Przetwarzanie danych w chmurach obliczeniowych
Dokumentacja projektu

Jan Zajda

24.12.2021

Link do działającej aplikacji: https://jan-zajda-clouds-project.herokuapp.com/

1. Temat projektu

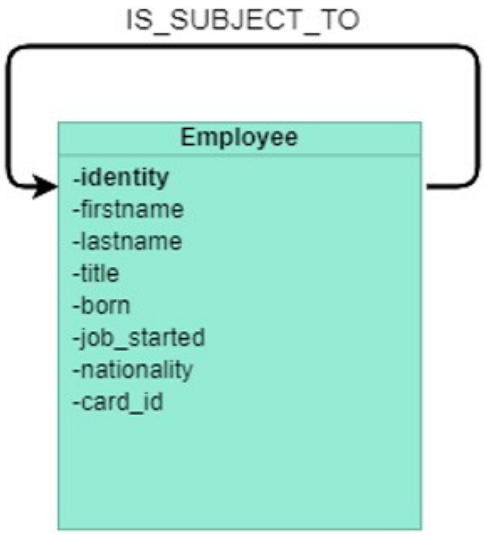
Celem projektu było stworzenie aplikacji do zarządzania organizacją pracowników firmy oraz możliwość sprawdzania zamieszczonych o pracownikach informacji w bazie danych.

2.Diagram UML

W grafowej bazie danych umieszczono węzły jednego typu “Employee” opisującej poszczególnych pracowników firmy. Każdy węzęł zawiera kilka podstawowych atrybutów każdego pracownika firmy:

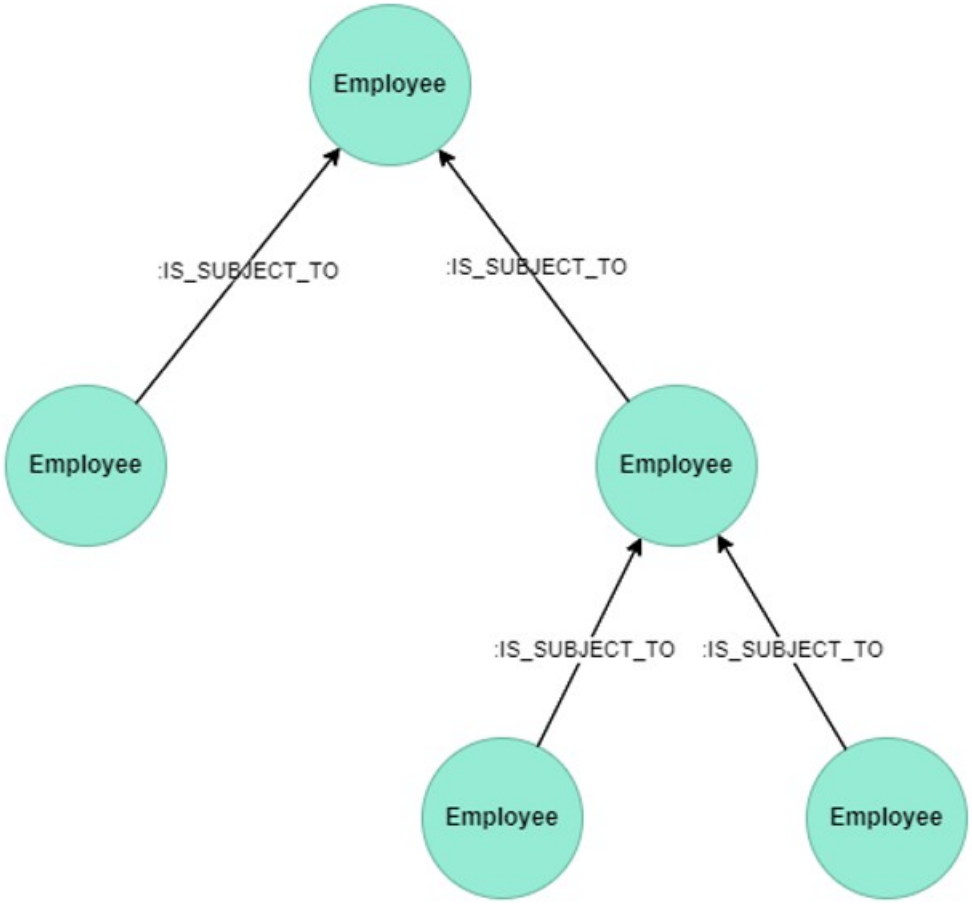
- firstname – imię pracownika
- lastname – nazwisko pracownika
- title – nazwę stanowiska w firmie
- born – rok urodzenia
- job_started – rok rozpoczęcia pracy w firmie
- nationality – kraj pochodzenia
- card_id – unikalny numer karty posiadanej przez każdego pracownika

Węzły połączone są przy pomocy relacji “:IS_SUBJECT_TO” obrazującej relację (podwładny-przełożony). Poniższy prosty diagram UML przedstawia opisane wyżej zależności.



Rys. 1: Diagram UML z użytym węzłem i relacją.

Przy pomocy tej klasy węzłów i relacji aplikacja buduje drzewo organizacyjne firmy, tak jak pokazano poniżej. Dbanie o spójność drzewa oraz zapobieganie tworzeniu cyklów leży w odpowiedzialności aplikacji.



Rys. 2: Przykładowe zrealizowanie w GBD hierarchii firmowej.

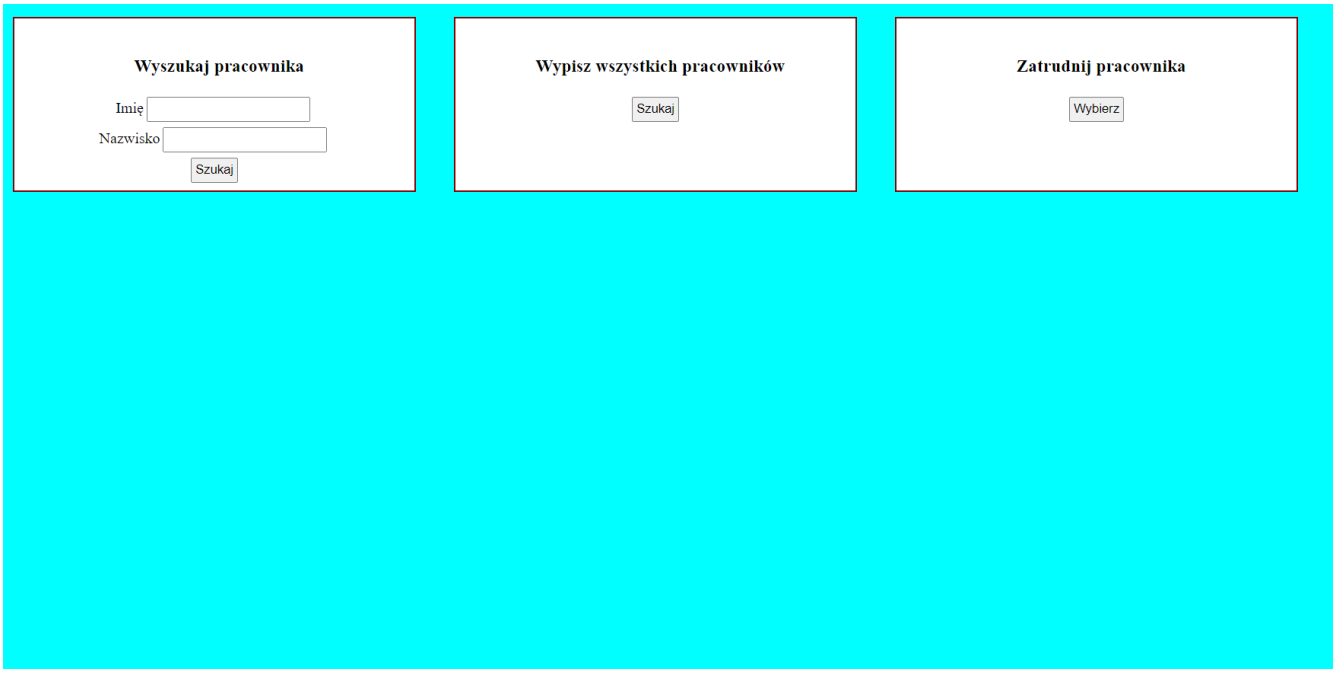
3. Opis aplikacji

W startowym panelu aplikacji mamy do dyspozycji trzy opcje.
Pierwsza pozwala na znalezienie pracownika po imieniu i nazwisku.
Druga daje możliwość zatrudnienia w firmie nowego pracownika.
W tym celu użytkownik otrzymuje formularz z danymi, które należy wypełnić.
Część z nich jest obligatoryjna jak np. nazwa stanowiska, lub wybór przełożonego z listy pracowników.
Trzecia pozwala na wyświetlenie listy wszystkich pracowników w kolejności alfabetycznej wraz z pakietem podstawowych danych na ich temat.

Po wciśnięciu imienia znalezionego pracownika, użytkownikowi wyświetla się panel informacji o pracowniku, z dodatkową listą bezpośrednich podwładnych oraz listą przełożonych w kolejności od najbliższego aż na sam szczyt firmowej hierarchii.

W tym panelu istnieje też możliwość:

- zmiany nazwy stanowiska zajmowanego przez pracownika
- zmiany przełożonego (z listy możliwych przełożonych)
- zwolnienia pracownika (ta opcja wymaga wcześniejszego przeniesienia podwładnych zwalnianego pracownika pod innych przełożonych)



Rys. 3: Panel główny aplikacji

4. Wykorzystane technologie

W aspekcie przechowywanie danych aplikacja korzysta z systemu zarządzania grafową bazą danych Neo4j AuraDB Free. Z poziomu aplikacji dokonywanych jest do bazy wiele zapytań dla różnych zadań (MATCH, CREATE, DELETE, SET). Sama aplikacja jest napisana w języku Python przy użyciu lekkiego framework'u Flask. Wykorzystuje ona REST API z dostępem do zasobów przy pomocy metod HTTP (GET, POST, PUT i DELETE). Aplikacja została rozlokowana na platformie chmurowej Heroku.

5. Krótki opis kodu źródłowego

W głównym katalogu projektu znajdziemy startowy plik aplikacji *clouds.py* oraz *employee.py* i *hire.py* odpowiadające za wyświetlanie danych pracownika i obsługę formularza zatrudnienia.
Najważniejszym plikiem jest jednak *connection.py*, w którym znajduje się klasa wykonująca wszystkie potrzebne zapytania do bazy danych. Ponadto w folderze *templates* znajdują się pliki *.html*, a w *static* proste pliki *.css* oraz *.js*.