**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](#_Toc161677644)

[2. Specyfikacja projektu 3](#_Toc161677645)

[2.1. Cel projektu 3](#_Toc161677646)

[2.2. Zakres projektu 3](#_Toc161677647)

[2.2.1. Zasoby 3](#_Toc161677648)

[2.2.2. Produkty końcowe projektu 4](#_Toc161677649)

[2.2.3. Harmonogram realizacji projektu 4](#_Toc161677650)

[2.3. Wymagania stawiane aplikacji / systemowi 4](#_Toc161677651)

[2.4. Panele / zakładki systemu, które będą oferowały potrzebne funkcjonalności 4](#_Toc161677652)

[2.5.1. Raporty PDF 5](#_Toc161677653)

[2.5.2. Inne dokumenty 5](#_Toc161677654)

[2.6. Przepływ informacji w środowisku systemu 5](#_Toc161677655)

[2.7. Użytkownicy aplikacji i ich uprawnienia 5](#_Toc161677656)

[2.8. Interesariusze 6](#_Toc161677657)

[3. Diagramy UML 6](#_Toc161677658)

[3.1. Diagram przypadków użycia 6](#_Toc161677659)

[3.2. Diagramy aktywności 6](#_Toc161677660)

[3.3. Diagramy sekwencji 6](#_Toc161677661)

[4. Baza danych 6](#_Toc161677662)

[4.1. Diagram ERD 6](#_Toc161677663)

[4.2. Skrypt do utworzenia struktury bazy danych 6](#_Toc161677664)

[5. Wykorzystane technologie 6](#_Toc161677665)

[6. Interfejs aplikacji / systemu 7](#_Toc161677666)

# 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

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



Rysunek . Diagram ERD

## Skrypt do utworzenia struktury bazy danych

Table Users {

id int [pk, increment]

personal\_id int

password varchar

role\_id int

email varchar

class\_id int [null] // Nie każdy użytkownik musi być przypisany do klasy

}

Table PersonalInfo {

id int [pk, increment]

first\_name varchar

last\_name varchar

country varchar

date\_of\_birth date

pesel varchar

phone\_number varchar

}

Table Classes {

id int [pk, increment]

name varchar

}

Table Subjects {

id int [pk, increment]

name varchar

}

Table ClassSubjects {

class\_id int

subject\_id int

}

Table TeacherSubjectsClasses {

user\_id int

subject\_id int

class\_id int // Połączenie nauczyciela, przedmiotu i klasy

}

Table Roles {

id int [pk, increment]

name varchar

}

Table Tasks {

id int [pk, increment]

title varchar

description text

due\_date datetime

completion\_date datetime [null] // Data zakończenia zadania

status\_id int

feedback text [null]

created\_by int

created\_at datetime

attachment blob

grade\_id int

subject\_id int [null] // Zadania mogą być powiązane z przedmiotem

}

Table TaskAssignments {

task\_id int

user\_id int // Przypisania zadań do użytkowników (uczniów lub grup uczniów)

}

Table Grades {

id int [pk, increment]

grade int

created\_at date

}

Table Notifications {

id int [pk, increment]

user\_id int

message text

date\_sent datetime

}

Table Raports {

id int [pk, increment]

user\_id int

report\_type varchar

generated\_date datetime

description text [null]

}

Table ParentChild {

parent\_id int

child\_id int

}

Table Statuses {

id int [pk, increment]

name varchar

}

// Relacje

Ref: Users.class\_id > Classes.id

Ref: ClassSubjects.class\_id > Classes.id

Ref: ClassSubjects.subject\_id > Subjects.id

Ref: TeacherSubjectsClasses.user\_id > Users.id

Ref: TeacherSubjectsClasses.subject\_id > Subjects.id

Ref: TeacherSubjectsClasses.class\_id > Classes.id

Ref: Tasks.created\_by > Users.id

Ref: Tasks.subject\_id > Subjects.id

Ref: TaskAssignments.task\_id > Tasks.id

Ref: TaskAssignments.user\_id > Users.id

Ref: Tasks.grade\_id > Grades.id

Ref: Notifications.user\_id > Users.id

Ref: Raports.user\_id > Users.id

Ref: ParentChild.parent\_id > Users.id

Ref: ParentChild.child\_id > Users.id

Ref: Users.role\_id > Roles.id

Ref: Users.personal\_id > PersonalInfo.id

Ref: Tasks.status\_id > Statuses.id

# 

# Wykorzystane technologie

* Spring Boot
  + Framework ułatwiający tworzenie aplikacji opartych na Springu poprzez automatyzację konfiguracji i zarządzania zależnościami, wykorzystywany do szybkiego tworzenia niezawodnych i łatwo skalowalnych usług backendowych (backend).
* Język Java 17
  + Jest to zaawansowaną wersją języka programowania używanego do budowy serwerowej logiki aplikacji, zapewniając stabilność i wsparcie długoterminowe (backend).
* Baza danych MySQL
  + System zarządzania bazą danych, który przechowuje, organizuje i udostępnia dane dla aplikacji, stanowiąc centralny element zarządzania danymi (backend).
* Electron
  + Umożliwia tworzenie natywnych aplikacji desktopowych przy użyciu technologii webowych, służąc jako platforma do budowy interfejsów użytkownika w aplikacjach desktopowych (frontend/desktop).
* Vite
  + Nowoczesne narzędzie do budowy, które zapewnia szybki czas startu i natychmiastowy hot reloading, używane głównie do efektywnej pracy nad interfejsem użytkownika (frontend).
* React
  + Biblioteka JavaScript do budowania interaktywnych interfejsów użytkownika, często używana do tworzenia dynamicznych stron i aplikacji internetowych (frontend).

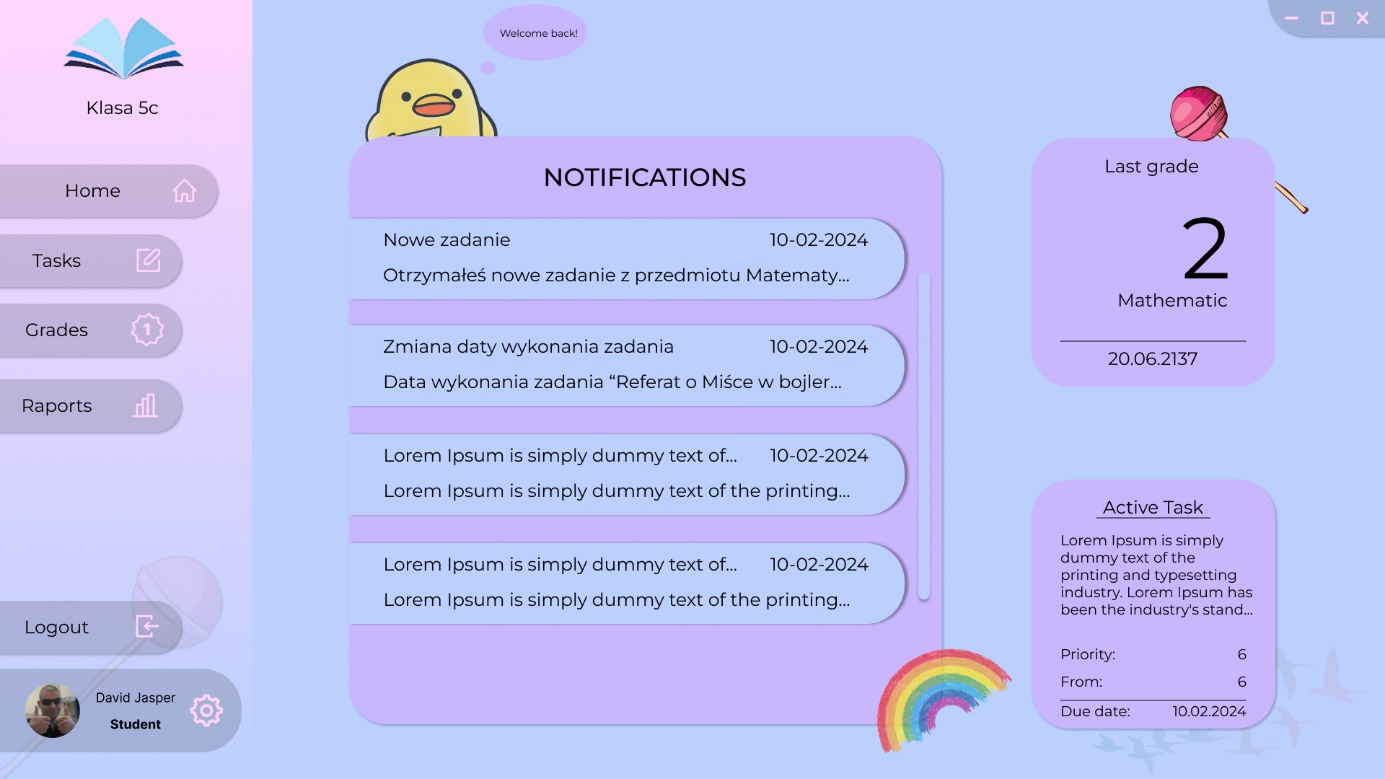
# Interfejs aplikacji / systemu

Na rysunku 10 został przedstawiony ekran logowania do aplikacji.A group of people posing for a picture

Description automatically generated

Rysunek . Logowanie

Użytkownik po zalogowaniu trafia do ekranu głównego. Na rysunku 11 został przedstawiony ekran główny dla ucznia, który przedstawia ostatnie powiadomienia z aplikacji, ostatnią otrzymaną ocenę oraz aktywne zadanie wybrane przez ucznia.



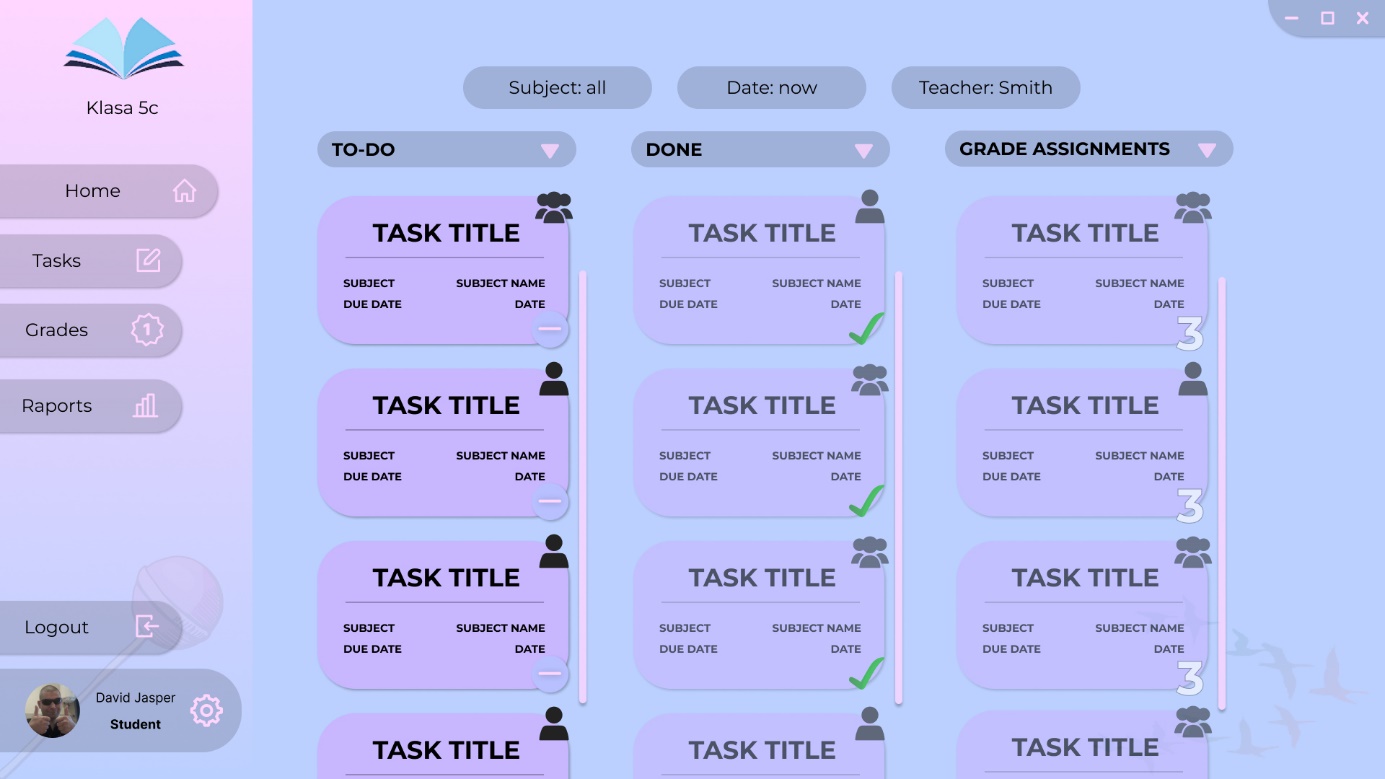
Rysunek . Ekran główny ucznia

Rysunek 12 przedstawia ekran ocen ucznia, w którym uczeń może sprawdzić swoje oceny za wykonane zadania z danych przedmiotów.



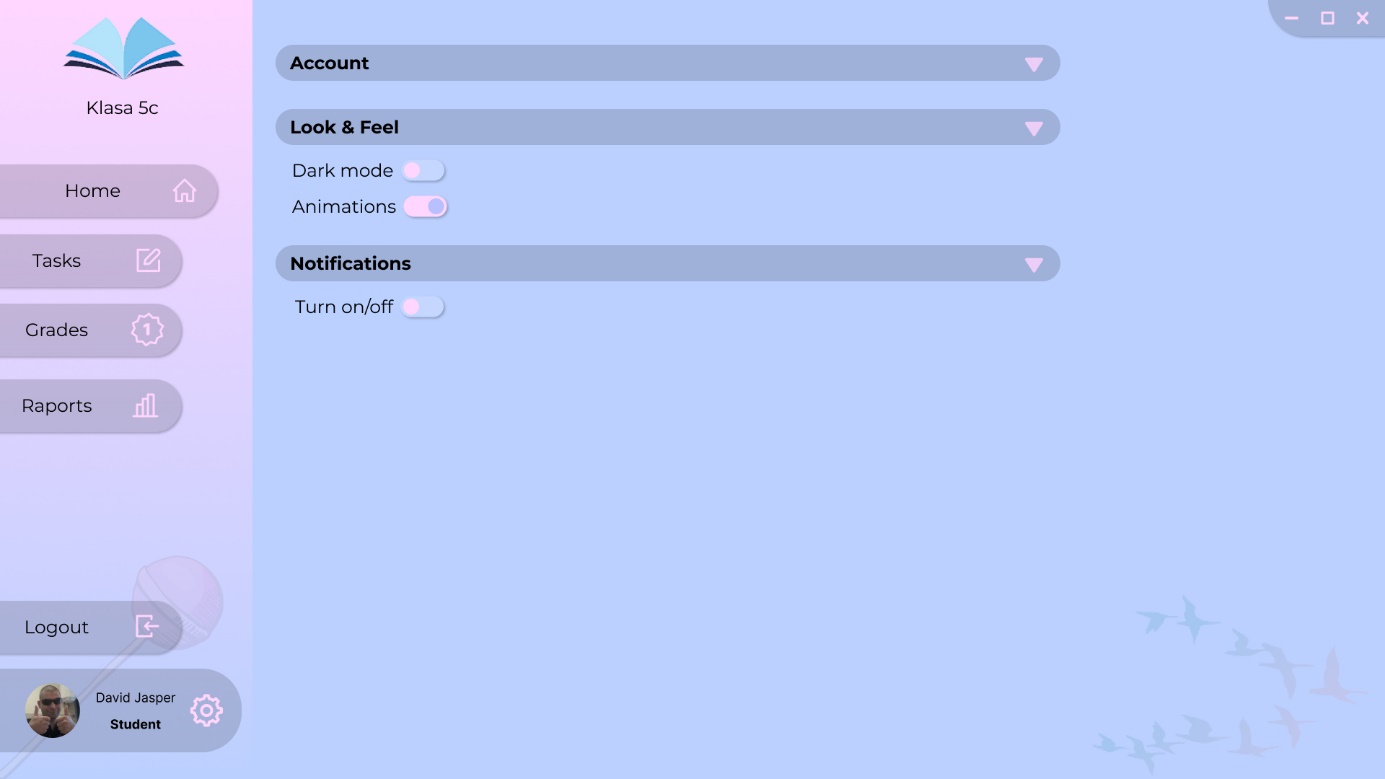
Rysunek . Ekran ocen ucznia

Na rysunku 13 przedstawiony jest ekran zadań ucznia, w którym uczeń może sprawdzić przypisane mu zadania, przeciągać do odpowiednich sekcji oraz sprawdzać ostatnio ocenione zadania.



Rysunek . Ekran zadań ucznia

Ekran ustawień przedstawiony na rysunku 14 umożliwia zmianę trybu wyświetlania (jasny, ciemny), wyłączenie animacji i inne.



Rysunek . Ekran ustawień