTUT INFORMATYKI



Oliwier Hędrzak
134913

Informatyka

Aplikacja YLO GradeBook – dokumentacja projektowa

Praca projektowa

Praca wykonana pod kierunkiem
mgr inż. Ewa Żesławska

Rzeszów 2025

Spis treści

1. Wprowadzenie	7
2. Opis założeń projektu	8
2.1. Cel projektu	8
2.2. Sposób realizacji projektu	8
2.3. Wymagania funkcjonalne	9
2.3.1. Logowanie	9
2.3.2. Funkcjonalność dla grupy nauczyciel	9
2.3.3. Funkcjonalność dla grupy uczniów	10
2.3.4. Opis poszczególnych danych	10
2.4. Wymagania niefunkcjonalne	12
2.4.1. Środowisko testowe	12
2.4.2. Wydajność	12
2.4.3. Bezpieczeństwo	13
2.4.4. Użyteczność	13
2.4.5. Skalowalność	14
2.4.6. Integralność danych	15
3. Opis struktury projektu	16
3.1. Wykorzystane technologie	16
3.2. Baza danych	17
3.2.1. Struktura encji i relacje	17
3.2.2. Opis tabel	18
3.2.3. Relacje między tabelami	20
3.3. Struktura aplikacji	21
3.3.1. Struktura pakietów w projekcie	21
3.3.2. Opis zaimplementowanych klas	21
3.3.3. Modele	23
3.3.4. Widoki graficzne (FXML)	24
3.3.5. Pozostałe zasoby	25
4. Harmonogram realizacji projektu	26
4.1. Etapy realizacji oraz wykres Gantta	26
4.2. System kontroli wersji i repozytorium projektu	26
5. Prezentacja warstwy użytkowej projektu	27
5.1. Opis szaty graficznej interfejsu	27
5.2. Prezentacja głównych widoków interfejsu użytkownika	27
5.2.1. Okno logowania	28
5.2.2. Okno resetowania hasła	29

5.2.3. Główne okno interfejsu ucznia	30
5.2.4. Główne okno interfejsu nauczyciela	33
5.2.5. Pozostałe zakładki	35
5.2.6. Widok okna PopUp	37
5.3. Mechanizm alertów systemowych	42
6. Podsumowanie	43
7. Instrukcja uruchomienia plikacji YLO GradeBook	44
7.1. Wymagania środowiskowe	44
7.2. Baza danych	45
7.3. Kroki uruchomienia projektu YLO GradeBook	45
Bibliografia	47
Spis rysunków	48
Oświadczenie studenta o samodzielności pracy	49

1. Wprowadzenie

YLO GradeBook to rozbudowana aplikacja desktopowa realizująca projekt dziennika elektronicznego, stworzona w języku Java z wykorzystaniem biblioteki JavaFX. Do przechowywania danych wykorzystano relacyjną bazę danych MySQL zarządzaną przez środowisko phpMyAdmin. Główny cel aplikacji to zarządzanie ocenami uczniów, jednak została ona wzbogacona również w wiele innych funkcji, takich jak notatki osobiste, zarządzanie terminami, uwagami oraz wyświetlanie statystyk.

Projekt został stworzony z myślą o nowoczesnym i przejrzystym interfejsie graficznym użytkownika oraz odpowiednio zaprojektowanej, logicznej strukturze danych dostosowanej do poszczególnych ról użytkowników. Aplikacja posiada okno logowania oraz możliwość zmiany danych logowania użytkownika.

YLO GradeBook is a developed desktop application implementing an electronic gradebook project, created in Java using the JavaFX library. A MySQL relational database managed via the phpMyAdmin environment was used to store data. The main goal of the application is to manage student grades, but it has also been enhanced with many other features such as personal notes, deadline management, comments, and displaying statistics.

The project was created with a modern and transparent graphical user interface and a properly designed, logical data structure tailored to individual user roles in mind. The application has a login window and the ability to change user login details.

2. Opis założeń projektu

2.1. Cel projektu

Celem projektu jest opracowanie aplikacji desktopowej w języku Java, pełniącej funkcję dziennika elektronicznego. System umożliwi intuicyjne zarządzanie ocenami oraz terminami, a także pozwala na prowadzenie własnych notatek i wyświetlanie statystyk. Po zalogowaniu do systemu ukazuje się przejrzysty i nowoczesny interfejs graficzny użytkownika, zaprojektowany z myślą o estetyce i prostocie. Dla nauczyciela i ucznia przygotowano osobny widok, który przemyślano, w taki sposób, aby funkcje dostosowane do ich roli były łatwo dostępne i komfortowe w obsłudze.

Projekt YLO GradeBook odpowiada na problem braku nowoczesnych, elastycznych i prostych w obsłudze aplikacji desktopowych działających jako dziennik elektroniczny. Dostępne na rynku rozwiązania często przeładowane są zbędnymi funkcjami i nie spełniają wymagań i potrzeb dzisiejszego użytkownika, co utrudnia codzienną pracę.

YLO GradeBook cechuje się przejrzystym interfejsem, który zachęca użytkownika do korzystania z aplikacji. W przeciwieństwie do wielu istniejących rozwiązań posiadających przestarzały interfejs, aplikacja oferuje dokładnie to, co niezbędne. Projekt realizować ma najpotrzebniejsze funkcje dziennika elektronicznego w sposób jasny i przejrzysty dla każdego użytkownika, niezależnie od jego doświadczenia z pracą przy komputerze co umożliwia łatwe korzystanie z aplikacji nawet dla osób, które nie są przekonane do elektronicznego rozwiązania omawianego problemu. Dodawanie ocen czy terminów odbywa się w intuicyjny sposób, aby realizacja zadania była szybka, komfortowa i bezproblemowa. Dodatkowo system został zaprojektowany w taki sposób, aby w przyszłości możliwe było wprowadzenie nowych funkcjonalności zgodnych z potrzebami placówek edukacyjnych.

2.2. Sposób realizacji projektu

Realizacja projektu przebiegać będzie etapowo:

- Analiza wymagań oraz zaprojektowanie najpotrzebniejszej funkcjonalności,
- Zaprojektowanie graficznego interfejsu na podstawie przyjętych założeń,
- Zaprojektowanie struktury bazy danych (MySQL), tak aby wszystkie dane przechowywane były w sposób klarowny i optymalny dla realizowanych funkcji,
- Stworzenie graficznego interfejsu z wykorzystaniem interfejsu JavaFX,
- Implementacja logiki aplikacji i połączenie z bazą danych,
- Testowanie funkcjonalności oraz obsługa ewentualnych błędów,
- Opracowanie dokumentacji projektu zawierającej opis założeń, celu, architektury i sposobu korzystania z aplikacji.

Wynikiem realizacji projektu będzie w pełni działająca i zgodna z założeniami aplikacja desktopowa YLO GradeBook umożliwiająca prowadzenie dziennika elektronicznego na potrzeby szkół. Projekt kładzie nacisk na spersonalizowaną funkcjonalność, czytelność i intuicyjną obsługę dostosowaną do potrzeb zarówno nauczycieli, jak i uczniów.

2.3. Wymagania funkcjonalne

W projekcie zastosowano podział na dwie grupy użytkowników, które posiadają przypisane uprawnienia i funkcjonalności.

2.3.1. Logowanie

- System umożliwia logowanie się do aplikacji za pomocą unikalnej nazwy użytkownika i hasła przechowywanego w bazie danych. Podczas logowania hasło jest ukryte, natomiast użytkownik może je chwilowo ukazać, korzystając z intuicyjnego przycisku do wyświetlenia hasła.
- Po zalogowaniu się system rozpoznaje, do której grupy należy użytkownik i otwiera odpowiedni interfejs graficzny dostosowany do jego roli.
- System od momentu zalogowania przechowuje aktualnie zalogowanego użytkownika, aż do momentu zamknięcia aplikacji.
- Użytkownik może zmienić swoje hasło (w przyszłości funkcja ta zostanie wyposażona w odpowiednie zabezpieczenia).

2.3.2. Funkcjonalność dla grupy nauczyciel

Nauczyciel posiada dostęp do rozszerzonych funkcji administracyjnych:

- Możliwość logowania się do systemu za pomocą unikalnej nazwy użytkownika oraz hasła (na potrzeby realizacji projektu założono, iż konta użytkowników istnieją już w bazie danych)
- Przełączanie widoków aplikacji za pomocą przycisków w panelu nawigacyjnym po lewej stronie interfejsu,
- Wyświetlanie listy uczniów przypisanych do wybranej klasy,
- Przeglądanie ocen uczniów wybranej klasy oraz dla wybranego przedmiotu wraz z obliczeniem średnich,
- Przeglądanie osobistych notatek oraz możliwość ich usuwania,
- Przeglądanie danych osobistych przypisanych do jego konta,
- Możliwość zmiany hasła,
- Dodawanie ocen dla wybranych uczniów konkretnej klasy,
- Dodawania swoich notatek osobistych,
- Dodawanie terminów wydarzeń dla wybranej klasy,
- Dodawania uwag dla wybranego ucznia
- Możliwość wylogowania się z systemu.

2.3.3. Funkcjonalność dla grupy uczniów

Uczeń posiada ograniczoną funkcjonalność w stosunku do nauczyciela. Posiada jednakże możliwości, które zostały zaimplementowane tylko i wyłącznie dla jego roli:

- Możliwość logowania się do systemu za pomocą unikalnej nazwy użytkownika oraz hasła (na potrzeby realizacji projektu założono iż konta użytkowników istnieją już w bazie danych)
- Przełączanie widoków aplikacji za pomocą przycisków w panelu nawigacyjnym po lewej stronie interfejsu,
- Wyświetlanie powiadomień związanych z nowymi ocenami oraz terminami, które zostały dodane w przeciągu ostatnich 7 dni,
- Wyświetlanie ogólnej średniej ucznia,
- Wyświetlanie ocen, które zostały dodane w przeciągu ostatnich 7 dni,
- Przeglądanie wszystkich ocen z podziałem na przedmioty oraz wyliczenie średniej dla każdego z nich,
- Przeglądanie osobistych notatek oraz możliwość ich usuwania,
- Przeglądanie terminów wydarzeń, które zostały przydzielone do klasy, do której należy,
- Przeglądanie uwag, które zostały przypisane przez nauczyciela do jego konta,
- Dodawania swoich notatek osobistych,
- Przeglądanie danych osobistych przypisanych do jego konta,
- Możliwość zmiany hasła,
- Możliwość wylogowania się z systemu.

2.3.4. Opis poszczególnych danych

Dane zostały zaprojektowane, tak aby posiadały swoje cechy, które umożliwiają ich rozróżnianie, sortowanie oraz grupowanie. Dzięki temu aplikacja zachowuje sprawną organizację informacji i realizację funkcji.

2.3.4.1. Oceny

Podczas dodawania ocen przez nauczyciela pierwszym krokiem jest wybranie klasy, a następnie ucznia, któremu przypisana zostanie ocena. Następnie wybierany jest przedmiot z listy, która znajduje się w bazie danych oraz typ, gdzie do wyboru są opcje takie jak: sprawdzian, kartkówka, odpowiedź ustna, zadanie, inne. Ostatnim elementem jest wybranie oceny w systemie 1.0 - 6.0.

Ocena składa się następujących elementów:

- Klasa,
- Uczeń,
- Przedmiot,
- Typ,
- Wartość oceny.

2.3.4.2. Notatki osobiste

Użytkownik niezależnie od swojej roli może dodawać swoje notatki. W pierwszym kroku podaje tytuł notatki, a następnie jej treść. Po jej dodaniu notatka umieszczana jest w bazie danych i zostaje automatycznie przypisana do zalogowanego użytkownika bez konieczności wprowadzania jej właściciela.

Notatka zawiera:

- Tytuł,
- Treść,
- Właściciela (ustalanego automatycznie).

2.3.4.3. Uwagi

Dodając uwagę, nauczyciel wybiera klasę, a następnie ucznia, którego uwaga dotyczy. Następnie wprowadza liczbę punktów (ujemnych lub dodatnich w zależności od tego, czy uwaga jest pozytywna czy negatywna). Ostatnim elementem jest treść uwagi, która ograniczona jest do 60 znaków.

Uwaga zawiera:

- Klasę,
- Ucznia,
- Liczbę punktów (ujemnych lub dodatnich)
- Treści uwagi.

2.3.4.4. Terminy

Nauczyciel, dodając nowy termin, proszony jest o wybór klasy, do której chciałby przypisać nowe wydarzenie. Następnie wybiera przedmiot z listy, która znajduje się w bazie danych, W kolejnym kroku wybiera typ (wydarzenie, sprawdzian, kartkówka, zadanie). Podaje datę wydarzenia, korzystając z pomocniczej kontrolki oraz podaje krótki opis. Uwaga zawiera:

Termin składa się z następujących elementów:

- Klasa,
- Przedmiot,
- Typ wydarzenia,
- Data końcowa,
- Opis.

2.4. Wymagania нефункционалне

2.4.1. Środowisko testowe

Testy projektu przeprowadzone zostały w środowisku o następującej konfiguracji:

- System operacyjny: Windows 11 Pro
- Processor: Intel® Core™ i5-13420H 2.10Ghz
- Pamięć RAM: 16GB DDR4
- Karta graficzna: NVIDIA GeForce RTX 4050 Laptop GPU
- Dysk: SSD 500GB
- Java: openjdk-23 (Oracle OpenJDK 23.0.2)
- Połączenie z bazą danych: mysql-connector-j-9.3.0
- Baza danych: mySQL (środowisko phpMyAdmin – XAMPP Control Panel 3.3.0):
 - Serwer: 127.0.0.1 via TCP/IP
 - Typ serwera: MariaDB
 - Połączenie z serwerem: SSL nie jest używany
 - Wersja serwera: 10.4.32-MariaDB - mariadb.org binary distribution
 - Wersja protokołu: 10
 - Użytkownik: root@localhost
 - Kodowanie znaków serwera: UTF-8 Unicode (utf8mb4)
 - phpMyAdmin - Informacja o wersji: 5.2.1

2.4.2. Wydajność

Aplikacja została stworzona z myślą o komfortowym działaniu i płynnej obsłudze użytkownika. Przeprowadzone testy wykazały następujące cechy:

- Czas uruchamiania aplikacji wynosi poniżej 3 sekund,
- Czas logowania nie przekracza jednej sekundy,
- Interfejs graficzny nie wykazuje żadnych opóźnień związanych z przełączaniem widoków lub przechodzeniem pomiędzy poszczególnymi funkcjami,
- Aplikacja jest stabilna i obsługuje kompletny scenariusz użytkowania nie zależnie od roli bez błędów. Wszystkie funkcje wykonywane są płynnie i komfortowo.
- Wyskakujące okno podczas otwiera się płynnie i nie obciąża komputera.
- Po zalogowaniu aplikacja zużywa około 250 MB pamięci RAM.

System został zoptymalizowany pod kątem efektywnej pracy, a jego struktura pozwala na odpowiednią organizację danych oraz funkcjonalność. Zaprojektowana architektura pozwala na łatwe dodawania nowych funkcjonalności bez naruszania obecnej struktury.

2.4.3. Bezpieczeństwo

System zawiera szereg rozwiązań zapewniających bezpieczeństwo danych oraz użytkownika podczas korzystania z aplikacji:

- Każdy użytkownik posiada swoją unikalną nazwę użytkownika, co pozwala na jednoznaczną jego identyfikację
- Użytkownicy posiadają hasło, które pozwala na logowanie się do systemu
- Istnieje możliwość zmiany hasła w prosty sposób (przewidziana jest aktualizacja wprowadzająca dodatkowe zabezpieczenia podczas tego procesu)
- Pola hasła zostały ukryte; użytkownik posiada jednak możliwość odsłonięcia haseł za pomocą intuicyjnego przycisku.
- Funkcje dostępne dla nauczyciela dostępne są po zalogowaniu się do systemu użytkownika posiadającego rolę nauczyciel.
- Wszelkie procesy (dodawanie ocen, notatek, terminów itp.) kończą się komunikatem informującym o sukcesie lub błędzie wykonania,
- Okna logowanie oraz zmiana hasła została wyposażona w odpowiednie alerty komunikujące przebieg działania. Dzięki temu np. podczas logowania użytkownik musi podać dokładną nazwę użytkownika oraz hasło, a podczas jego resetowania musi wprowadzić hasło dwukrotnie w celu jego potwierdzenia.
- W całej strukturze aplikacji zastosowano obsługę błędów.

Aplikacja stworzona została z myślą o dalszym rozwoju pod kątem zabezpieczeń, a więc nie zamyka się ona na nowe możliwości zabezpieczeń takie jak weryfikacja dwuetapowe lub zmiana hasła za pomocą wysłania odpowiedniej wiadomości na adres e-mail przypisany do konta.

2.4.4. Użyteczność

Aplikacja YLO GradeBook została zaprojektowana z myślą o prostocie i intuicyjności, tak aby mogły z niej swobodnie korzystać osoby o różnym poziomie doświadczenia pracy przy komputerze – zarówno dla nauczycieli, którzy przyzwyczajonych do tradycyjnych papierowych dzienników, jak i uczniów, którzy na co dzień korzystają z różnych aplikacji komputerowych.

- Interfejs graficzny stworzony przy użyciu biblioteki JavaFX oparto na przejrzystym wyglądzie, z podziałem na zakładki umożliwiającymi przechodzenie pomiędzy widokami.
- Panel nawigacyjny znajdujący się po lewej stronie jest klarowny i prosty w obsłudze. Takie rozwiązanie pozwala na lepszą optymalizację, ponieważ nie jest wymagane uruchamianie nowych okien aplikacji.
- Wszystkie kluczowe funkcje są dostępne w jednym oknie. W momencie dodawania danych do bazy pojawia się nowe okno, które nie obciąża systemu.
- Wszystkie procesy zostały zaopatrzone w odpowiednie komunikaty, dzięki czemu użytkownik informowany jest o przebiegu działania, które w danej chwili podejmuje, co sprawia, że użytkownik posiada większą kontrolę i bezpieczeństwo.

- System dostosowany jest do użytkownika w taki sposób, że najczęściej wykonywane działania są łatwo dostępne. Strona główna wyposażona jest w przyciski, które poprawiają automatyzację.

Wszystkie powyższe zabiegi sprawiają, że system jest czytelny dla każdego użytkownika, a nauczanie się go nie sprawia żadnych problemów, a korzystanie z niego to czysta przyjemność. Użyteczność aplikacji stanowi jeden z głównych atutów YLO GradeBook.

2.4.5. Skalowalność

Aplikacja YLO GradeBook została zaprojektowana w sposób umożliwiający łatwą rozbudowę, aby możliwe było szybkie dostosowanie się do potrzeb użytkowników. W związku z tym struktura cechuje się skalowalnością pod względem rozbudowy.

- Struktura bazy danych jest przygotowana pod kątem rozbudowy i łatwej organizacji danych. Możliwe jest dodanie nowych tabel i relacji bez konieczności ingerowania w istniejące. W zależności od wymagań szkół możliwe jest dodanie nowej funkcjonalności jak np. zbieranie informacji o obecnościach czy też konwersacje pomiędzy uczniami i nauczycielami. Struktura bazy danych jest na to gotowa, więc przykładowe aktualizacje nie będą trudne do zaimplementowania.
- Interfejs graficzny posiada taką architekturę, która pozwoli na dodawanie nowych zakładki, co pozwoli na nowe rozwiązania.
- W przyszłości możliwe jest integrowanie aplikacji z zewnętrznymi systemami jak np. połączenie z pocztą elektroniczną.
- Kod został zorganizowany w taki sposób, aby twórcy byli w stanie łatwo odnaleźć interesujący ich fragment i go zmodyfikować lub dodać nowe funkcjonalności.
- Na ten moment aplikacja nie posiada możliwości zmiany rozmiaru okien, jednak jej budowa pozwala na łatwe wprowadzenie tej opcji w przyszłości.

Dzięki tym cechom możliwa jest skalowalność aplikacji i jej dalszy rozwój pod kątem spełniania oczekiwań użytkowników. Zorganizowana architektura jest elastyczna, co sprawia, że kod nie będzie potrzebował gruntownej przebudowy.

2.4.6. Integralność danych

W projekcie YLO GradeBook zachowanie integralności danych stanowi jeden z kluczowych aspektów działania systemu. Wdrożono takie rozwiązania, aby zapewnić spójność danych i zminimalizować prawdopodobieństwo wystąpienia błędów.

- Wszystkie dane, które wprowadzane do bazy danych przechodzą podstawową walidację po stronie aplikacji (np. wymagane pola, limity znaków, typy danych),
- Aplikacja posiada zautomatyzowane dodawanie danych, w taki sposób, aby użytkownik nie musiał ręcznie wpisywać niektórych informacji. Wystarczy, że wybierze on jedną z opcji, co ogranicza możliwość wprowadzania błędnych danych i konieczności walidacji,
- Wartości danych są ograniczone logicznie (np. oceny od 1.0 do 6.0, punkty w przypadku dodawania uwag to liczby całkowite),
- Baza danych wykorzystuje klucze główne i obce, co pozwala na powiązania relacyjne między encjami,
- Operacje zapisu i aktualizacji w bazie danych wykorzystują odpowiednio przemyślane zapytania w języku SQL z parametrami przekazywanymi w aplikacji. Wszystkie metody realizujące zadania tego typu zaopatrzone są w odpowiednią obsługę błędów,
- Dla kluczowych operacji zastosowano system wyświetlających się komunikatów o przebiegu operacji,
- Dzięki odpowiednio przemyślanej strukturze bazy danych możliwe jest odpowiednie pobieranie danych oraz ich wyświetlanie w zależności od spełnionych wymagań

Dzięki tym mechanizmom YLO GradeBook zapewnia wysoki poziom spójności danych, co zapewnia dokładność danych – ma to kluczowe znaczenie w środowisku szkolnym.

3. Opis struktury projektu

Projekt zrealizowany został w języku programowania Java z wykorzystaniem technologii JavaFX oraz MySQL. Głównym celem było stworzenie systemu o czytelnej strukturze i dużej elastyczności rozwojowej – łączącego logikę aplikacji z relacyjną bazą danych oraz graficznym interfejsem użytkownika

Do każdego widoku FXML zaimplementowano osobny kontroler, co pozwoliło zachować przejrzystość oraz zgodność paradygmatem MVC. Do realizacji operacji związanych z bazą danych wykorzystano modele odpowiadające poszczególnym tabelom.

Struktura projektu YLO GradeBook została opracowana zgodnie z zasadami dobrych praktyk programistycznych — w sposób ułatwiający jej dalszy rozwój bez konieczności gruntownej przebudowy kodu źródłowego.

3.1. Wykorzystane technologie

W realizacji projektu zastosowano nowoczesne technologie oraz narzędzia, które wspierają tworzenie aplikacji desktopowych o wysokiej wydajności i estetyce interfejsu użytkownika:

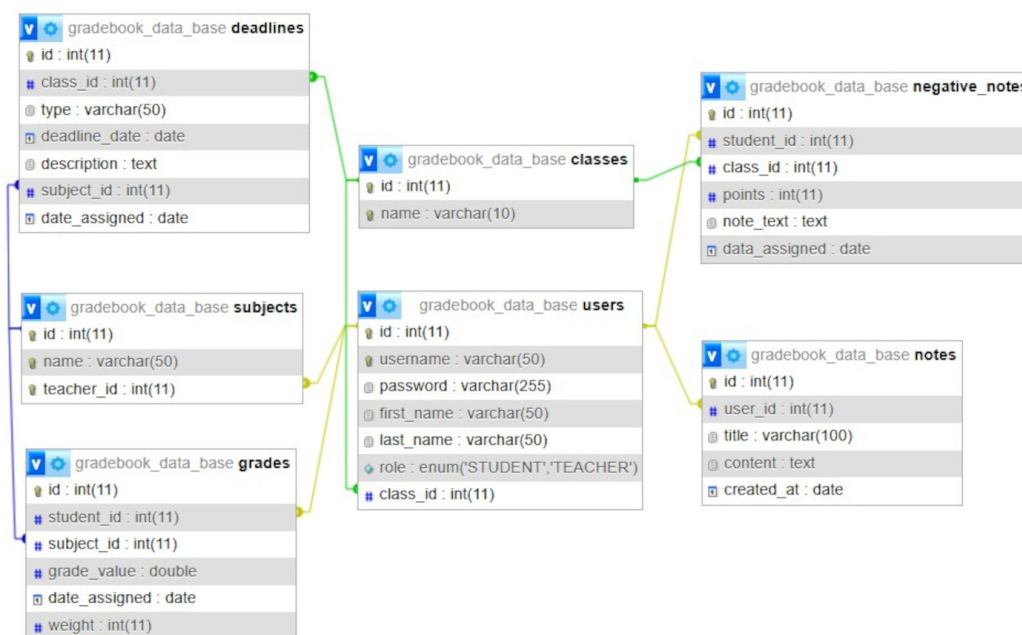
- Java 23 (OpenJDK) – główny język programowania aplikacji, umożliwiający tworzenie wydajnych aplikacji wieloplatformowych,
- JavaFX – biblioteka graficzna umożliwiająca projektowanie interfejsów użytkownika z wykorzystaniem stylizacji i animacji,
- FXML – deklaratywny język XML do definiowania struktury widoków graficznych, ułatwiający separację warstwy GUI od logiki aplikacji,
- CSS – kaskadowe arkusze stylów, użyte do dostosowania wyglądu interfejsu oraz zapewnienia spójnej estetyki aplikacji,
- MySQL + phpMyAdmin – relacyjny system zarządzania bazą danych oraz narzędzie webowe do jej wygodnej administracji,
- MySQL Connector/J – sterownik JDBC umożliwiający komunikację aplikacji Java z bazą danych MySQL,
- IntelliJ IDEA Community Edition 2024.3.3 – środowisko programistyczne wykorzystywane do implementacji kodu źródłowego,
- Scene Builder – aplikacja wspomagająca wizualne projektowanie interfejsów w FXML, znacznie przyspieszająca pracę nad GUI.

3.2. Baza danych

W celu realizacji projektu *YLO GradeBook* zaprojektowano relacyjną bazę danych o nazwie *gradebook_data_base*, stworzoną w systemie MySQL i zarządzaną za pomocą środowiska phpMyAdmin. Jej zadaniem jest przechowywanie danych dotyczących użytkowników, ocen, przedmiotów, terminów, notatek oraz uwag. Struktura bazy została zaprojektowana z uwzględnieniem zasad integralności danych oraz możliwości szybkiego filtrowania i przeszukiwania informacji poprzez odpowiednie relacje pomiędzy encjami. Model został stworzony w sposób umożliwiający łatwą rozbudowę o dodatkowe moduły i funkcjonalności w przyszłości.

3.2.1. Struktura encji i relacje

Baza danych została podzielona na tabele reprezentujące użytkowników, dla roli uczniów powstała tabela zawierająca klasy, do których należą. Utworzono tabele przechowujące dane takie jak terminy, notatki osobiste oraz uwagi, a także tabela z listą przedmiotów szkolnych. Relacje między encjami opierają się na kluczach obcych i pozwalają zachować spójność logiczną oraz integralność danych.



Rys. 3.1. Diagram ERD bazy danych *gradebook_data_base*

3.2.2. Opis tabel

Struktura bazy danych została zaprojektowana w sposób modularny — każda tabela pełni ściśle określoną funkcję w systemie YLO GradeBook. Tabele powiązane są ze sobą relacjami typu *jeden do wielu* lub *wiele do wielu*, co zapewnia przejrzystość, skalowalność i integralność.

Poniżej przedstawiono opis wszystkich tabel:

Tabela **users**

Tabela zawiera informacje o wszystkich użytkownikach systemu (uczniach oraz nauczycielach).

- `id` – unikalny identyfikator użytkownika (klucz główny),
- `username, password` – dane logowania,
- `first_name, last_name` – dane osobowe,
- `role` – rola użytkownika (np. „student”, „teacher”),
- `class_id` – odwołanie do tabeli `classes` (klucz obcy).

Tabela **grades**

Tabela przechowuje oceny przypisywane uczniom.

- `id` – identyfikator oceny (klucz główny),
- `student_id` – identyfikator ucznia (klucz obcy do `users`),
- `subject_id` – przedmiot (klucz obcy do `subjects`),
- `grade_value` – wartość oceny (np. 4.5),
- `type` – typ oceny (np. sprawdzian, zadanie),
- `created_at` – data wystawienia.

Tabela **notes**

Tabela przechowuje notatki osobiste dodawane przez użytkowników systemu.

- `id` – identyfikator notatki (klucz główny),
- `user_id` – użytkownik, do którego przypisana jest notatka (klucz obcy do `users`),
- `title` – tytuł notatki,
- `content` – treść notatki,
- `created_at` – data utworzenia.

Tabela deadlines

Tabela przechowuje informacje o nadchodzących terminach.

- `id` – identyfikator terminu (klucz główny),
- `class_id` – klasa, której dotyczy wydarzenie,
- `subject_id` – przedmiot związany z terminem,
- `type` – typ wydarzenia (np. sprawdzian, zadanie),
- `date` – data wydarzenia,
- `description` – krótki opis.

Tabela negative_notes

Tabela służy do zapisywania uwag pozytywnych i negatywnych przypisanych uczniom.

- `id` – identyfikator uwagi (klucz główny),
- `student_id` – uczeń, którego dotyczy uwaga,
- `class_id` – klasa ucznia,
- `points` – liczba punktów (ujemne/dodatnie),
- `message` – treść uwagi (do 60 znaków),
- `created_at` – data wystawienia.

Tabela subjects

Tabela zawiera przedmioty występujące w systemie.

- `id` – identyfikator przedmiotu (klucz główny),
- `name` – nazwa przedmiotu (np. „matematyka”),
- `teacher_id` – nauczyciel prowadzący (klucz obcy do `users`).

Tabela classes

Lista klas funkcjonujących w systemie.

- `id` – identyfikator klasy (klucz główny),
- `name` – nazwa klasy (np. „1TIA”),
- `school_year` – rok szkolny przypisany do klasy.

3.2.3. Relacje między tabelami

Relacje pomiędzy tabelami w bazie *gradebook_data_base* zostały zaprojektowane w sposób zapewniający pełną integralność danych oraz umożliwiający logiczne powiązanie informacji pomiędzy różnymi obiektami systemu.

- Każdy użytkownik (*users*) przypisany jest do konkretnej klasy (*classes*), co pozwala na filtrowanie danych względem przynależności uczniów.
- Użytkownicy mogą pełnić rolę ucznia lub nauczyciela, dzięki czemu możliwe jest rozdzielenie funkcjonalności dla obu grup.
- Oceny (*grades*) są powiązane zarówno z użytkownikiem (ucznem), jak i przedmiotem (*subjects*), którego dotyczą.
- Terminy (*deadlines*) są przypisane do konkretnej klasy i odnoszą się do określonego przedmiotu.
- Notatki (*notes*) tworzone są indywidualnie przez użytkowników i przypisane tylko do ich kont.
- Uwagi (*negative_notes*) są przypisane do ucznia oraz zawierają liczbę punktów (ujemnych lub dodatnich) i treść.

Podsumowanie

Struktura bazy danych *gradebook_data_base* została zaprojektowana z myślą o przejrzystości, integralności oraz skalowalności danych. Dzięki odpowiednio powiązanym encjom, system pozwala na jednoznaczную identyfikację użytkowników, przypisywanie ocen, notatek, terminów i uwag w kontekście klas i przedmiotów. Zaprojektowana architektura relacyjna nie tylko wspiera bieżącą funkcjonalność aplikacji, ale umożliwia także łatwą rozbudowę systemu w przyszłości — zarówno pod względem danych, jak i logiki aplikacyjnej.

3.3. Struktura aplikacji

Aplikacja *YLO GradeBook* została stworzona w języku **Java** z wykorzystaniem biblioteki **JavaFX**. Struktura projektu podzielona została na warstwy odpowiadające za logikę, prezentację danych, komunikację z bazą danych oraz warstwę wizualną.

3.3.1. Struktura pakietów w projekcie

Aplikacja *YLO GradeBook* posiada podział na kilka głównych części:

- pakiet `główny` – zawiera wszystkie klasy aplikacji odpowiedzialne za:
 - logikę kontrolerów powiązanych z plikami `.fxml`,
 - klasę startową `Main.java`,
 - połączenie z bazą danych (`DataBaseConnection.java`),
 - klasy pomocnicze
- `models` – pakiet zawierający klasy odwzorowujące encje z bazy danych. Każdy model reprezentuje jedną tabelę.
- `resources` – pakiet zawierający pliki wykorzystywane w warstwie prezentacji, takie jak:
 - pliki `FXML` opisujące strukturę widoków graficznych,
 - style `CSS` odpowiadające za wygląd interfejsu,
 - czcionki i ikony wykorzystywane w interfejsie użytkownika.

Chociaż klasy w pakiecie głównym nie zostały podzielone na pod-pakiety ze względu na funkcjonalność, zastosowano odpowiednie nazewnictwo, które jasno określa ich przeznaczenie. Dzięki temu struktura pozostaje spójna i zrozumiała.

3.3.2. Opis zaimplementowanych klas

Diagram klas (rys. 3.2) przedstawia strukturę głównych komponentów systemu, ich wzajemne relacje dziedziczenia oraz implementację.

Main.java

Klasa startowa aplikacji. Zawiera metodę `start()` z `JavaFX` i ładuje widok logowania (`LoginWindow.fxml`). Odpowiada za inicjalizację sceny głównej i ustawienie tytułu aplikacji.

DataBaseConnection.java

Klasa służy do połączenia z bazą danych `MySQL`. Wykorzystuje sterownik `JDBC` oraz dane logowania zapisane w kodzie.

MainWindow.java

Klasa pełniący zadanie kontrolera dla głównego kontenera (`MainWindow.fxml`) do ładowania poszczególnych widoków.

ViewLoadingManager.java

Klasa pomocnicza pełniący funkcję menadżera ładowania widoków. Każda metoda przyjmuje nazwę pliku `.fxml` i zastępuje bieżący widok w kontenerze `MainWindow.fxml`.

Session.java

Klasa pomocnicza przechowująca informacje o zalogowanym użytkowniku, dzięki czemu w trakcie trwania sesji aplikacji mamy dostęp do jego danych.

SessionController.java

Abstrakcyjna klasa bazowa zawierająca metody do wyświetlania komunikatów systemowych (błędy, ostrzeżenia, informacje). Umożliwia klasom pochodnym stosowanie jednolitego mechanizmu komunikacji z użytkownikiem, eliminując powielanie kodu.

AuthenticationInterface.java

W strukturze skorzystano z możliwości implementacji interfejsu, który wymusza na klasach go implementujących zdefiniowanie odpowiednich metod. Metody, które implementuje umożliwiają pokazywanie hasła podczas logowania lub resetowania hasła oraz pozwalające na włączenie możliwości przemieszczanie się tabulatorem po interfejsie użytkownika.

LoginWindow.java

Kontroler powiązany z widokiem `LoginWindow.fxml`. Odpowiada za logowanie użytkownika i zapis jego danych w klasie `Session`. Implementuje interfejs `AuthenticationInterface`, co umożliwia m.in. obsługę pokazywania hasła oraz przełączanie fokusu tabulatorem. Udostępnia również metodę otwierającą widok resetowania hasła.

PasswordReset.java

Kontroler przypisany do widoku `PasswordReset.fxml`. Pozwala na aktualizację hasła użytkownika po podaniu poprawnych danych. Również implementuje interfejs `AuthenticationInterface` i definiuje odpowiednie metody zgodne z tym widokiem.

StudentWindow.java

Kontroler odpowiadający za widok `StudentWindow.fxml`. Odpowiada za obsługę graficznego interfejsu ucznia, który zawiera panele z ocenami, terminami, uwagami oraz notatkami osobistymi. Klasa odpowiada m.in. za:

- ładowanie i aktualizowanie danych ucznia (oceny, średnia, uwagi, terminy),
- filtrowanie wpisów dodanych w ciągu ostatnich 7 dni,
- obsługę notatek użytkownika (dodawanie i usuwanie),
- przełączanie widoków i obsługę przycisków w panelu bocznym,
- dostęp do widoków danych osobowych oraz zmiany hasła.

TeacherWindow.java

Kontroler widoku `TeacherWindow.fxml`, pełniący funkcję głównego panelu nauczyciela. Klasa umożliwia:

- przegląd uczniów przypisanych do wybranej klasy,
- przegląd i analizę ocen z podziałem na przedmioty i uczniów,
- dodawanie ocen, terminów, uwag oraz notatek osobistych,
- zarządzanie widokiem danych osobowych i zmiany hasła,
- przełączanie sekcji aplikacji za pomocą panelu nawigacyjnego.

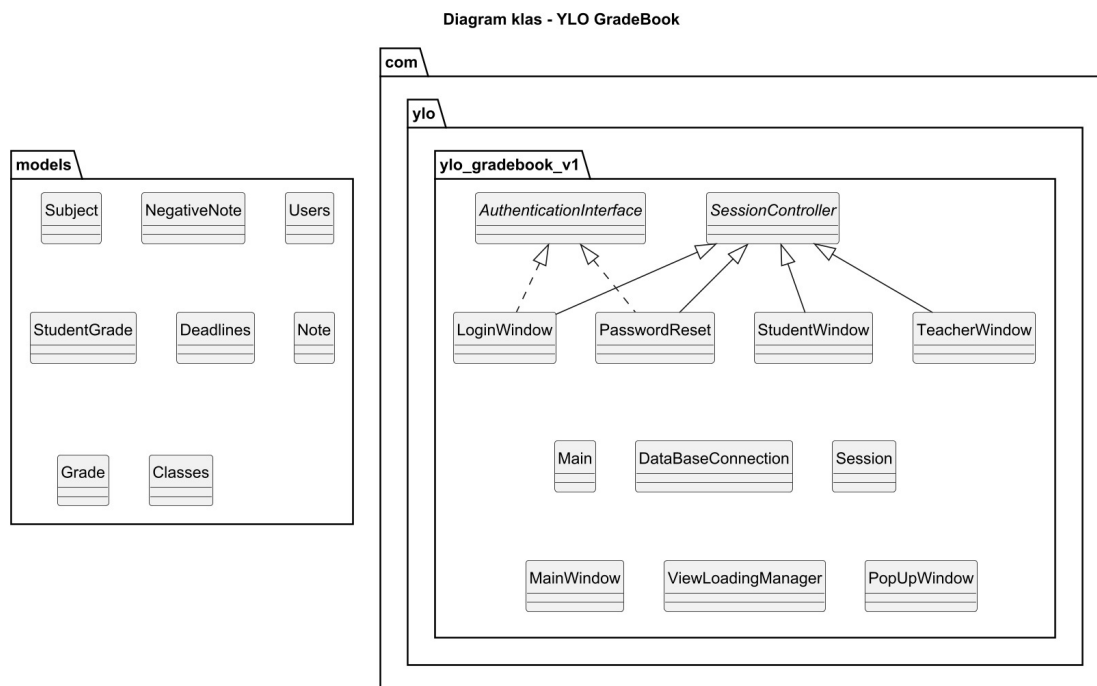
PopUpWindow.java

Klasa pomocnicza obsługująca otwieranie widoków typu *popup* w osobnych oknach. Umożliwia ładowanie plików `.fxml` zawierających formularze, np. do dodawania ocen, terminów, notatek czy uwag. Zapewnia spójny i modularny sposób wyświetlania dodatkowych interfejsów użytkownika bez zakłócania głównego widoku.

3.3.3. Modele

W strukturze systemu zastosowano modele odpowiadające tabelom bazy danych, co umożliwia odwzorowanie danych w postaci obiektów Java. Każda klasa modelowa zawiera zestaw atrybutów reprezentujących kolumny w tabeli oraz odpowiednie metody dostępne (getter i setter).

- `Classes.java` – reprezentuje klasę uczniowską; zawiera m.in. nazwę klasy.
- `Deadlines.java` – odwzorowuje terminy wydarzeń przypisanych do klas i przedmiotów, takich jak sprawdziany czy kartkówki.
- `Grade.java` – odpowiada za pojedynczą ocenę ucznia.
- `NegativeNote.java` – przechowuje informacje o punktowych uwagach przypisanych do uczniów, zarówno pozytywnych, jak i negatywnych.
- `Note.java` – model notatki utworzonej przez użytkownika, zawierający tytuł i treść.
- `StudentGrade.java` – pomocniczy model wykorzystywany do wyświetlania ocen z przypisanym przedmiotem i obliczoną średnią ocen.
- `Subject.java` – reprezentuje przedmiot szkolny.
- `Users.java` – ogólny model użytkownika systemu (ucznia lub nauczyciela), przechowujący dane logowania, imię, nazwisko i rolę.



Rys. 3.2. Diagram klas systemu YLO GradeBook

3.3.4. Widoki graficzne (FXML)

Interfejs użytkownika aplikacji YLO GradeBook został zaprojektowany z użyciem technologii **FXML**, co umożliwia oddzielenie warstwy wizualnej od logiki aplikacji. Każdy plik FXML odpowiada konkretnemu widokowi i powiązany jest ze swoim kontrolerem w języku Java.

- `MainWindow.fxml` – główny kontener aplikacji, do którego ładowane są pozostałe widoki (ucznia lub nauczyciela). Stanowi miejsce dynamicznie zmieniających się widoków.
- `LoginWindow.fxml` – widok logowania. Zawiera pola do wpisania loginu i hasła, przycisk logowania oraz odnośnik do resetowania hasła. Umożliwia użytkownikowi dostęp do systemu.
- `PasswordReset.fxml` – formularz zmiany hasła. Zawiera pola do wprowadzenia loginu, nowego hasła i jego potwierdzenia.
- `PopUpWindow.fxml` – szablon wykorzystywany do wyświetlania popupów, np. do dodawania ocen, terminów, notatek czy uwag. Umożliwia ładowanie okien pomocniczych bez zakłócania pracy głównej aplikacji.
- `StudentWindow.fxml` – główny widok ucznia. Zawiera pasek nawigacyjny, umożliwiający przełączanie się pomiędzy wieloma zakładkami. Zawiera rozbudowaną strukturę, która realizuje wszystkie funkcje udostępnione dla roli ucznia.
- `TeacherWindow.fxml` – główny widok nauczyciela. Udostępnia pełną funkcjonalność do zarządzania danymi. Podobnie jak widok ucznia posiada panel nawigacyjny do przełączania się między zakładkami.

3.3.5. Pozostałe zasoby

Folder `resources` zawiera nie tylko pliki FXML odpowiadające widokom graficznym, ale także zasoby wizualne, które wspierają spójny i nowoczesny wygląd aplikacji. W jego strukturze znajdują się:

- `styles/` – pliki CSS definiujące styl graficzny aplikacji. Wpływają one na kolory, marginesy, zaokrąglenia przycisków i ogólną estetykę interfejsu użytkownika.
- `icons/` – zestaw ikon w formacie `.png`, wykorzystywany w przyciskach, paskach nawigacyjnych i oknach popup. Ułatwiają szybką identyfikację funkcji.
- `fonts/` – pliki czcionek używane w interfejsie. Ich dołączenie zapewnia spójność typograficzną.

Zestaw ikon użytych w interfejsie graficznym pochodzi ze strony `www.freepik.com`. Ikony zostały pobrane w ramach licencji wymagającej przypisania autorstwa i były modyfikowane graficznie na potrzeby projektu.

Zastosowanie zasobów oddzielonych od logiki aplikacji zwiększa przejrzystość projektu i ułatwia jego stylowanie bez ingerencji w kod źródłowy.

4. Harmonogram realizacji projektu

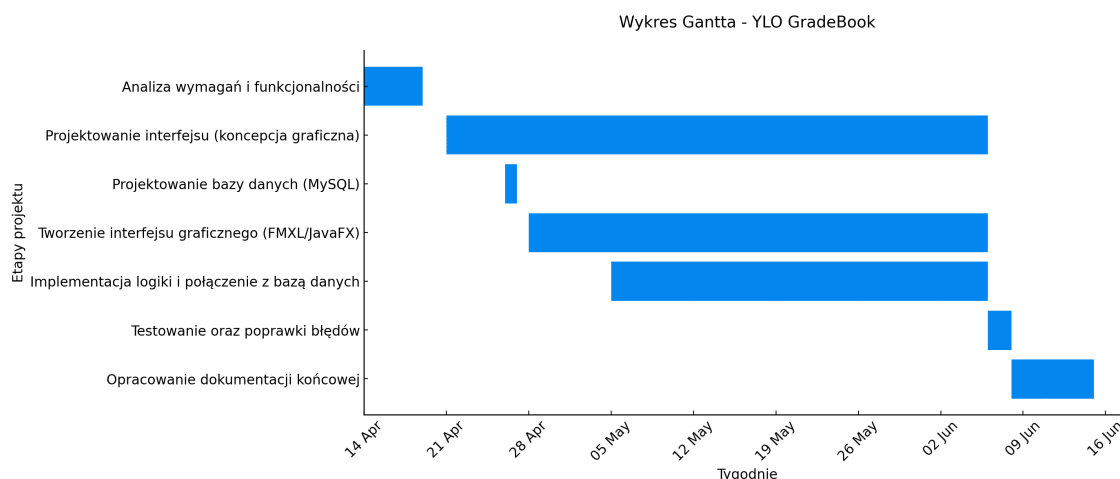
4.1. Etapy realizacji oraz wykres Gantta

Proces tworzenia aplikacji YLO GradeBook został zaplanowany jako ciąg kolejnych etapów, obejmujących m.in. analizę wymagań, projektowanie architektury, tworzenie interfejsu użytkownika, implementację logiki systemu, testowanie oraz przygotowanie dokumentacji technicznej.

Realizacja przebiegała zgodnie z przyjętym harmonogramem, jednak w trakcie realizacji konieczne było kilkukrotne cofnięcie się do wcześniejszych faz projektu — w celu wprowadzenia poprawek lub dostosowania struktury do nowych funkcjonalności.

Warto podkreślić, że wiele zadań prowadzono równolegle, ponieważ dawało to swobodę realizacji nowych pomysłów.

W celu zobrazowania kolejnych faz projektu, opracowano wykres Gantta (rys. 4.1) przedstawiający plan pracy.



Rys. 4.1. Wykres Gantta przedstawiający harmonogram realizacji projektu YLO GradeBook

4.2. System kontroli wersji i repozytorium projektu

W trakcie realizacji projektu wykorzystano system kontroli wersji **Git**, który umożliwił systematyczne zapisywanie kodu. Do przechowywania i synchronizacji kodu źródłowego użyto platformy **GitHub**.

Pełne repozytorium projektu YLO GradeBook jest dostępne pod adresem:

<https://github.com/oleiy/YLO-GradeBook>

Repozytorium zawiera ostateczną wersję projektu, okumentację projektowa oraz bazę danych. Zgodnie z wymaganiami, projekt dostępny będzie przez cały rok.

5. Prezentacja warstwy użytkowej projektu

Warstwa użytkowa aplikacji YLO GradeBook została opracowana z myślą o intuicyjności, estetyce oraz ergonomii użytkownika. Interfejs powstał przy użyciu technologii **JavaFX**, a do jego definiowania wykorzystano pliki **FXML**, co umożliwiło czytelne oddzielenie logiki od prezentacji. W projekcie zastosowano również kaskadowe arkusze stylów CSS w celu zdefiniowania wyglądu widoków.

W tym rozdziale zaprezentowano strukturę graficzną głównych widoków aplikacji wraz z opisem funkcji dostępnych dla poszczególnych ról użytkownika (uczeń/nauczyciel). Dodatkowo omówiono styl graficzny interfejsu (kolorystyka, czcionki, ikony) oraz zastosowane mechanizmy wspomagające interakcję z systemem (komunikaty, formularze, walidacja danych).

5.1. Opis szaty graficznej interfejsu

Zastosowana kolorystyka interfejsu opiera się na stonowanej palecie odcieni niebieskiego, bieli i jasnej szarości, co sprzyja przejrzystości i komfortowi wizualnemu użytkownika. Dominujące barwy zostały dobrane w taki sposób, aby nie rozpraszać, a jednocześnie nadawać aplikacji nowoczesny i estetyczny wygląd.

- **rgb(3,134,238)** — intensywny niebieski wykorzystywany do przycisków.
- **rgb(220,220,220)** — jasnoszare tło głównego widoku, stanowiące neutralne tło dla treści,
- **rgb(255,255,255)** oraz **rgb(245,245,245)** — używane dla zwiększenia czytelności sekcji,
- **rgb(238,248,255)** — wyróżnienie niektórych elementów np. główny przycisk w sekcji.
- **rgb(220,220,220)** — kolor ramek różnych elementów interfejsu.

Taka paleta kolorów nie tylko spełnia funkcje estetyczne, ale również wpływa na ergonomię oraz odczucia użytkownika w trakcie korzystania z aplikacji.

Dodatkowym elementem nadającym interfejsowi nowoczesny charakter jest zastosowanie czcionki **Poppins** — geometrycznego, bezszeryfowego kroju pisma, który zapewnia doskonałą czytelność i estetykę.

Ponadto w całej aplikacji wykorzystano delikatne zaokrąglenia rogów przycisków, kart i paneli, co — w połączeniu z jasną kolorystyką i minimalistycznym układem — nadaje całości subtelnie *futurystyczny wygląd*, przy jednoczesnym zachowaniu prostoty oraz przejrzystości układu graficznego.

5.2. Prezentacja głównych widoków interfejsu użytkownika

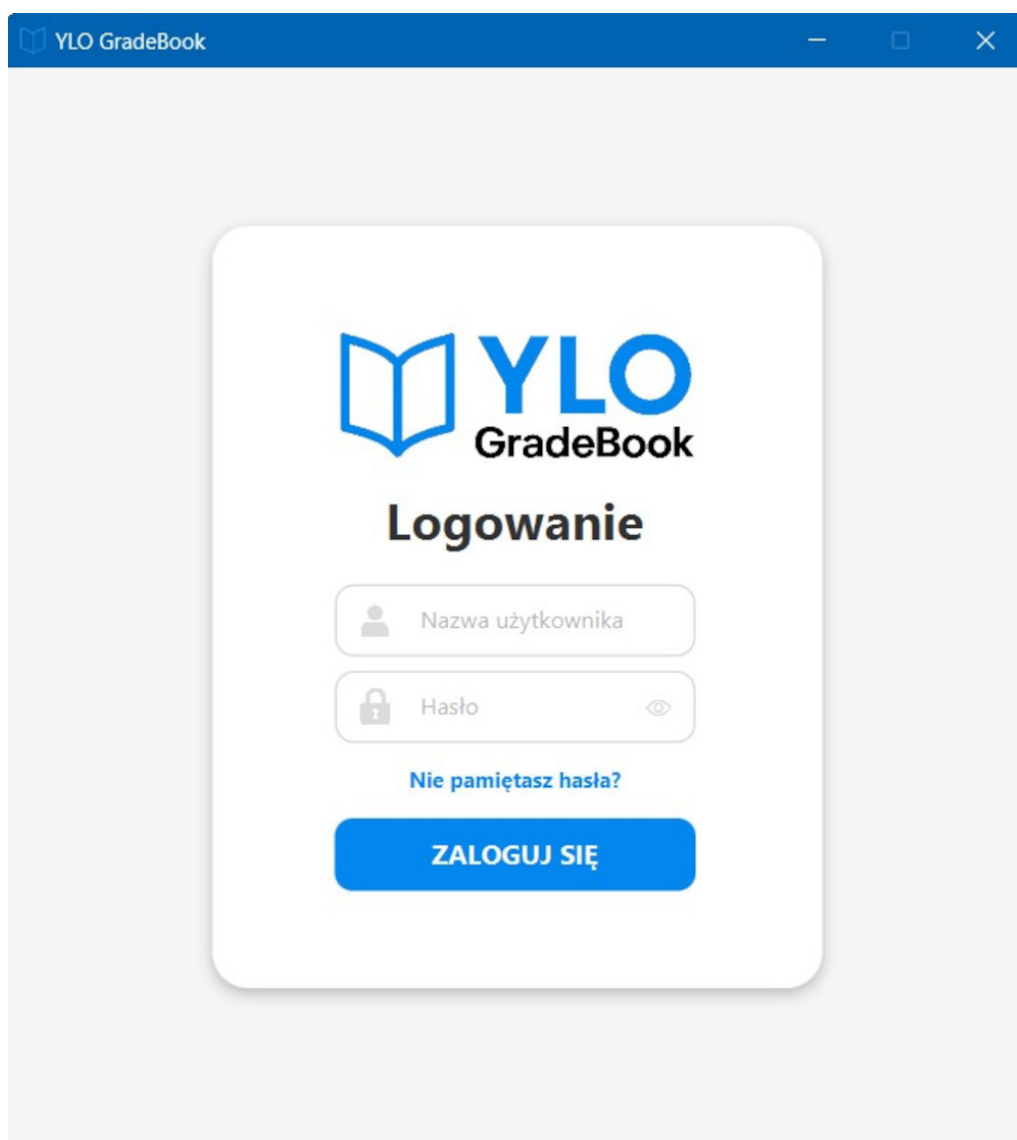
W tej sekcji przedstawiono główne widoki interfejsu użytkownika aplikacji YLO GradeBook wraz z opisem ich funkcjonalności i przeznaczenia. Każdemu ekranowi towarzyszy zrzut ekranu ukazujący jego wygląd oraz interakcje z użytkownikiem.

Przedstawione zrzuty ekranu pochodzą z sesji użytkowników: ucznia o nazwie Igor Lis oraz nauczyciela Jan Nowak. Wykorzystano je w celu zaprezentowania widoków odpowiadających poszczególnym rolom w systemie.

5.2.1. Okno logowania

Widok logowania stanowi punkt wejścia do aplikacji i umożliwia użytkownikowi dostęp do systemu poprzez weryfikację jego danych uwierzytelniających.

- Interfejs zawiera dwa pola wejściowe: **Nazwa użytkownika** oraz **Hasło**, przycisk **Zaloguj się**, a także opcję **Nie pamiętasz hasła?**, umożliwiającą resetowanie hasła.
- Posiada również możliwość pokazania hasła, za pomocą intuicyjnego przycisku na polu hasła.
- W przypadku błędnego logowania pojawia się odpowiedni komunikat walidacyjny.

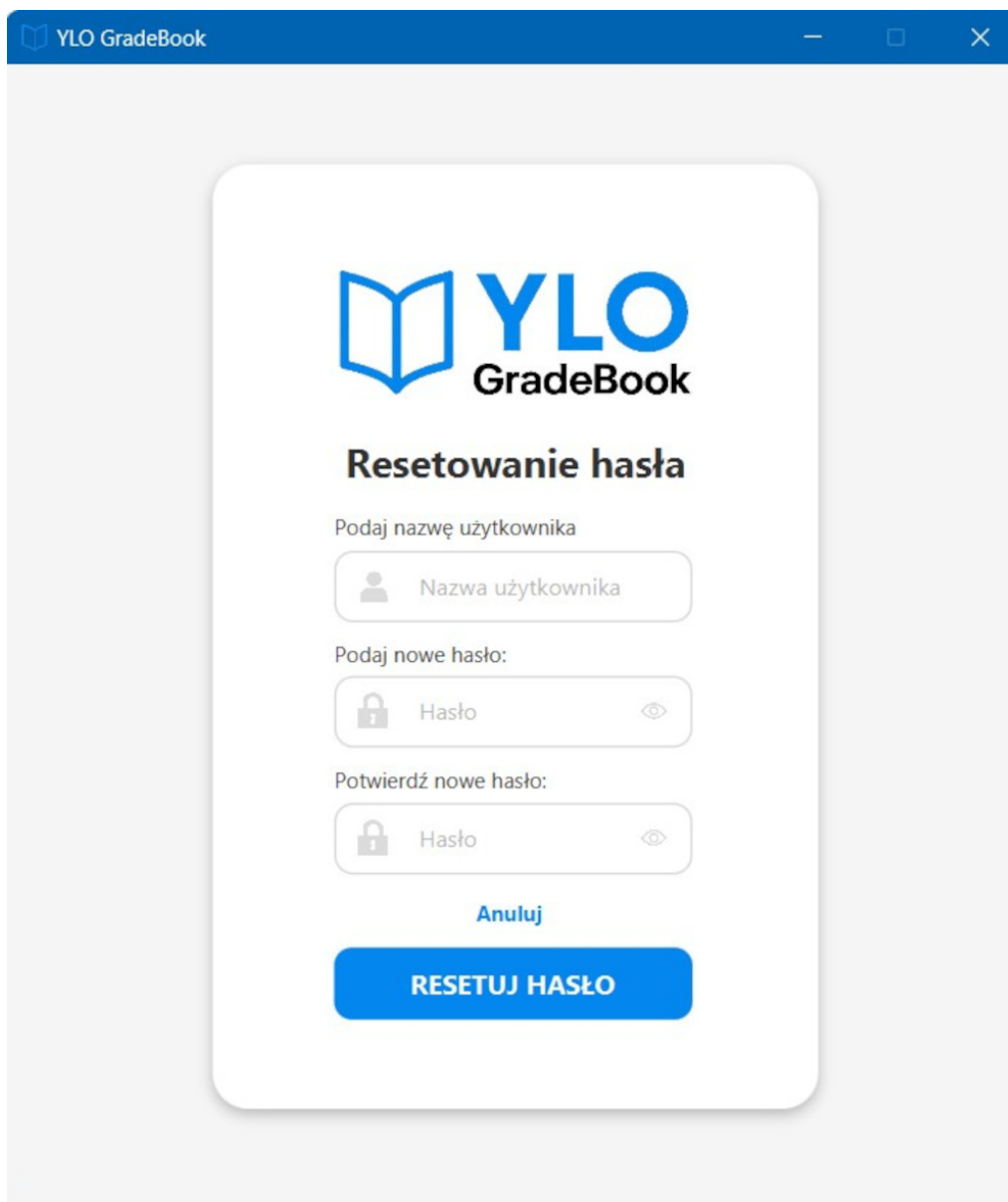


Rys. 5.1. Okno logowania - weryfikacja danych uwierzytelniających

5.2.2. Okno resetowania hasła

Widok ten umożliwia zresetowanie hasła użytkownika poprzez podanie jego nazwy konta oraz nowego hasła. Na tym etapie proces nie zawiera jeszcze pełnych mechanizmów bezpieczeństwa (np. weryfikacji mailowej), jednak przewidziano ich wdrożenie w przyszłych wersjach aplikacji.

- Interfejs posiada trzy pola: **Nazwa użytkownika**, **Hasło** oraz **Potwierdź hasło**. Posiada również przycisk **Anuluj** oraz przycisk **Resetuj hasło**, który pozwala na zaaktualizowanie hasła w bazie danych.
- Podobnie jak w przypadku okna logowania dodano możliwość wyświetlenia haseł.
- W przypadku błędnego wypełnienia pól pojawia się komunikat.



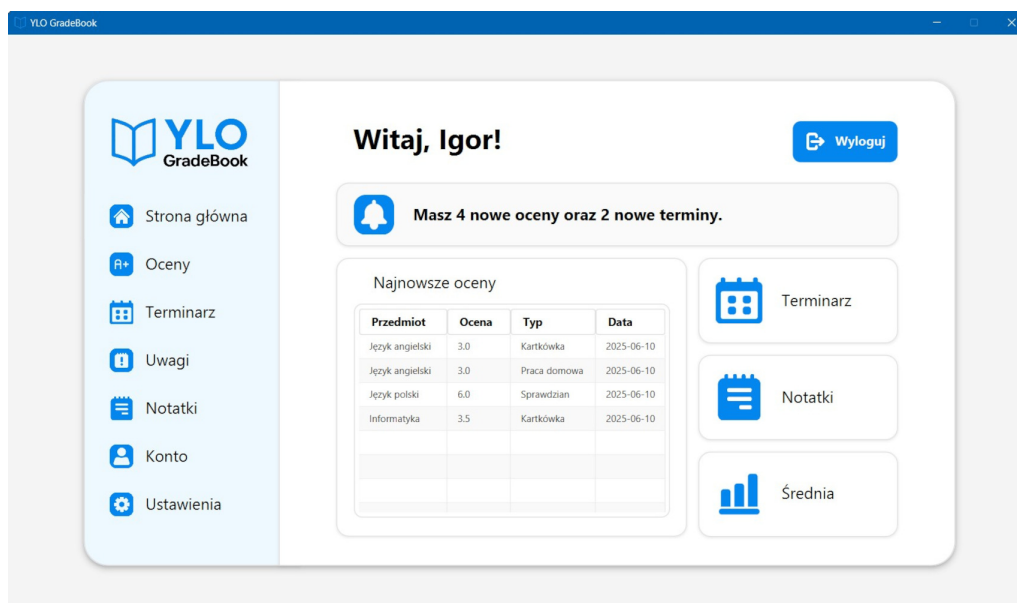
Rys. 5.2. Okno resetowania hasła - aktualizacja hasła w bazie danych

5.2.3. Główne okno interfejsu ucznia

Główne okno interfejsu ucznia pojawia się bezpośrednio po jego zalogowaniu do systemu YLO GradeBook. Zostało zaprojektowane z myślą o szybkim i intuicyjnym dostępie do najważniejszych informacji oraz funkcji systemu. Podział treści na zakładki oraz wyraźna struktura ułatwiają poruszanie się po aplikacji.

- Po lewej stronie znajduje się panel nawigacyjny z zakładkami: **Strona główna**, **Oceny**, **Terminarz**, **Uwagi**, **Notatki**, **Konto** oraz **Ustawienia**.
- W górnej części widoku wyświetlane jest powitanie użytkownika wraz z jego imieniem, a także powiadomienie z liczbą nowych ocen i nadchodzących terminów. Po prawej stronie znajduje się przycisk **Wyloguj**, umożliwiający zakończenie sesji.
- Centralną część okna zajmuje sekcja **Najnowsze oceny**, prezentująca ostatnio dodane wyniki ucznia.
- Po prawej stronie umieszczono szybkie odnośniki do najczęściej wykorzystywanych zakładek, co ułatwia nawigację. W tym miejscu znajduje się również panel ze średnią, który po kliknięciu ukazuje ogólną średnią ucznia.

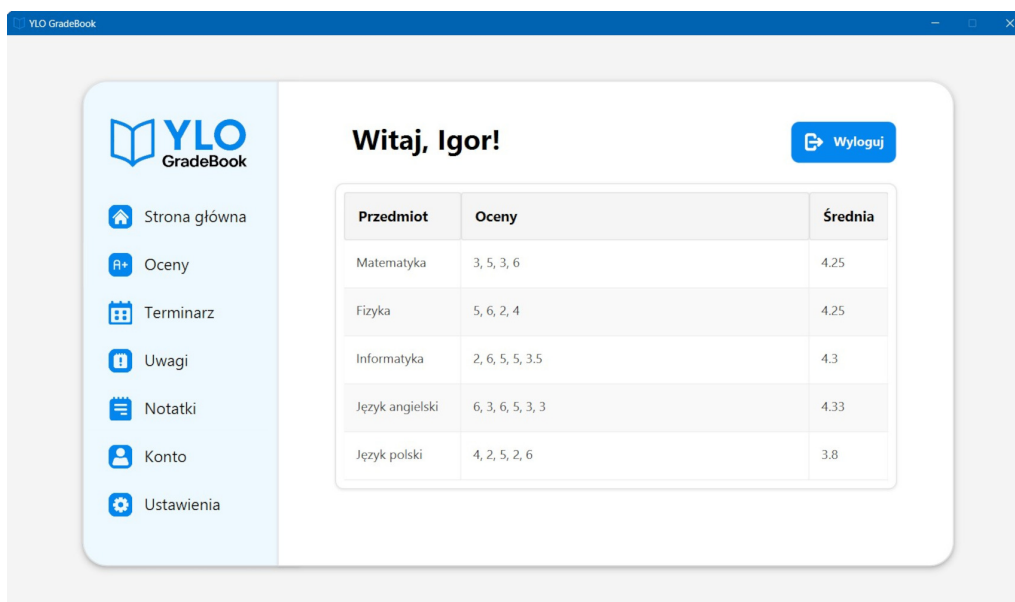
Taki układ pozwala użytkownikowi w intuicyjny sposób monitorować postępy oraz mieć wgląd w najnowsze wydarzenia.



Rys. 5.3. Główny widok ucznia

5.2.3.1. Zakładka „Oceny”

Widok ten umożliwia uczniowi przegląd ocen z podziałem na przedmioty oraz wyświetlenie średniej ocen z każdego z nich.



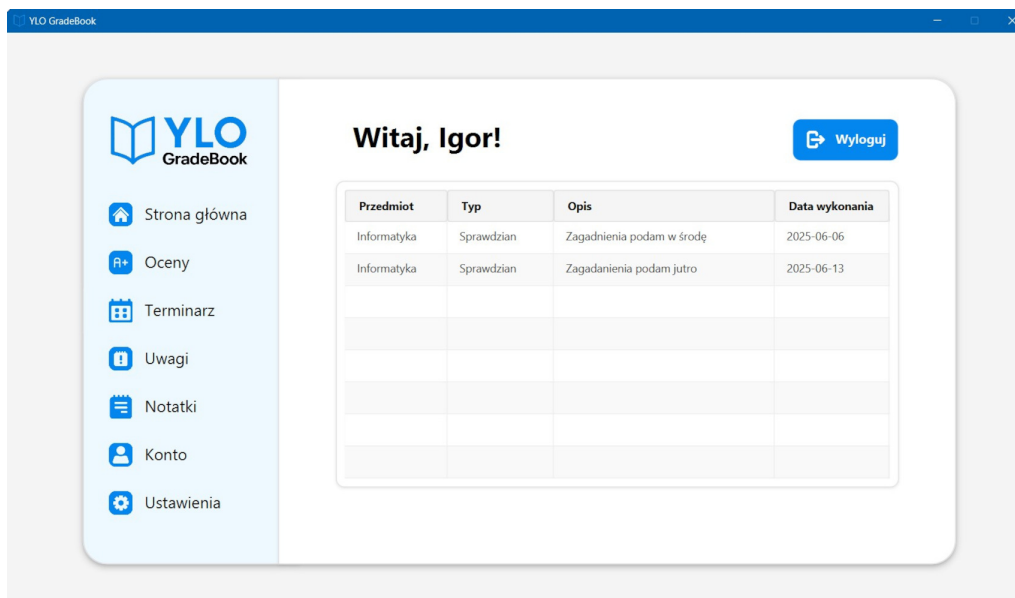
The screenshot shows the 'Oceny' tab in the YLO GradeBook application. The sidebar on the left contains the following menu items: Strona główna, Oceny, Terminarz, Uwagi, Notatki, Konto, and Ustawienia. The main content area displays a welcome message 'Witaj, Igor!' and a table of grades.

Przedmiot	Oceny	Średnia
Matematyka	3, 5, 3, 6	4.25
Fizyka	5, 6, 2, 4	4.25
Informatyka	2, 6, 5, 5, 3.5	4.3
Język angielski	6, 3, 6, 5, 3, 3	4.33
Język polski	4, 2, 5, 2, 6	3.8

Rys. 5.4. Okno ucznia - Zakładka „Oceny”

5.2.3.2. Zakładka „Terminarz”

Zakładka służy do wyświetlania wszystkich nadchodzących terminów wydarzeń przypisanych do klasy ucznia.



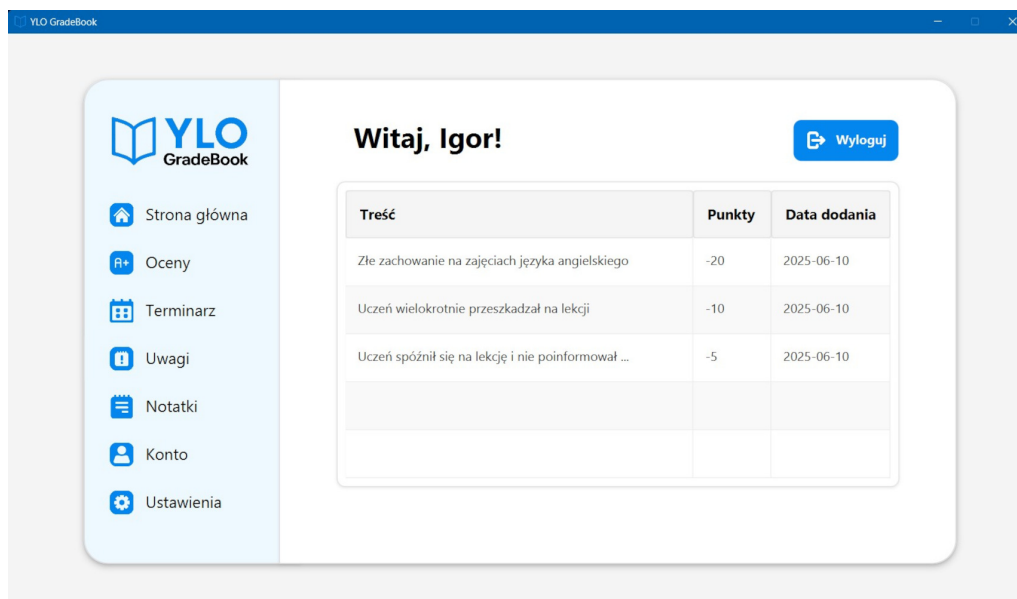
The screenshot shows the 'Terminarz' tab in the YLO GradeBook application. The sidebar on the left contains the following menu items: Strona główna, Oceny, Terminarz, Uwagi, Notatki, Konto, and Ustawienia. The main content area displays a welcome message 'Witaj, Igor!' and a table of upcoming events.

Przedmiot	Typ	Opis	Data wykonania
Informatyka	Sprawdzian	Zagadnienia podam w środę	2025-06-06
Informatyka	Sprawdzian	Zagadnienia podam jutro	2025-06-13

Rys. 5.5. Okno ucznia - Zakładka „Terminarz”

5.2.3.3. Zakładka „Uwagi”

Zakładka umożliwia przegląd uwag przypisanych do konta ucznia.



Rys. 5.6. Okno ucznia - Zakładka „Uwagi”

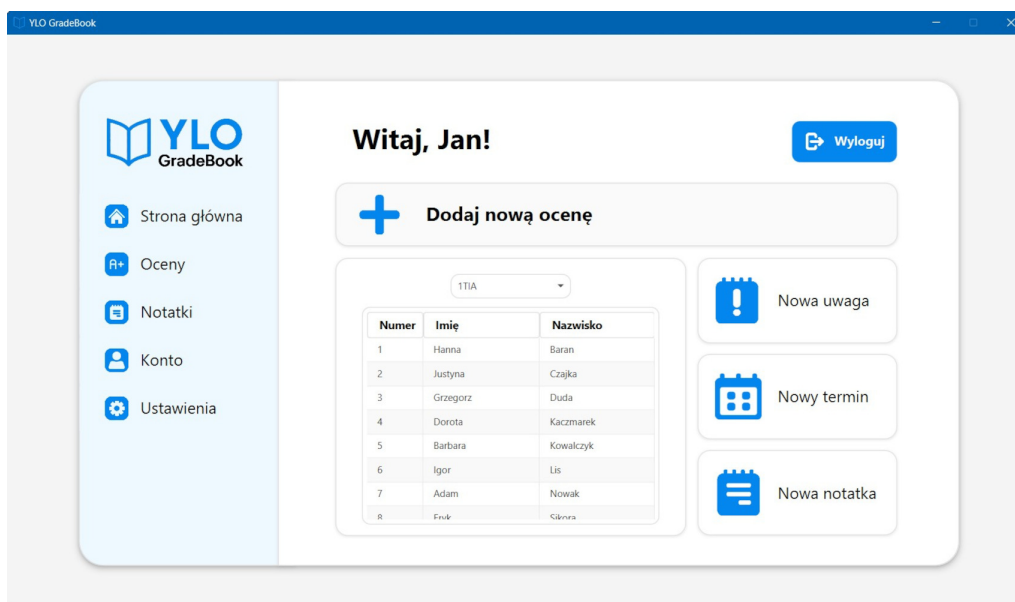
Pozostałe zakładki

Pozostałe zakładki zostaną zaprezentowane zbiorczo na końcu podrozdziału, ponieważ ich wygląd i funkcjonalność są identyczne niezależnie od przypisanej roli użytkownika

5.2.4. Główne okno interfejsu nauczyciela

Główne okno interfejsu nauczyciela pojawia się bezpośrednio po jego zalogowaniu do systemu YLO GradeBook. Zostało zaprojektowane z myślą o szybkim i intuicyjnym dostępie do najważniejszych funkcji systemu. Podział treści na zakładki oraz wyraźna struktura interfejsu ułatwiają sprawne poruszanie się po aplikacji.

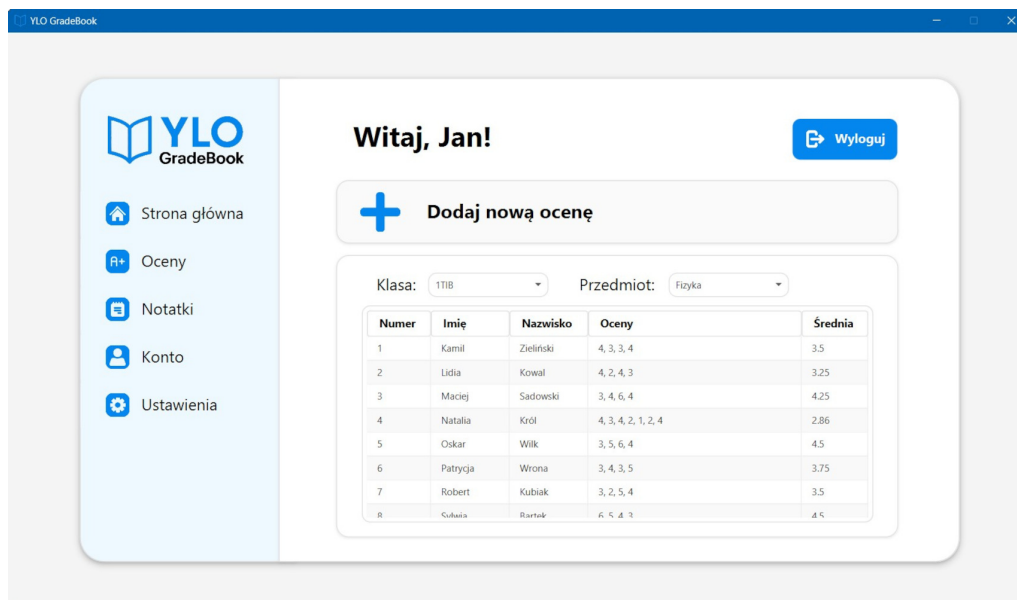
- Po lewej stronie znajduje się panel nawigacyjny z zakładkami: **Strona główna**, **Oceny**, **Notatki**, **Konto** oraz **Ustawienia**.
- W górnej części widoku wyświetlane jest powitanie użytkownika wraz z jego imieniem, a także przycisk odpowiedzialny za kluczową funkcję — **dodawanie ocen**.
- Po prawej stronie umieszczono przycisk **Wyloguj**, umożliwiający szybkie zakończenie sesji.
- Centralna część okna prezentuje listę uczniów aktualnie wybranej klasy.
- Po prawej stronie znajdują się szybkie odnośniki do najczęściej wykonywanych operacji.



Rys. 5.7. Główny widok nauczyciela

5.2.4.1. Zakładka „Oceny”

Zakładka zawiera wyróżniony przycisk umożliwiający otwarcie okna typu pop-up, w którym nauczyciel może przypisać ocenę konkretnemu uczniowi. Poniżej znajduje się tabela przedstawiająca oceny wszystkich uczniów wybranej klasy z danego przedmiotu.



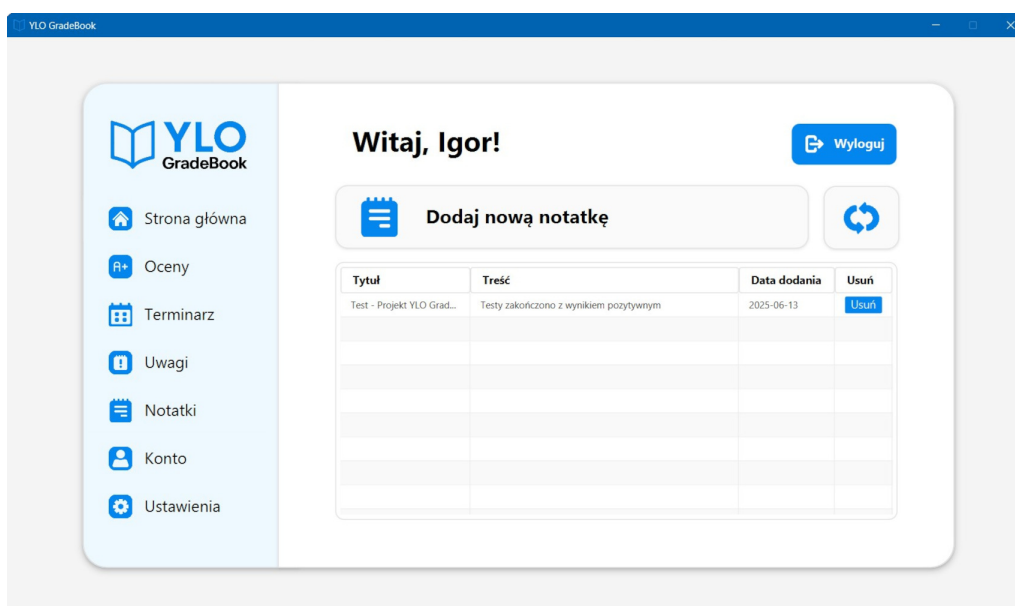
Rys. 5.8. Okno nauczyciela - Zakładka „Oceny”

5.2.5. Pozostałe zakładki

Poniżej przedstawiono listę zakładek, które posiadają identyczną budowę, niezależnie od roli zalogowanego użytkownika.

5.2.5.1. Zakładka „Notatki”

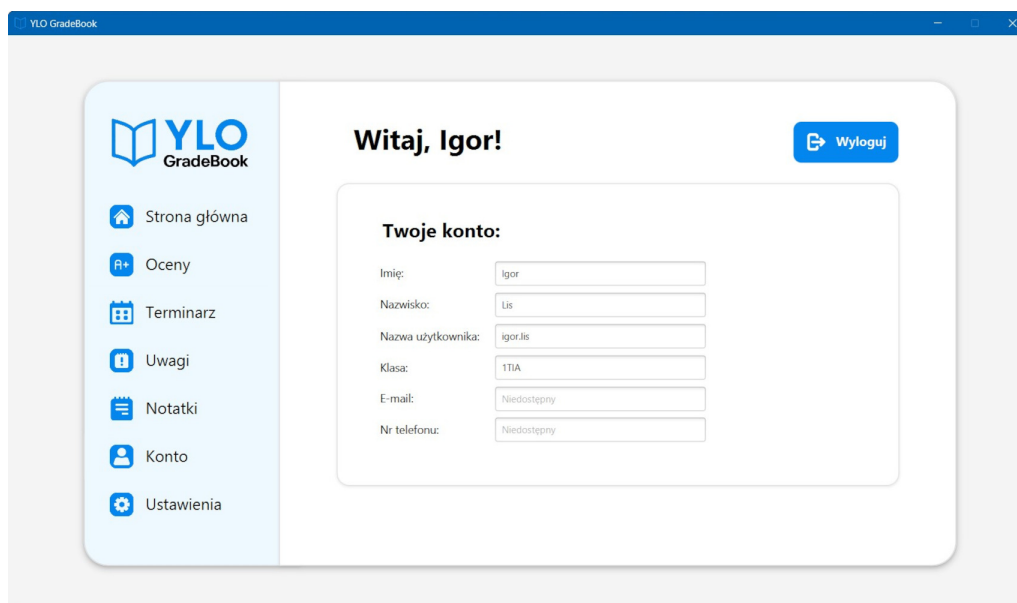
Widok zawiera przycisk umożliwiający otwarcie okna typu pop-up, w którym użytkownik może dodać własną notatkę osobistą. Obok znajduje się również przycisk służący do odświeżania tabeli prezentującej wszystkie notatki przypisane do jego konta. Przy każdym rekodzie dodano przycisk `textbfUsuń`, który umożliwia usunięcie notatki z tabeli oraz z bazy danych.



Rys. 5.9. Okno ucznia - Zakładka „Notatki”

5.2.5.2. Zakładka „Konto”

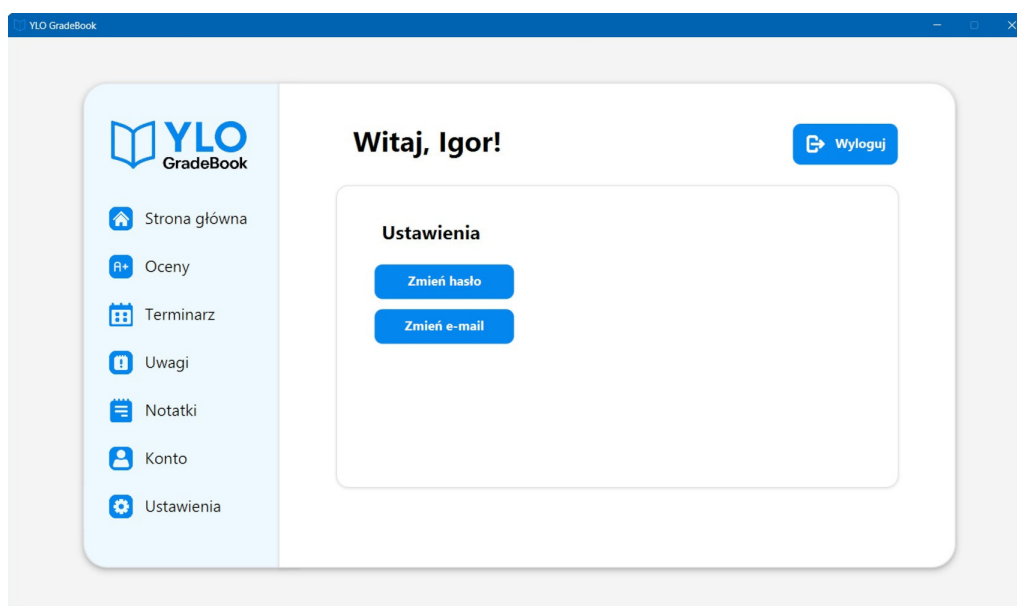
Widok pozwala na wyświetlenie danych użytkownika aktualnie zalogowanego do systemu.



Rys. 5.10. Zakładka „Konto”

5.2.5.3. Zakładka „Ustawienia”

Ostatnia zakładka umożliwia użytkownikowi zarządzanie jego danymi konta. Zawiera przycisk uruchamiający okno resetowania hasła oraz przycisk przygotowany pod planowaną funkcję zmiany adresu e-mail przypisanego do profilu użytkownika.



Rys. 5.11. Zakładka „Ustawienia”

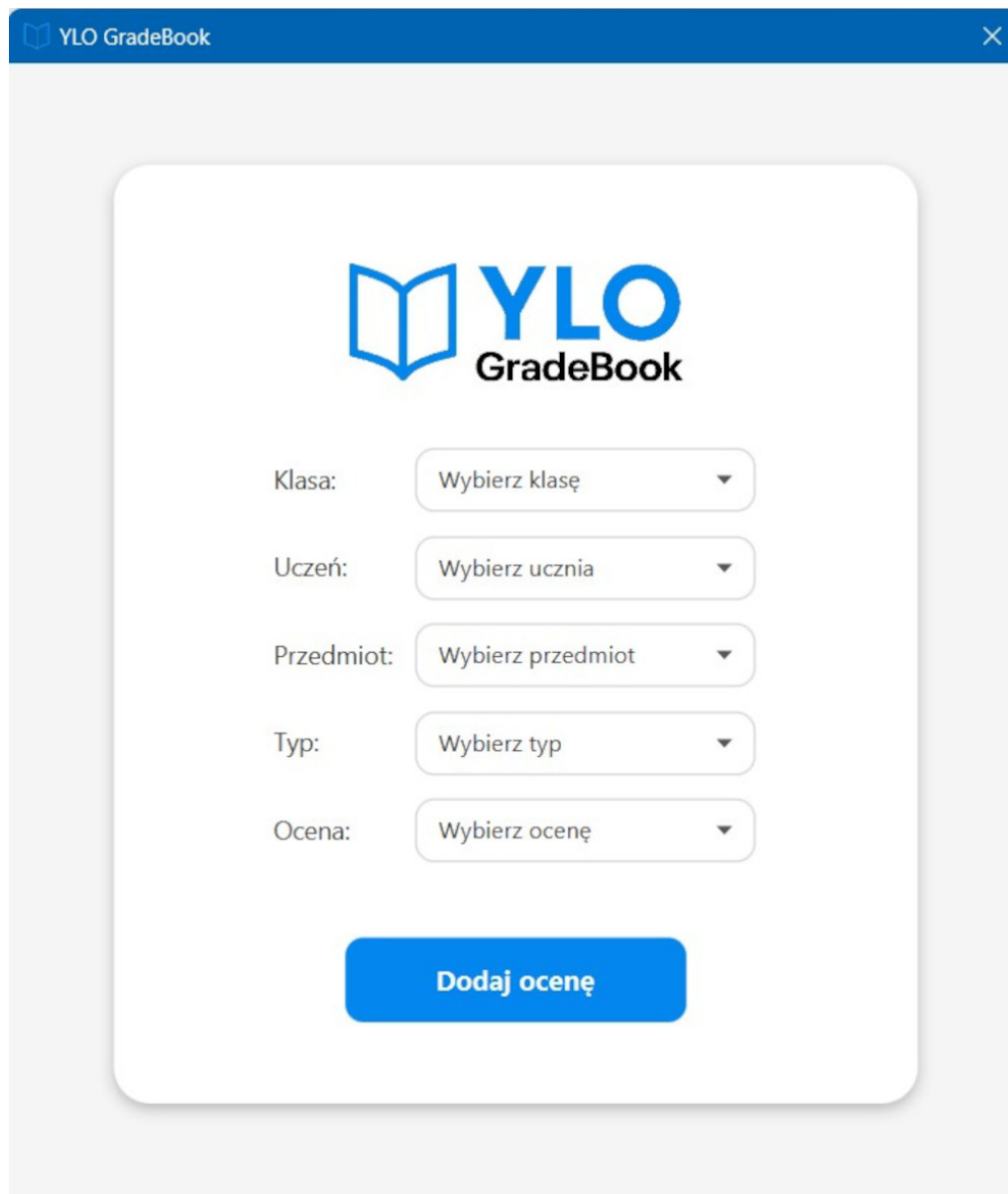
5.2.6. Widok okna PopUp

Widok ten służy do wyświetlenia formularza umożliwiającego dodanie nowych danych do systemu. W zależności od kontekstu, posiada cztery różne wersje, z których każda odpowiada innej funkcji (np. dodawanie ocen, uwagi, terminu lub notatki).

- Każda wersja zawiera pola do wprowadzania odpowiednich danych oraz przycisk umożliwiający dodanie ich do bazy danych.
- W przypadku błędnego wypełnienia formularza, system wyświetla odpowiedni komunikat walidacyjny.
- Po zatwierdzeniu formularza pojawia się komunikat informujący o rezultacie operacji — czy została wykonana pomyślnie, czy wystąpił błąd.

5.2.6.1. Okno PopUp – Dodawanie oceny

Formularz umożliwia nauczycielowi przypisanie nowej oceny wybranemu uczniowi. Widok zawiera listy rozwijane do wyboru klasy, ucznia, przedmiotu i typu oceny, a także pole do wpisania wartości punktowej.



The image shows a screenshot of a web application window titled "YLO GradeBook". Inside the window is a white rounded rectangle containing the YLO GradeBook logo at the top. Below the logo are five rows of form elements, each consisting of a label and a dropdown menu:

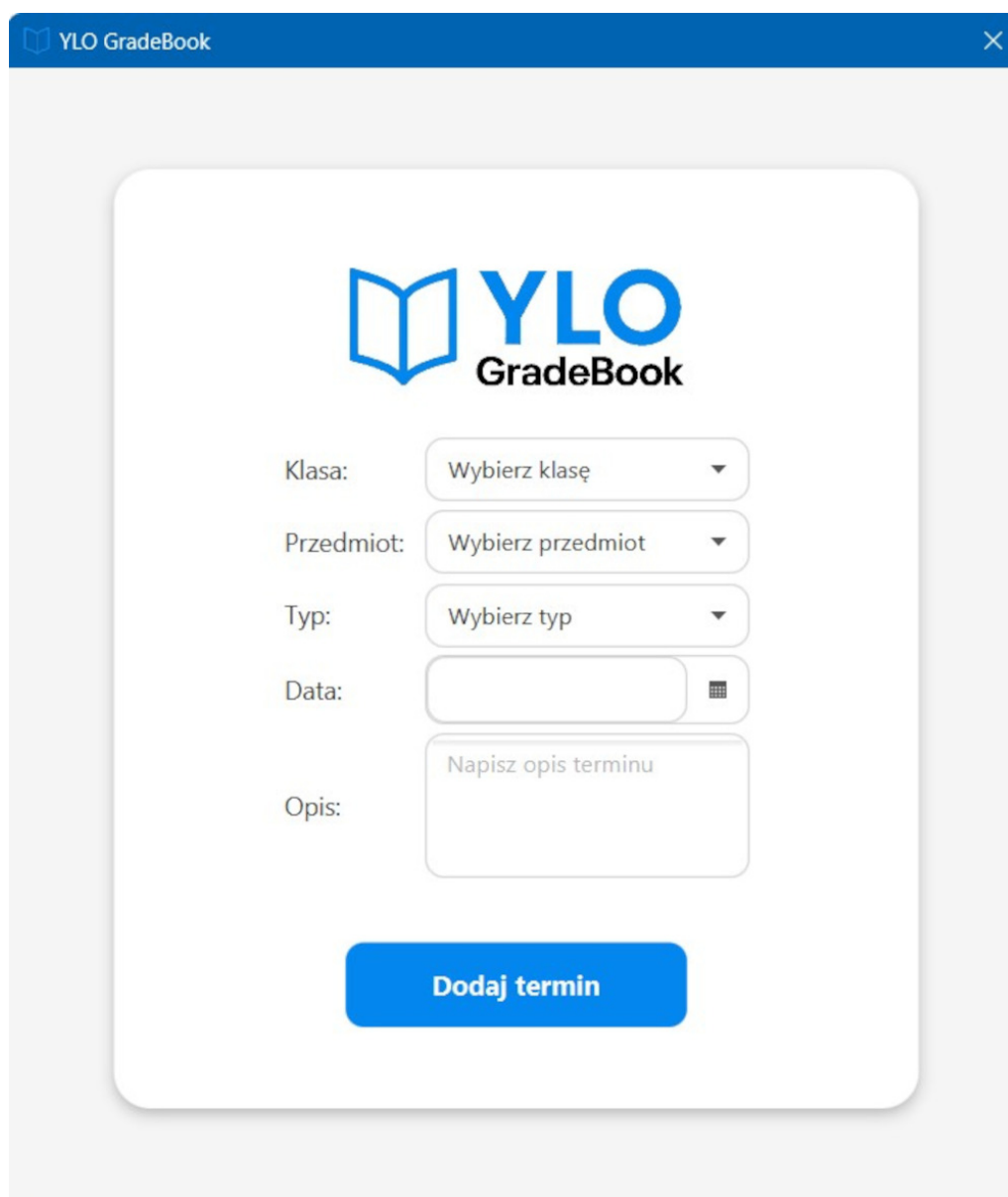
- Klasa: Wybierz klasę
- Uczeń: Wybierz ucznia
- Przedmiot: Wybierz przedmiot
- Typ: Wybierz typ
- Ocena: Wybierz ocenę

At the bottom of the form is a large blue button with the text "Dodaj ocenę".

Rys. 5.12. PopUp — dodawanie nowej oceny

5.2.6.2. Okno PopUp – Dodawanie terminów

Okno to pozwala nauczycielowi na dodanie nowego wydarzenia (np. sprawdzianu, zadania) przypisanego do konkretnej klasy i przedmiotu. Użytkownik wybiera datę oraz podaje krótki opis.

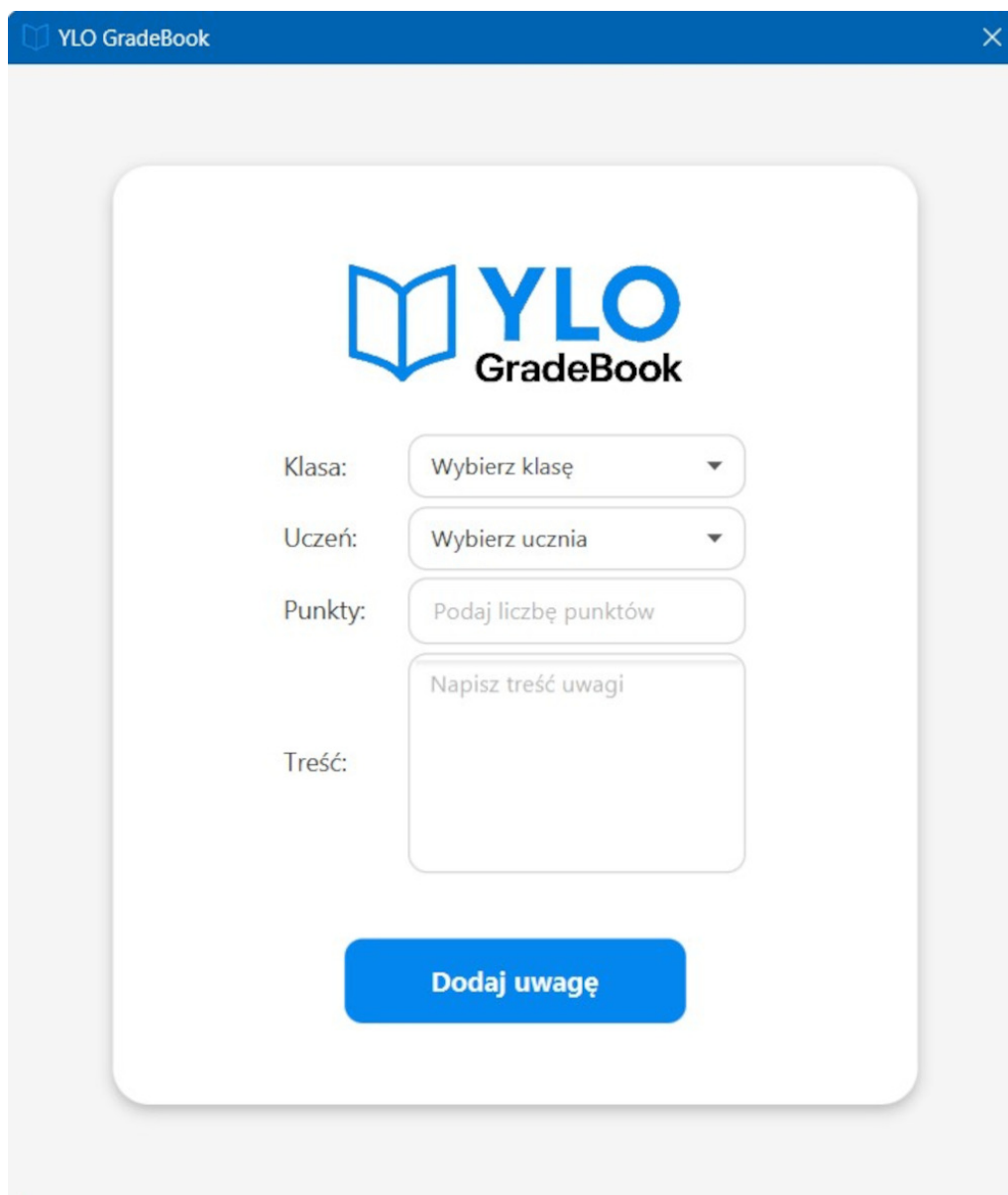


The screenshot shows a web application window titled "YLO GradeBook" with a close button in the top right corner. The main content area is a light gray rectangle containing a white rounded rectangle. At the top of the white area is the "YLO GradeBook" logo, which consists of a blue book icon and the text "YLO GradeBook". Below the logo are five form fields arranged vertically, each with a label to its left: "Klasa:" followed by a dropdown menu with the text "Wybierz klasę"; "Przedmiot:" followed by a dropdown menu with the text "Wybierz przedmiot"; "Typ:" followed by a dropdown menu with the text "Wybierz typ"; "Data:" followed by a date input field with a calendar icon; and "Opis:" followed by a text input field with the placeholder text "Napisz opis terminu". At the bottom of the white area is a blue button with the white text "Dodaj termin".

Rys. 5.13. PopUp — dodawanie nowego terminu

5.2.6.3. Okno PopUp – Dodawanie uwag

Formularz służy do wystawiania uwag punktowych uczniom. Nauczyciel wybiera ucznia, wprowadza liczbę punktów oraz wpisuje treść ograniczoną do 60 znaków.

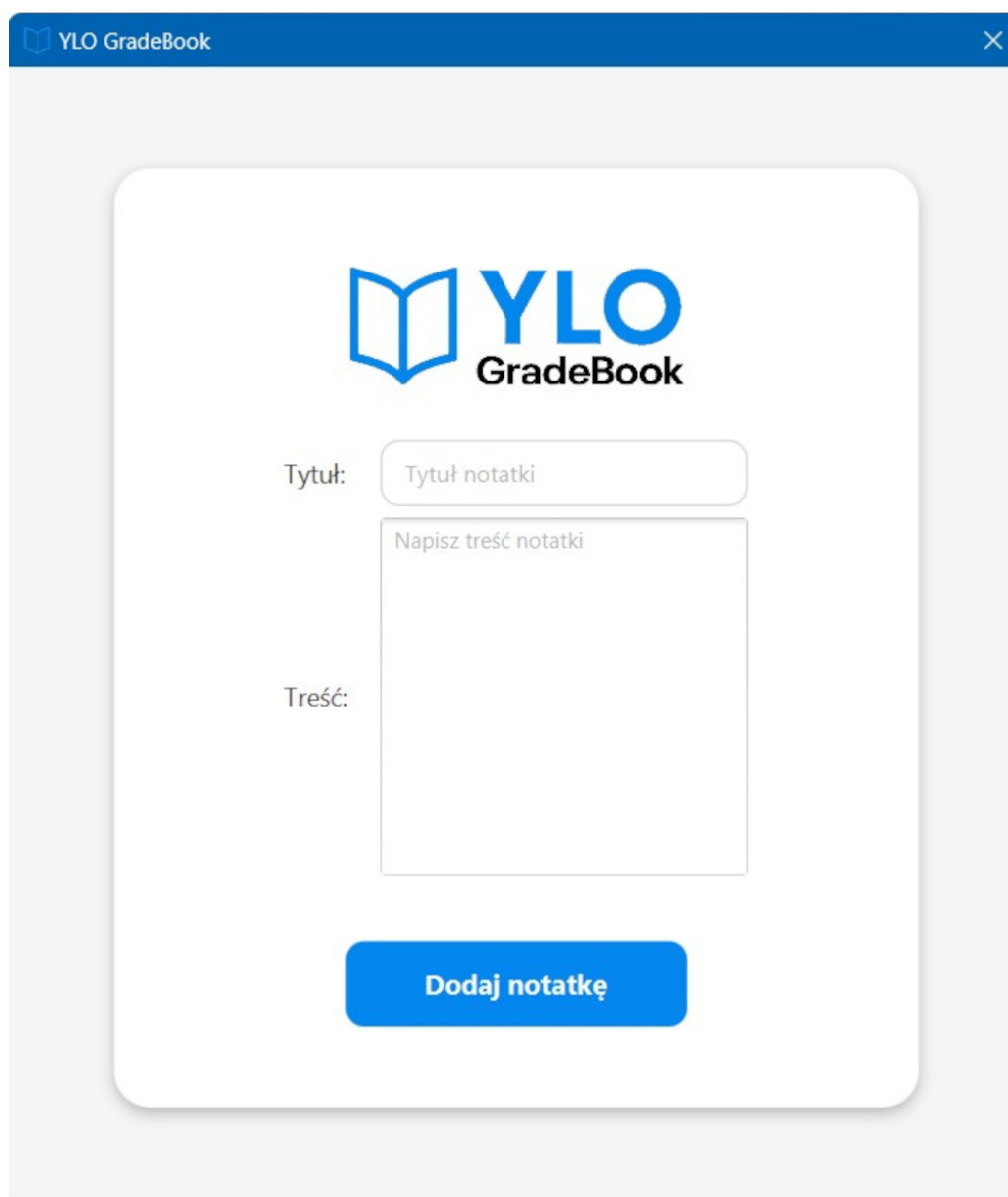


The image shows a screenshot of a web application window titled "YLO GradeBook". Inside the window is a white rounded rectangle containing the YLO GradeBook logo at the top. Below the logo are four input fields: "Klasa:" with a dropdown menu labeled "Wybierz klasę", "Uczeń:" with a dropdown menu labeled "Wybierz ucznia", "Punkty:" with a text input field labeled "Podaj liczbę punktów", and "Treść:" with a larger text input field labeled "Napisz treść uwagi". At the bottom of the white rectangle is a blue button with the text "Dodaj uwagę".

Rys. 5.14. PopUp — dodawanie uwagi uczniowi

5.2.6.4. Okno PopUp – Dodawanie notatek

Widok umożliwia dodanie osobistej notatki. Użytkownik podaje tytuł oraz treść, a następnie zapisuje ją w systemie — notatka przypisywana jest automatycznie do jego konta.



The image shows a screenshot of a web application window titled "YLO GradeBook". Inside the window is a light gray rounded rectangle containing the YLO GradeBook logo at the top. Below the logo, there are two input fields. The first is labeled "Tytuł:" and contains the placeholder text "Tytuł notatki". The second is labeled "Treść:" and contains the placeholder text "Napisz treść notatki". At the bottom of the form is a blue button with the text "Dodaj notatkę".

Rys. 5.15. PopUp — dodawanie notatki

5.3. Mechanizm alertów systemowych

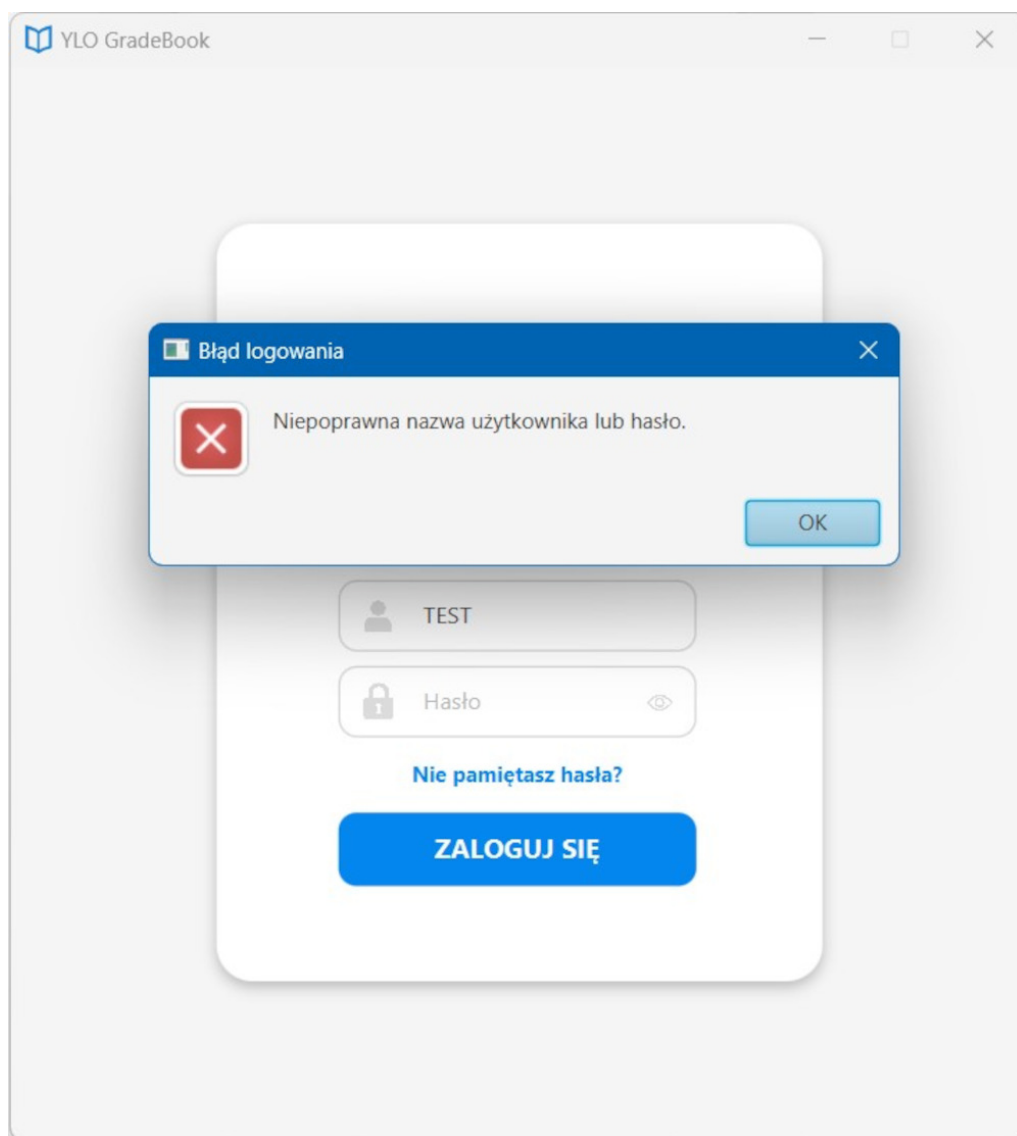
Aplikacja YLO GradeBook została wyposażona w system alertów, który informuje użytkownika o rezultacie wykonywanej operacji. Komunikaty pojawiają się m.in. przy:

- nieprawidłowym logowaniu,
- błędnym wypełnieniu formularzy,
- anulowaniu operacji,
- wylogowaniu z systemu,

Funkcjonalność ta została zrealizowana w klasie bazowej `SessionController`, która udostępnia wspólne metody. Dzięki ich zastosowaniu interfejs użytkownika informuje o przebiegu operacji w sposób przejrzysty, spójny i natychmiastowy.

Rozwiązanie to znacząco podnosi komfort i poczucie bezpieczeństwa przy wprowadzaniu danych do systemu.

Poniżej przedstawiono jeden z alertów pojawiających się podczas próby logowania.



Rys. 5.16. Przykładowy alert systemowy — nieprawidłowe logowanie

6. Podsumowanie

Celem projektu było stworzenie nowoczesnej, desktopowej aplikacji YLO GradeBook, która w przystępny sposób umożliwia zarządzanie ocenami uczniów pozwoli na organizację pracy szkolnej placówki. Dzięki zastosowaniu języka Java, biblioteki JavaFX oraz relacyjnej bazy danych MySQL, udało się zaprojektować system wyróżniający się przejrzystością interfejsu, intuicyjnością obsługi oraz rozszerzalnością struktury danych.

Zrealizowane funkcjonalności umożliwiają:

- logowanie użytkowników z rozróżnieniem ról (nauczyciel/uczeń),
- zarządzanie ocenami, terminami, notatkami i uwagami,
- dynamiczne wyświetlanie danych w zależności od kontekstu,
- edycję danych konta oraz resetowanie hasła,
- wyświetlanie średnich ocen,
- komfortową obsługę poprzez nowoczesny interfejs oparty o komponenty graficzne FXML oraz stylizację CSS.

Proces tworzenia aplikacji przebiegał etapowo: od analizy wymagań, przez projektowanie graficzne i implementację logiki, aż po testowanie oraz przygotowanie dokumentacji.

Efekt końcowy spełnia założone cele, a dzięki przemyślanej strukturze kodu i bazy danych, możliwa jest dalsza rozbudowa aplikacji. Przykładowe opcje rozwoju:

- dodanie weryfikacji dwuetapowej podczas logowania i resetowania hasła,
- zintegrowanie systemu z usługami poczty elektronicznej,
- rozszerzenie funkcjonalności o system obecności lub komunikator wewnętrzny,

Aplikacja YLO GradeBook stanowi stabilną i gotową do wdrożenia podstawę elektronicznego dziennika, ukierunkowaną na prostotę, czytelność i dopasowanie do realnych potrzeb użytkowników niezależnie od ich doświadczenia w pracy z komputerem.

7. Instrukcja uruchomienia plikacji YLO GradeBook

7.1. Wymagania środowiskowe

Do poprawnego uruchomienia aplikacji YLO GradeBook wymagane jest przygotowanie środowiska zgodnie z poniższymi specyfikacjami programowymi:

- **Środowisko JDK:** `OpenJDK 23.0.2` (Oracle)
- **JavaFX SDK:** wersja 23 (dołączona lokalnie do projektu)
- **IDE:** IntelliJ IDEA Community Edition 2024.3.3
- **Sterownik JDBC:** `mysql-connector-j-9.3.0`
- **Baza danych:** MariaDB 10.4.32 (środowisko `phpMyAdmin 5.2.1`)
- **Panel sterowania:** `XAMPP Control Panel 3.3.0`

Uwaga: Domyślnym użytkownikiem serwera bazy danych jest `root`, a hasło pozostaje puste (""), zgodnie z domyślną konfiguracją XAMPP.

Zainstalowane wtyczki środowiska IntelliJ IDEA

W trakcie pracy nad aplikacją wykorzystano kilka rozszerzeń środowiska IntelliJ IDEA. Poniżej zestawiono je z podziałem na wtyczki wymagane do poprawnego działania aplikacji oraz te, które służyły jedynie do wsparcia procesu implementacji lub dokumentacji.

Wymagane do działania aplikacji:

- **JavaFX Runtime for Plugins** — wersja `1.0.4`
(Źródło: JetBrains)
Wtyczka niezbędna do uruchamiania komponentów JavaFXw środowisku IntelliJ.

Wykorzystane pomocniczo (opcjonalne):

- **PlantUML Integration** — wersja `7.11.2-IJ2023.2`
(Autorzy: Eugene Steinberg, Vojtech Krasa)
Umożliwiła stworzenie diagramu klas systemu w postaci graficznej (rozdział 3).
- **PlantUML Parser** — wersja `0.0.9`
(Autor: shuzijun)
Parser współpracujący z powyższą wtyczką, pomocny przy generowaniu diagramu UML.
- **GitHub Copilot** — wersja `1.5.46-243`
(Źródło: GitHub)
Używana jako wsparcie przy pisaniu kodu, jednak nie jest wymagana do kompilacji ani działania aplikacji.

7.2. Baza danych

- Do repozytorium dołączony jest plik `gradebook_data_base.sql`, który zawiera strukturę i przykładowe dane bazy danych.
- Po uruchomieniu serwera MySQL, należy zaimportować bazę danych do środowiska phpMyAdmin.
- Domyślny użytkownik: `root` Hasło: `*(puste lub zgodne z konfiguracją lokalną)*`
- Po zaimportowaniu bazy danych należy upewnić się, że jej nazwa to `gradebook_data_base`, zgodnie z konfiguracją aplikacji w klasie `DataBaseConnection`.

7.3. Kroki uruchomienia projektu YLO GradeBook

Poniżej przedstawiono listę kroków, prowadzącą do uruchomienia aplikacji.

1. Klonowanie projektu z GitHub

1. Przejdź do repozytorium projektu:
`https://github.com/oleiy/YLO-GradeBook`
2. Pobierz repozytorium lokalnie, wykonując:
`git clone https://github.com/oleiy/YLO-GradeBook`
3. Otwórz folder projektu w środowisku IntelliJ IDEA Community Edition 2024.3.3

2. Konfiguracja środowiska Java i JavaFX

1. Skonfiguruj środowisko zgodnie z powyższą specyfikacją.

3. Dodanie sterownika JDBC

1. Umieść folder `mysql-connector-j-9.3.0` w katalogu projektu (dołączony do repozytorium)
2. Dodaj go jako bibliotekę do projektu w `Project Structure` → `Modules`

4. Import bazy danych MySQL (phpMyAdmin)

- Uruchom środowisko XAMPP Control Panel 3.3.0 i wystartuj moduł MySQL
- Otwórz phpMyAdmin w przeglądarce (`http://localhost/phpmyadmin`)
- Utwórz nową bazę danych o nazwie `gradebook_data_base`
- Zaimportuj plik `gradebook_data_base.sql` z repozytorium
- Domyślny użytkownik: `root`, hasło: (puste)

5. Uruchomienie aplikacji

1. Kliknij przycisk Run znajdując się w klasie `Main.java`
2. Aplikacja otworzy ekran logowania

6. Dane testowe (do logowania)

- **Nauczyciel:**

- nazwa użytkownika: `jan.nowak`
- hasło: `jan`

- **Uczeń:**

- nazwa użytkownika: `igor.lis`
- hasło: `igor`

Po wykonaniu powyższych kroków środowisko programistyczne jest gotowe do pracy projektu YLO GradeBook. W przypadku trudności zaleca się weryfikację konfiguracji JavaFX oraz parametrów połączenia z bazą danych w klasie `DataBaseConnection`.

Bibliografia

- [1] Freepik. Zestaw ikon użytych w interfejsie aplikacji ylo gradebook, 2025. Wszystkie ikony zostały zmodyfikowane graficznie na potrzeby projektu.

Spis rysunków

3.1	Diagram ERD bazy danych <i>gradebook_data_base</i>	17
3.2	Diagram klas systemu YLO GradeBook	24
4.1	Wykres Gantta przedstawiający harmonogram realizacji projektu YLO GradeBook	26
5.1	Okno logowania - weryfikacja danych uwierzytelniających	28
5.2	Okno resetowania hasła - aktualizacja hasła w bazie danych	29
5.3	Główny widok ucznia	30
5.4	Okno ucznia - Zakładka „Oceny”	31
5.5	Okno ucznia - Zakładka „Terminarz”	31
5.6	Okno ucznia - Zakładka „Uwagi”	32
5.7	Główny widok nauczyciela	33
5.8	Okno nauczyciela - Zakładka „Oceny”	34
5.9	Okno ucznia - Zakładka „Notatki”	35
5.10	Zakładka „Konto”	36
5.11	Zakładka „Ustawienia”	36
5.12	PopUp — dodawanie nowej oceny	38
5.13	PopUp — dodawanie nowego terminu	39
5.14	PopUp — dodawanie uwagi uczniowi	40
5.15	PopUp — dodawanie notatki	41
5.16	Przykładowy alert systemowy — nieprawidłowe logowanie	42

Załącznik nr 2 do Zarządzenia nr 228/2021 Rektora Uniwersytetu Rzeszowskiego z dnia 1 grudnia 2021 roku w sprawie ustalenia procedury antyplagiatowej w Uniwersytecie Rzeszowskim

OŚWIADCZENIE STUDENTA O SAMODZIELNOŚCI PRACY

.....Oliwier Hędrzak.....
Imię (imiona) i nazwisko studenta

Wydział Nauk Ścisłych i Technicznych

.....Informatyka.....
Nazwa kierunku

.....134913.....
Numer albumu

1. Oświadczam, że moja praca projektowa pt.: Aplikacja YLO GradeBook – dokumentacja projektowa
 - 1) została przygotowana przeze mnie samodzielnie*,
 - 2) nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2021 r., poz. 1062) oraz dóbr osobistych chronionych prawem cywilnym,
 - 3) nie zawiera danych i informacji, które uzyskałem/am w sposób niedozwolony,
 - 4) nie była podstawą otrzymania oceny z innego przedmiotu na uczelni wyższej ani mnie, ani innej osobie.
2. Jednocześnie wyrażam zgodę/nie wyrażam zgody** na udostępnienie mojej pracy projektowej do celów naukowo-badawczych z poszanowaniem przepisów ustawy o prawie autorskim i prawach pokrewnych.

(miejscowość, data)

(czytelny podpis studenta)

* Uwzględniając merytoryczny wkład prowadzącego przedmiot

** – niepotrzebne skreślić