**Akademia Nauk Stosowanych w Elblągu**



Instytut Informatyki Stosowanej im. Krzysztofa Brzeskiego

**PRACA INŻYNIERSKA**

**Temat: Aplikacja wspierająca planowanie obciążeń dydaktycznych w systemie USOS.**

***Application to support the planning of didactic loads in the USOS system***

Imię i nazwisko: Łukasz Gajewski

Kierunek studiów: Informatyka

Specjalizacja: Projektowanie baz danych i oprogramowanie użytkowe

Opiekun pracy: dr inż. Jerzy Buriak

Elbląg, rok akadem. 2024/2025

**Spis treści**

[Wstęp 3](#_Toc194941661)

[1. Analiza wymagań i projekt interfejsu użytkownika 4](#_Toc194941662)

[1.1. Logika biznesowa i analiza schematów bazy USOS 4](#_Toc194941663)

[1.1.1. Integracja z bazą danych USOS: 4](#_Toc194941664)

[1.1.2 Obliczanie obciążeń dydaktycznych: 4](#_Toc194941665)

[1.1.3 Generowanie raportów: 4](#_Toc194941666)

[1.2. Wymagania funkcjonalne i niefunkcjonalne 5](#_Toc194941667)

[1.2.1. Wymagania funkcjonalne: 5](#_Toc194941668)

[1.2.2. Wymagania niefunkcjonalne: 6](#_Toc194941669)

[1.3. Projekt interfejsu użytkownika w PyQt5 6](#_Toc194941670)

[1.3.1. Struktura okna głównego (MainWindow): 6](#_Toc194941671)

[1.3.2. Funkcjonalności interaktywne: 6](#_Toc194941672)

[1.3.3. Przykładowy fragment kodu: 7](#_Toc194941673)

[1.3.4. Wizualizacja interfejsu: 7](#_Toc194941674)

[1.3.5. Obsługa zdarzeń: 7](#_Toc194941675)

[2. Implementacja aplikacji desktopowej w PyQt5 8](#_Toc194941676)

[2.1 Projektowanie okien aplikacji 8](#_Toc194941677)

[2.2 Implementacja przeglądania obciążeń dydaktycznych 8](#_Toc194941678)

[2.3 Kontrola dostępu na podstawie ról użytkowników 8](#_Toc194941679)

[3. Połączenie z bazą danych i logika biznesowa 8](#_Toc194941680)

[3.1 Implementacja modelu danych w SQLAlchemy 8](#_Toc194941681)

[3.2 Obsługa zapytań SQL 8](#_Toc194941682)

[3.3 Obliczanie pensum dydaktycznego 8](#_Toc194941683)

[4. Eksport danych i generowanie raportów 8](#_Toc194941684)

[4.1 Generowanie raportów Excel 8](#_Toc194941685)

[4.2 Filtrowanie raportów według jednostek, lat akademickich i stanowisk 8](#_Toc194941686)

[5. Testowanie, bezpieczeństwo i wdrożenie 8](#_Toc194941687)

[5.1 Testy funkcjonalne i integracja z aplikacją 8](#_Toc194941688)

[5.2 Optymalizacja połączenia z bazą USOS 8](#_Toc194941689)

[5.3 Wdrożenie aplikacji jako plik wykonywalny 8](#_Toc194941690)

[6. Podsumowanie i wnioski 8](#_Toc194941691)

[7. Literatura 8](#_Toc194941692)

# **Wstęp**

**Obciążenia dydaktyczne stanowią kluczowy element zarządzania zasobami akademickimi w uczelniach wyższych. W systemie USOS proces ten jest realizowany poprzez złożone reguły zapisane w skoroszytach Excel, co generuje szereg wyzwań związanych z aktualnością danych, ich przejrzystością oraz możliwością analizy. Wdrożenie nowego modułu rozliczeń dydaktycznych w systemie USOS nie eliminuje potrzeby istnienia narzędzia pomocniczego dla dyrekcji instytutów i dziekanatów, które pozwoli na szybkie i intuicyjne monitorowanie obciążeń.**

**Celem niniejszej pracy inżynierskiej jest opracowanie samodzielnej aplikacji desktopowej w języku Python z wykorzystaniem biblioteki PyQt5. Aplikacja ta ma umożliwiać pobieranie, przeglądanie, edycję oraz analizę danych dotyczących obciążeń dydaktycznych bezpośrednio z bazy USOS. System zapewni wygodne narzędzie wspierające proces podejmowania decyzji przez kadrę zarządzającą, ułatwiając analizę i raportowanie obciążeń.**

**Zakres pracy obejmuje analizę wymagań, projekt interfejsu użytkownika, implementację aplikacji, integrację z bazą danych oraz opracowanie funkcji generowania raportów. Istotnym aspektem projektu jest zapewnienie bezpieczeństwa danych oraz optymalizacji wydajności połączenia z bazą USOS, tak aby system był nie tylko funkcjonalny, ale również odporny na szybkie dezaktualizowanie się danych. Wdrożenie aplikacji jako samodzielnego pliku wykonywalnego umożliwi łatwą dystrybucję i użytkowanie narzędzia przez odpowiednie jednostki uczelni.**

**Praca ta stanowi próbę usprawnienia procesu zarządzania obciążeniami dydaktycznymi poprzez stworzenie intuicyjnego i skutecznego narzędzia informatycznego, które może być wykorzystane przez dyrekcje instytutów oraz dziekanaty w celu efektywnego planowania i monitorowania zasobów dydaktycznych.**

# Analiza wymagań i projekt interfejsu użytkownika

## Logika biznesowa i analiza schematów bazy USOS

**System Rozliczania Obciążeń Dydaktycznych (ROD) został zaprojektowany w celu automatyzacji procesu obliczania pensum pracowników dydaktycznych na podstawie danych z systemu USOS. Główne komponenty logiki biznesowej obejmują:**

### 1.1.1. Integracja z bazą danych USOS:

**System wykorzystuje bezpośrednie połączenie z bazą danych Oracle za pośrednictwem SQLAlchemy, odczytując tabele takie jak DZ\_PRACOWNICY, DZ\_GRUPY, DZ\_PROWADZACY\_GRUP oraz DZ\_CYKLE\_DYDAKTYCZNE.**

**Dane są mapowane na modele ORM (np. Employee, Group, GroupInstructor), co umożliwia operacje CRUD bez pisania natywnych zapytań SQL.**

### 1.1.2 Obliczanie obciążeń dydaktycznych:

**Algorytm uwzględnia godziny prowadzenia zajęć (GroupInstructor.LICZBA\_GODZ\_DO\_PENSUM), opiekę nad pracami dyplomowymi (ThesisSupervisors) oraz recenzje (Reviewer).**

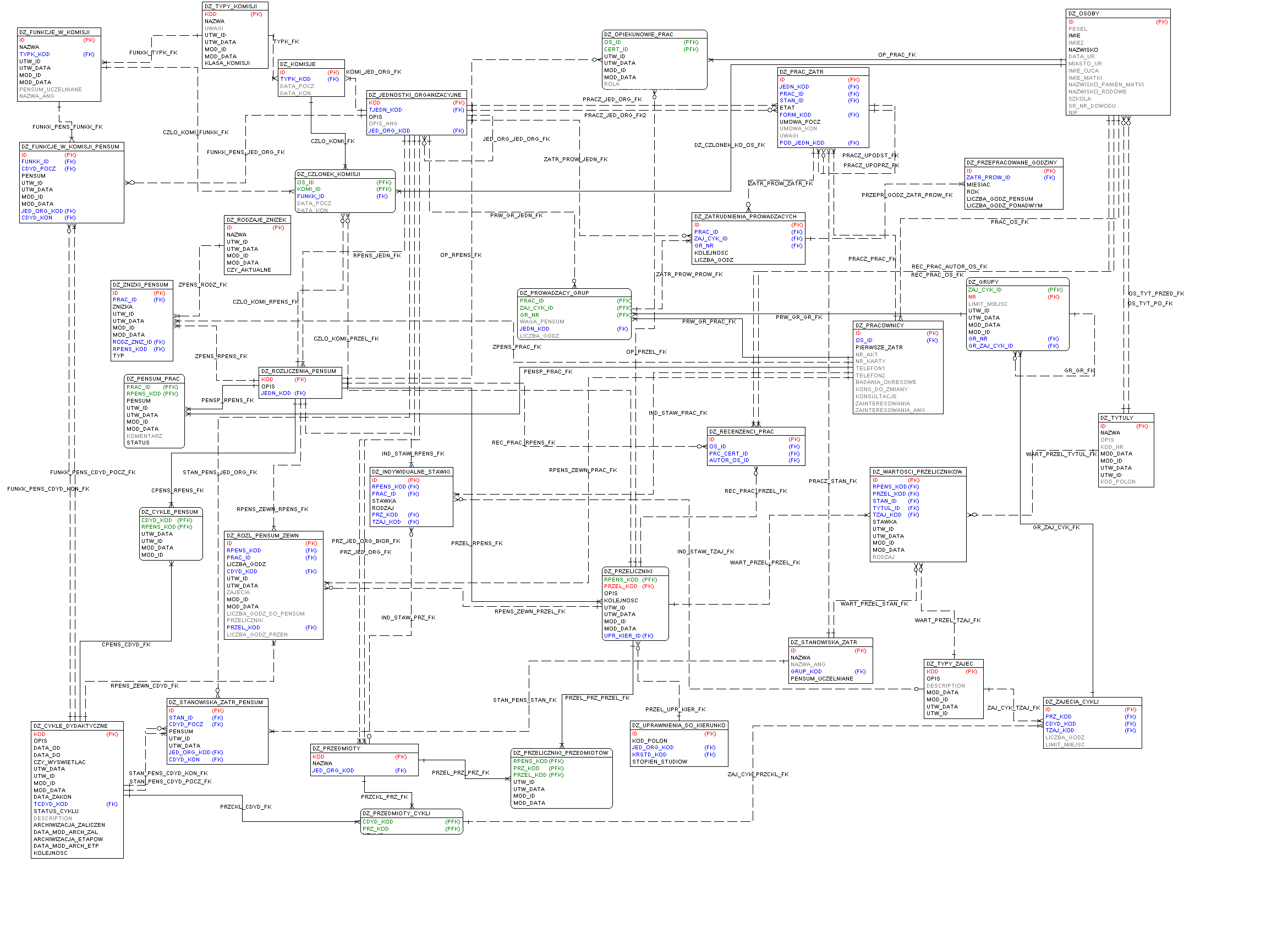
**Pensum bazowe jest dostosowywane przez indywidualne stawki (IndividualRates) i zniżki (Discount).**

### 1.1.3 Generowanie raportów:

**Dane eksportowane są do plików Excel z podziałem na:**

**Raport 1: Podsumowanie obciążeń pracowników (godziny dydaktyczne, nadgodziny, stawki).**

**Raport 2: Dane grup zajęciowych (np. limity miejsc, prowadzący).**



**"Rysunek 1 przedstawia model relacyjny bazy danych USOS wykorzystywany w systemie. Widoczne są kluczowe tabele, takie jak DZ\_PRACOWNICY, DZ\_GRUPY, oraz ich powiązania (relacje). Źródło: Opracowanie własne na podstawie analizy schematu bazy USOS."**

Schemat bazy danych (kluczowe tabele):

* DZ\_PRACOWNICY – pracownicy dydaktyczni.
* DZ\_GRUPY – grupy zajęciowe.
* DZ\_PROWADZACY\_GRUP – przypisania prowadzących do grup.
* DZ\_CYKLE\_DYDAKTYCZNE – semestry (np. "Rok akademicki 2023/2024").

## 1.2. Wymagania funkcjonalne i niefunkcjonalne

### 1.2.1. Wymagania funkcjonalne:

#### 1.2.1.1. Filtrowanie danych:

* + Wybór roku akademickiego i jednostki organizacyjnej (np. "Instytut Informatyki").
  + Przełączanie między widokiem "Grupy" a "Wykładowcy".

#### 1.2.1.2. Tryb edycji:

* + Modyfikacja przypisań prowadzących do grup przez przeciąganie elementów (drag-and-drop).

#### 1.2.1.3. Generowanie raportów:

* + Eksport do Excel z danymi obciążeń i grup.

#### 1.2.1.4. Autoryzacja:

* + Logowanie za pomocą poświadczeń USOS (integracja z tabelą DZ\_OSOBY).

### 1.2.2. Wymagania niefunkcjonalne:

#### 1.2.2.1. Wydajność:

* + Optymalizacja zapytań SQL (np. indeksowanie kolumn PRAC\_ID, ZAJ\_CYK\_ID).
  + Cache’owanie często używanych danych (np. listy jednostek).

#### 1.2.2.2. Bezpieczeństwo:

* + Walidacja danych wejściowych (np. sprawdzanie poprawności dat).
  + Ochrona przed SQL injection przez użycie ORM.

#### 1.2.2.3. Niezawodność:

* + Obsługa błędów połączenia z bazą danych (np. automatyczne ponowienia).

## 1.3. Projekt interfejsu użytkownika w PyQt5

**Interfejs został zbudowany w oparciu o bibliotekę PyQt5, wykorzystując następujące komponenty:**

### 1.3.1. Struktura okna głównego (MainWindow):

* + **Layout: Główny kontener QVBoxLayout z podziałem na:**
    - **Pasek filtrów: QHBoxLayout z comboboxami (QComboBox) do wyboru roku i jednostki.**
    - **Zakładki: QTabWidget z widokami "Grupy" i "Wykładowcy".**
    - **Listy: QListWidget dla grup i prowadzących.**
    - **Przyciski: QPushButton do generowania raportów.**

### 1.3.2. Funkcjonalności interaktywne:

* + **Drag-and-drop: W trybie edycji (QCheckBox) możliwe jest przeciąganie prowadzących między listami.**
  + **Dynamiczne ładowanie danych: Zmiana filtra roku/jednostki odświeża listę grup (populate\_groups()).**

### 1.3.3. Przykładowy fragment kodu:

### 1.3.4. Wizualizacja interfejsu:

### 1.3.5. Obsługa zdarzeń:

* + **generate\_report(): Eksportuje dane do Excel za pomocą pandas.DataFrame i openpyxl.**
  + **toggle\_edit\_mode(): Aktywuje/dezaktywuje tryb przeciągania.**

# **2.** Implementacja aplikacji desktopowej w PyQt5

## **2.1** Projektowanie okien aplikacji

## **2.2** Implementacja przeglądania obciążeń dydaktycznych

## **2.3** Kontrola dostępu na podstawie ról użytkowników

# **3.** Połączenie z bazą danych i logika biznesowa

## 3.1 Implementacja modelu danych w SQLAlchemy

## 3.2 Obsługa zapytań SQL

## 3.3 Obliczanie pensum dydaktycznego

# **4.** Eksport danych i generowanie raportów

## 4.1 Generowanie raportów Excel

## 4.2 Filtrowanie raportów według jednostek, lat akademickich i stanowisk

# **5. Testowanie,** bezpieczeństwo i wdrożenie

## 5.1 Testy funkcjonalne i integracja z aplikacją

## 5.2 Optymalizacja połączenia z bazą USOS

## 5.3 Wdrożenie aplikacji jako plik wykonywalny

# **6. Podsumowanie i wnioski**

# **7. Literatura**