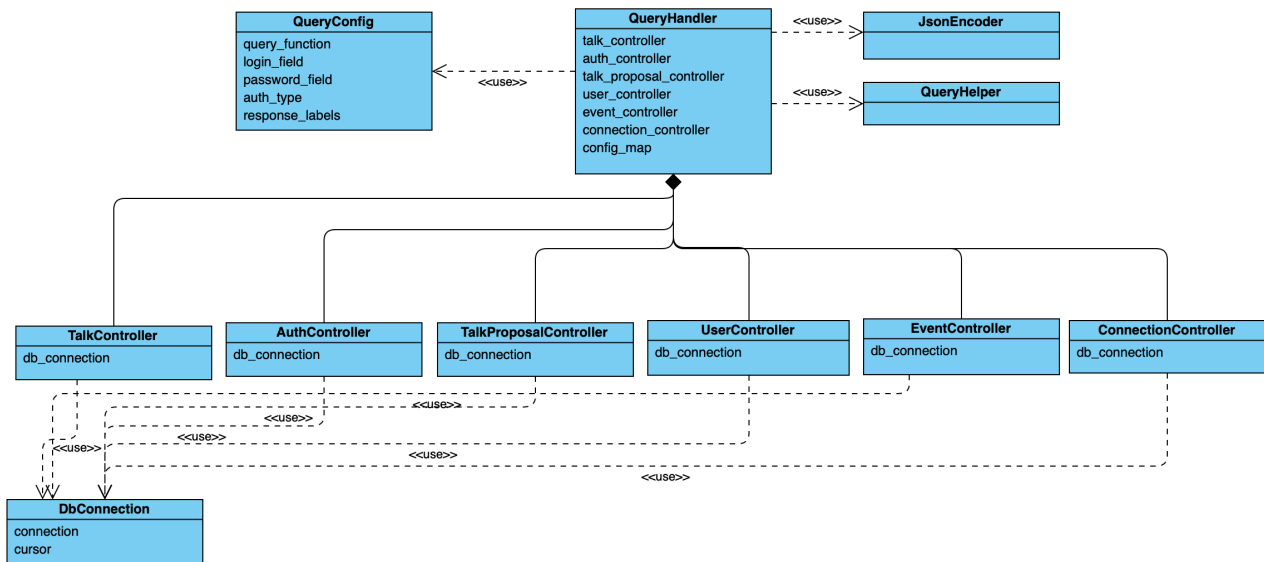


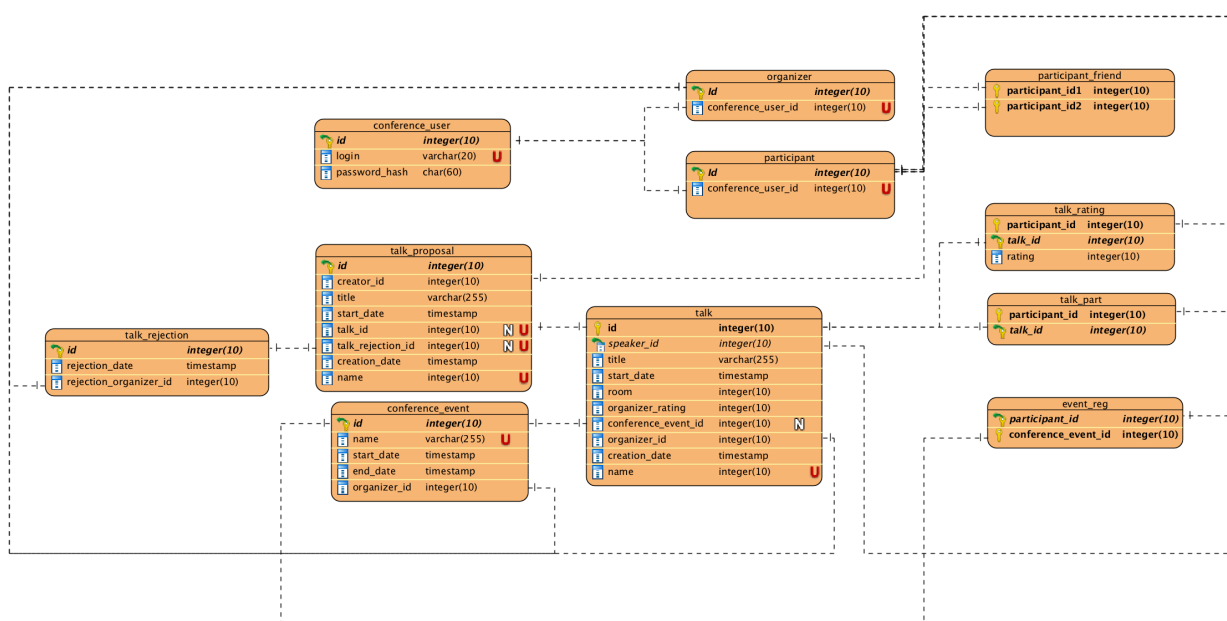
1. Opis klas

- JsonEncoder - klasa odpowiedzialna za enkodowanie JSON'a (dziedziczy po domyślnym enkoderze w celu dodawania możliwości enkodowania dat)
- DbConnection - klasa odpowiedzialna za komunikację z bazą danych: inicjowanie połączenia, wykonywanie zapytań i commitowanie transakcji
- QueryHelper - zawiera pomocnicze funkcje zwracające domyślne statusy zapytań
- QueryConfig - zawiera konfigurację zapytania, w tym informacje o sposobie autoryzacji, użytkownika oraz zwracanych kolumnach w zapytaniu
- QueryHandler - zajmuje się przechwytywaniem zapytań i odpowiadaniem na nie
- ConnectionController - kontroler odpowiedzialny za inicjowanie połączenia z bazą danych
- AuthController - kontroler odpowiedzialny za autoryzację użytkownika, który wysłał zapytanie
- UserController - kontroler odpowiedzialny za dodawanie nowych użytkowników oraz zarządzaniem ich danymi
- EventController - kontroler odpowiedzialny za tworzenie nowych wydarzeń i rejestrację użytkowników na już istniejące wydarzenia
- TalkController - kontroler odpowiedzialny za tworzenie wykładów, ich ocenę oraz listę uczestników, którzy są zapisani
- TalkProposalController - kontroler odpowiedzialny za zarządzanie propozycjami wykładów

2. Diagram UML klas



3. Diagram E-R bazy danych



4. Dodatkowe więzy i widoki

- przy dodawaniu lub zmianie w tabeli 'talk_proposal' sprawdzane jest za pomocą funkcji 'ensure_talk_proposal_status', czy 'talk_id' oraz 'talk_rejection_date' nie są ustawione jednocześnie na wartości niepuste. (oznaczałoby to, że referat został jednocześnie odrzucony i zatwierdzony)
- przy dodawaniu znajomości sprawdzane jest, czy nie została dodana znajomość ze samym Sobą za pomocą funkcji 'ensure_correct_friendship'
- funkcje 'add_organizer' oraz 'add_participant' pozwalają na utworzenie organizatora lub uczestnika (tworząc przy tym odpowiedniego użytkownika)
- funkcje 'remove_organizer' oraz 'remove_participant' pozwalają na usunięcie organizatora lub uczestnika (usuając przy tym odpowiedniego użytkownika)
- funkcje 'reject_talk_proposal' oraz 'confirm_talk_proposal' pozwalają odpowiednio odrzucić i zatwierdzić daną propozycję referatu
- widok 'participant_bilateral_friend' zapewnia abstrakcyjną tabelę, w której krotki oznaczają, że dane osoby są znajomymi

5. Opis ról i funkcjonalności

Oprócz administratora, który może dowolnie modyfikować stan bazy, istnieje jeszcze rola 'application_user'. Rola ta jest wykorzystywana przez aplikacje, które łączą się z bazą i ma na celu wyeliminowanie możliwości zmian tabel przez użytkownika, który nie powinien tego robić. Oprócz tego, że nie może on zmieniać schematów tabel, dodawać swoich triggerów, to nie wolno mu też dodawać, usuwać z tabel 'conference_user', 'organizer', 'participant', 'talk_rejection'. Zamiast tego powinien on używać odpowiednich funkcji, które zapewniają, że baza nie zostanie zmodyfikowana w nieodpowiedni sposób.

6. Sposób uruchomienia programu

Program należy uruchomić poleceniem `python db_conference.py` (dokładniej została wykorzystana wersja Pythona 2.7.13). Do uruchomienia są potrzebne biblioteki: `psycopg2`, `bcrypt`. Można je doinstalować poleceniami: `pip install psycopg2` oraz `pip install bcrypt` (w przypadku braku programu `pip` należy go zainstalować poleceniem `sudo apt-get install python-pip` na systemie Linux oraz `brew install pip` na systemie OS X). Został dodany też plik `run.sh`, który automatycznie instaluje potrzebne oprogramowanie wykonując powyższe komendy.