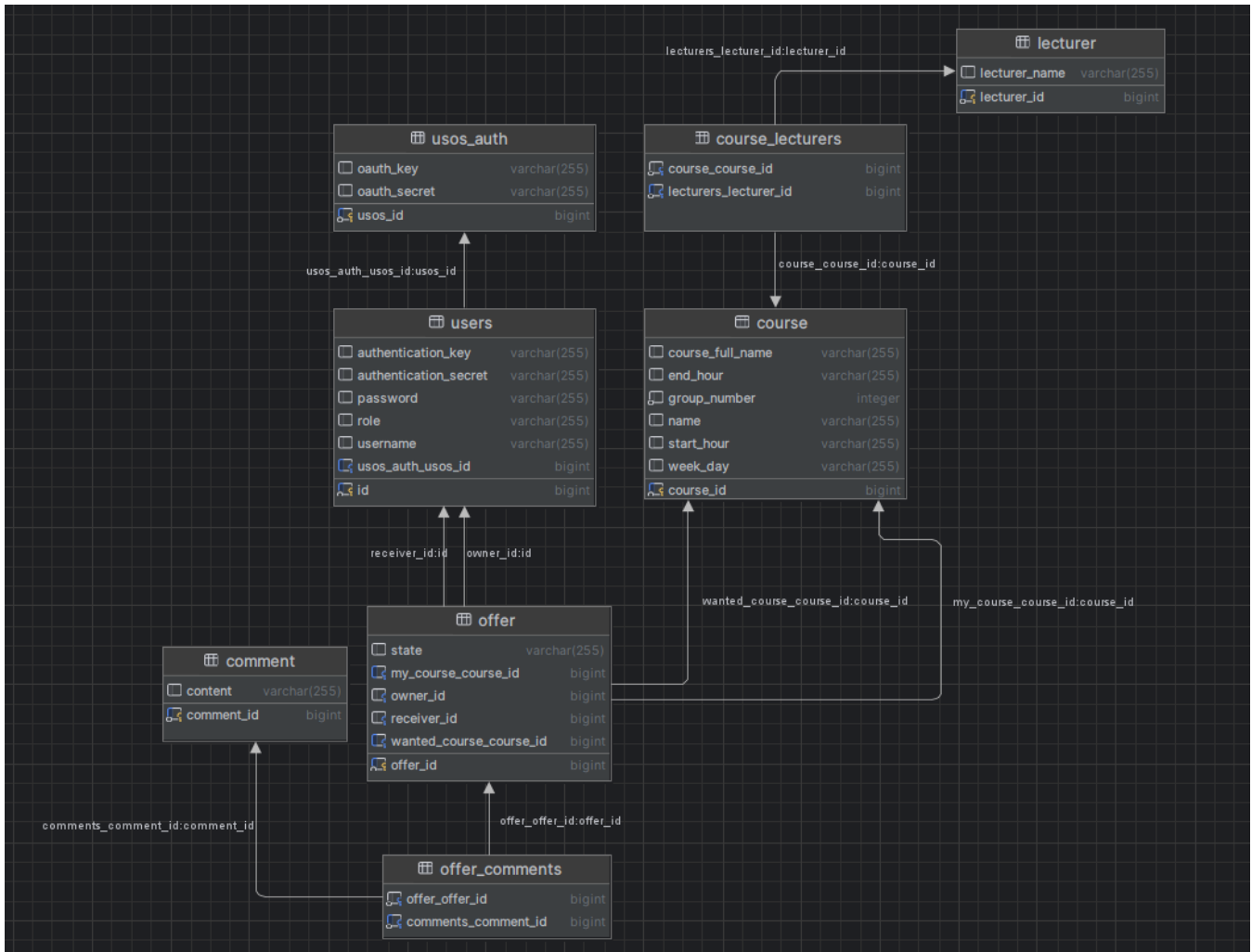


1. Listą wymagań funkcjonalnych i procentowy stopień ich realizacji:

- a. Użytkownik może stworzyć konto na naszym portalu. Ma dostępny przejrzysty formularz do rejestracji nowego konta. Użytkownik zostaje poinformowany w formularzu jeśli przy rejestracji podany przez niego login jest już zajmowany. Użytkownicy są zapisywani w naszej bazie danych z hashowanymi hasłami. Stopień realizacji funkcjonalności: 100%
- b. Użytkownik może połączyć swoje konto z istniejącym kontem studenckim USOS. Przy wykorzystaniu USOS api oraz security springboot 3 daliśmy możliwość połączenia z USOS i uzyskiwania informacji uczelnianych. Klucze identyfikacyjne są trzymane w rekordach o danych użytkowników w bazie. Jeśli USOS „pamięta” danego użytkownika nie jest wymagane ponowne logowanie. Stopień realizacji funkcjonalności: 100%
- c. Użytkownik może zalogować się na swoje istniejące konto na naszym portalu. Ma dostępny przejrzysty formularz do logowania. Użytkownik zostaje poinformowany w formularzu jeśli podał złe dane do logowania. Stopień realizacji funkcjonalności: 100%
- d. Użytkownik może wyświetlić pomoc do aplikacji (help). Użytkownik po zalogowaniu/zarejestrowaniu się oraz po połączeniu konta z USOS ma dostęp do strony głównej. Tam może wejść na prostą podstronę wyświetlającą „poradnik” jak korzystać z naszej aplikacji. Stopień realizacji funkcjonalności: 100%
- e. Użytkownik może zmienić swój login. Użytkownik ma dostęp do swojego profilu. Jest opcja zmienić login. Gdy użytkownik próbuje zmienić login na taki który już istnieje w bazie danych zostaje o tym poinformowany a akcja zostaje zablokowana. Stopień realizacji funkcjonalności: 100%
- f. Użytkownik może zmienić swój awatar. Użytkownik ma dostęp do swojego profilu. Jest opcja zmiany awatara. Stopień realizacji funkcjonalności: 100%
- g. Użytkownik może zobaczyć wszystkie swoje zajęcia. Użytkownik ma dostęp do zakładki kursy w którym widzi swoje obecne zajęcia. Jest to indywidualna informacja pobierana z USOS-a dla każdego użytkownika. Stopień realizacji funkcjonalności 100%.
- h. Użytkownik może stworzyć ofertę wymiany. Użytkownik ma możliwość stworzenia nowej oferty wymiany. Ma do wyboru swoje przedmioty. Nie może wymieniać np. laboratorium za ćwiczenia. Nie może wymieniać wykładów. Nie może wybrać innej grupy niż jest mu przypisana w USOS. Nie może stworzyć więcej niż jednej oferty wymiany do danego przedmiotu. Po stworzeniu oferta pojawia się na rynku widocznym dla innych użytkowników. Stopień realizacji funkcjonalności 100%.
- i. Użytkownik w zakładce moje oferty ma możliwość wycofania swojej oferty wymiany z rynku. Znika ona z rynku oraz jest usuwana z bazy danych. Stopień realizacji funkcjonalności 100%.
- j. Użytkownik może zaakceptować ofertę wymiany. Użytkownik ma możliwość akceptacji oferty wymiany gdy ktoś na rynku postanowi złożyć prośbę o jego ofertę. Musi dojść do obustronnej akceptacji aby oferta wymiany była wiążąca. Gdy oferta zostaje zaakceptowana trafia do sekcji zakończone wymiany gdzie można skomunikować się z partnerem. Stopień realizacji funkcjonalności 100%.

- k. Użytkownik może odrzucić ofertę wymiany. Użytkownik ma możliwość odrzucenia oferty wymiany gdy ktoś na rynku postanowi złożyć prośbę o jego ofertę ale nie ma ochoty z nim jej finalizować. Musi dojść do obustronnej akceptacji aby oferta wymiany była wiążąca. Stopień realizacji funkcjonalności 100%.
- l. Użytkownik zostaje poinformowany o pozytywnym przebiegu wymiany. Użytkownik widzi zakończone wymiany w odpowiedniej zakładce. Ma możliwość komunikacji z drugim studentem. Stopień realizacji funkcjonalności 100%.
- m. Użytkownik może przeglądać oferty wymiany innych osób. Użytkownik ma dostęp do rynku na którym są wystawione oferty innych osób szukających wymiany. Przez odpowiednie filtry użytkownik widzi tylko oferty dotyczące jego przedmiotów. Widzi również własne wystawione oferty odpowiednio oznaczone. Stopień realizacji funkcjonalności 100%.
- n. Użytkownik może filtrować oferty po przedmiocie (np. szukamy tylko ofert dotyczących przedmiotu Programowanie aplikacyjne). Na rynku z ofertami można filtrować wyniki tylko do takich które nas interesują. Stopień realizacji funkcjonalności 100%.
- o. Użytkownik może filtrować oferty po dacie dodania (np. szukamy ofert najnowszych); Użytkownik ma możliwość filtracji ofert po dacie dodania, ma do tego odpowiednie przyciski. Stopień realizacji funkcjonalności 100%.
- p. Użytkownik może filtrować oferty po godzinie (np. szukamy ofert tylko na poniedziałek 12:15. Na rynku z ofertami można filtrować wyniki tylko do takich które nas interesują. Stopień realizacji funkcjonalności 100%.
- q. Użytkownik może filtrować oferty po prowadzącym (np. chcemy zdobyć danego prowadzącego). Na rynku z ofertami można filtrować wyniki tylko do takich które nas interesują. Stopień realizacji funkcjonalności 100%.
- r. Użytkownik może dodawać komentarze do zaakceptowanych wymian. Gdy oferta przejdzie całe flow (od wystawienia, do akceptacji z obu stron) jest wyświetlana w zakładce zakończone oferty. Użytkownik ma możliwość komunikacji z partnerem wymiany przez dodawania komentarzy (jest to forma czatu). Dzięki temu studenci mogą się ze sobą skomunikować aby „dogadać” finalizację wymiany. Stopień realizacji funkcjonalności 100%
- s. Użytkownik może zobaczyć wszystkie swoje przeszłe wymiany. Widać je w zakładce zakończone wymiany. Tam są czaty zapisane i można tam zawsze zajrzeć. Stopień realizacji funkcjonalności 100%

2. Opis zrealizowanej aplikacji z opisem wykorzystywanych technologii, schematem bazy danych (diagram ER – Entity Relationship Diagram): Projekt zrealizowano od strony backend przy użyciu framework do Javy: springboot 3. Strona klienta jest zbudowana przy użyciu biblioteki do JavaScript – React, HTML + CSS. Cały serwer został postawiony na nginx. Konteneryzacja z użyciem docker + baza danych Postgresql.



3. Przykład udokumentowanego modułu:
- UserController
  - AuthenticationCotroller
  - UserService
  - AuthenticationService
  - Model: UsosAuth, User, Role, Avatar, AuthenticationRequest, AuthenticationResponse, AvatarRequest, UserDataResponse, UsernameRequest