

# Dokumentacja wykonawcza

Eryk Mika, Michał Łabowicz, Wojciech Hajnowski

Listopad 2023

## 1 Model danych

### 1.1 Tabele w bazie danych

Przyjęto model danych, w którym zdefiniowane są następujące tabele:

1. Studenci - przechowująca dane o studentach z poszczególnych kierunków i semestrów
2. Oceny - oceny studentów
3. Kursy - zawiera informacje o kursach
4. Prowadzący - dane prowadzących
5. Kierunki studiów
6. Komunikaty

Ponadto założono stworzenie dwóch tabel, które łączą dwie inne tabele na zasadzie relacji "wiele do wielu". Są to tabele:

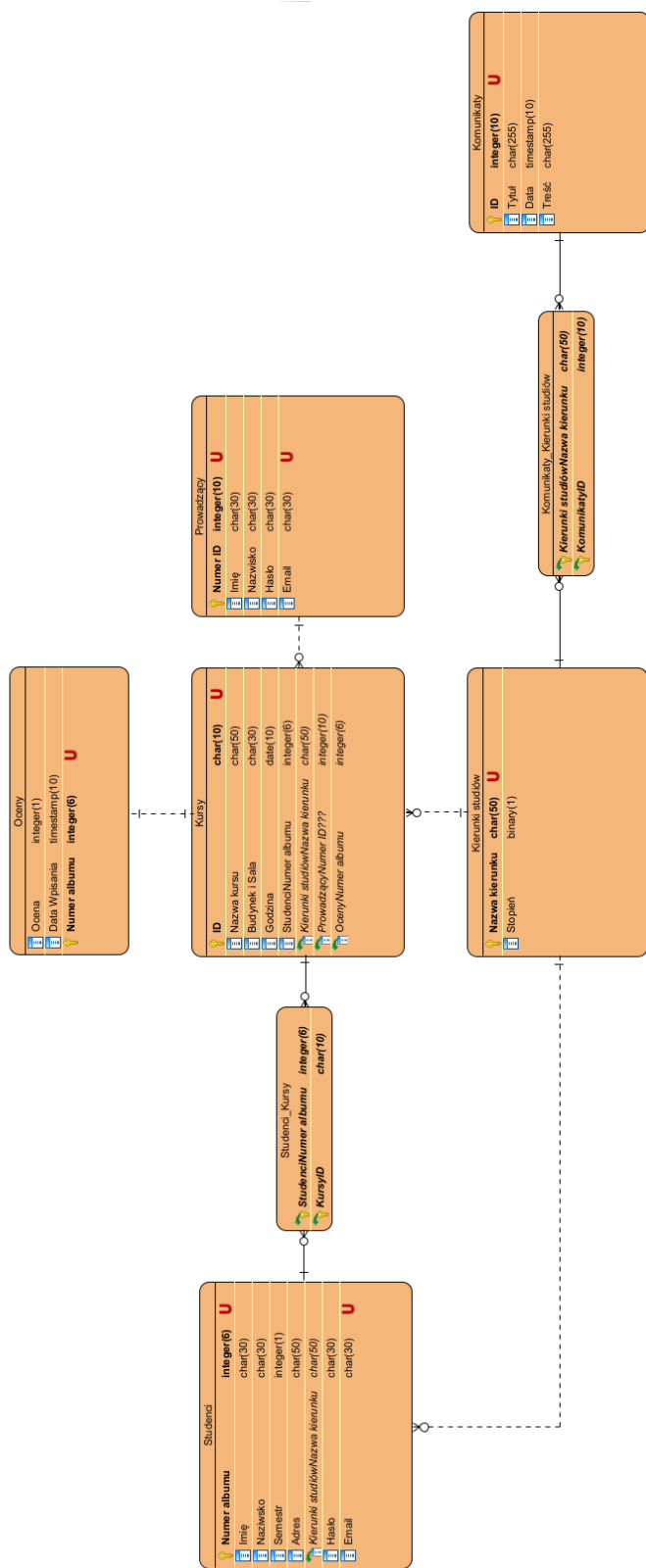
1. Studenci-Kursy
2. Komunikaty-Kierunki-Studiów

### 1.2 Relacje w bazie danych

Relacje w bazie danych są przedstawione na schemacie ERD dostępnym na następnej stronie (Rysunek 1).

### 1.3 Implementacja bazy danych

Przyjęto, że jako system bazodanowy zostanie użyty SQLite. Zostaną napisane skrypty budujące strukturę bazy danych oraz sprawdzające jej poprawność. Zostaną napisane skrypty umożliwiające przeprowadzenie operacji CRUD.



Rysunek 1: Model danych

## **2 Implementacja - założenia**

### **2.1 Wybrane technologie**

Należy wykorzystać następujące technologie:

1. Python - jako główny język programowania użyty do stworzenia aplikacji webowej oraz skryptów obsługujących bazę danych
2. Flask - mikroframework tworzenia aplikacji webowej
3. SQLite - system bazodanowy
4. git i GitHub - system kontroli wersji oraz jego hosting
5. Redmine - system zarządzania projektami

### **2.2 Organizacja pracy**

W systemie Redmine został zdefiniowany harmonogram prac na cały semestr. Członkowie grupy mogą sprawdzać przydzielone zadania i je rozliczać. Praca z kodem odbywa się w repozytorium git, w którym dla poszczególnych zadań tworzone są gałęzie (branche), które następnie są mergowane do głównej gałęzi main za zasadzie pull requestów.

## **3 Wstępna instrukcja obsługi**

Użytkownik przed rozpoczęciem działań w aplikacji musi się zalogować do systemu. Może to zrobić jako student, prowadzący lub administrator. Logowanie następuje z wykorzystaniem unikatowego adresu email oraz hasła. Następnie do wyboru są następujące funkcjonalności aplikacji:

### **3.1 Dostęp do ocen**

Każdy student może sprawdzić swoje oceny. Prowadzący może sprawdzić oraz edytować oceny swoich studentów. Administrator ma dostęp do wszystkich ocen.

### **3.2 Dostęp do planu zajęć**

Każdy student może sprawdzić swój plan zajęć. Prowadzący może sprawdzić swój plan zajęć - prowadzonych przez siebie grup. Administrator może filtrować i wyświetlać plan zajęć dowolnego użytkownika.

### **3.3 Zapisy na zajęcia**

Student może zapisywać się do grup zajęciowych przeznaczonych dla jego kierunku. Prowadzący może zapisać studentów do swoich grup. Administrator może zapisać studenta do każdej grupy przeznaczonej dla jego kierunku.

### **3.4 Przeglądanie komunikatów z uczelni**

Student może przeglądać komunikaty przeznaczone dla jego kierunku studiów. Prowadzący może przeglądać komunikaty kierunków studiów, dla których prowadzi zajęcia. Administrator widzi wszystkie komunikaty.

### **3.5 Dostęp do informacji o studencie**

Student może wyświetlić swoje dane. Prowadzący może wyświetlić dane swoich studentów. Administrator może wyświetlić dane dowolnego studenta.