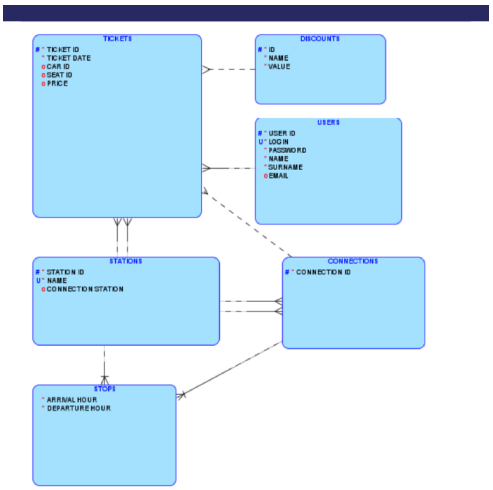
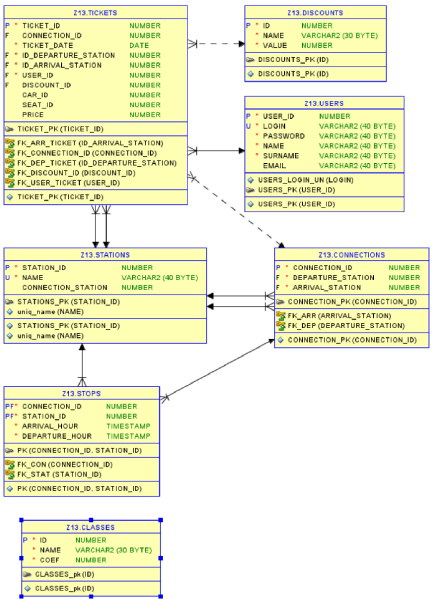
PROJEKT BD1

- **model ER**



- **model relacyjny**



Funkcjonalność aplikacji z poziomu administratora i użytkownika

- **Administrator:**
  - **Logowanie do panelu administratora**
    Uruchamia panel administratora umożliwiający podjęcie działań opisanych poniżej.
  - **Zarządzanie stacjami**
    Dodaje, edytuje lub usuwa stację (operacje na bazie danych).
  - **Dodawanie nowych połączeń**
    Dodaje do bazy danych nowe połączenie, wymaga podania stacji początkowej i końcowej oraz przystanków pośrednich wraz z godzinami przyjazdu i odjazdu, automatycznie dodaje połączenie powrotne.
  - **Usuwanie istniejących połączeń**
    Usuwa istniejące połączenie.
  - **Usuwanie i edytowanie istniejących kont użytkowników**
    Usuwa oraz edytuje istniejące konto użytkownika.
  - **Zarządzanie dostępnymi klasami oraz zniżkami**
    Pozwala na dodawanie, edytowanie i usuwanie klas oraz zniżek dostępnych dla użytkownika podczas zakupu biletu.

- **Użytkownik:**
  - **Tworzenie nowego konta użytkownika**
    Umożliwia użytkownikowi założenie konta poprzez podanie swoich danych (unikalny login, imię, nazwisko, adres email oraz hasło), spowoduje to dodanie do tabeli USERS nowego rekordu.
  - **Logowanie do istniejącego konta**
    Pozwala użytkownikowi na zalogowanie się do aplikacji, dane użytkownika zostają wtedy zapamiętane i możliwy jest imienny zakup biletu.
  - **Wyszukiwanie dostępnych połączeń**
    Możliwe jest wyszukiwanie połączenia o podanym przystanku początkowym i końcowym. Jeśli nie zostanie znalezione połączenie bezpośrednie, wyszukane zostanie połączenie z przesiadką.
  - **Wybór połączenia**
    Pozwala użytkownikowi wybrać konkretne połączenie z listy dostępnych połączeń. Po akceptacji zostaje przekierowany do panelu zakupu biletu.
  - **Zakup biletu**
    Możliwy jest zakup biletu poprzez wybranie miejsca (spośród dostępnych miejsc w postaci rozwijanej listy), klasy (I lub II) oraz ulgi (brak ulgi, studenci, pracownicy, osoby niepełnosprawne). Na tej podstawie obliczana jest cena biletu. Po zakupie biletu pojawi się on w panelu MY ACCOUNT dostępnym z poziomu menu głównego. Będzie tam też dostępna historia przejazdów użytkownika.

Struktura bazy danych:

Baza składa się z 7 tabel, a ich dokładny opis znajduje się poniżej

- Connections - rekordy tabeli reprezentują połączenia, każde połączenie ma swoje id, id stacji początkowej i id stacji końcowej.
  Kluczami obcymi są id stacji początkowej i końcowej.
- Stations - rekordy tabeli reprezentują stacje, każda stacja ma swoje id, nazwę oraz binarną wartość oznaczającą stację przesiadkową (0 - stacja nie jest przesiadkowa, 1 - stacja jest przesiadkowa).
- Stops - rekordy tabeli reprezentują przystanki, każdy przystanek ma id połączenia, id stacji, godzinę przyjazdu oraz godzinę odjazdu.
  Kluczami obcymi są: id połączenia i id stacji.
- Users - rekordy tabeli reprezentują użytkowników, każdy użytkownik ma swoje id, login, hasło (hasło jest szyfrowane), imię, nazwisko oraz email (email jest opcjonalny)
- Tickets - rekordy tabeli reprezentują bilety, każdy bilet ma swoje id, id połączenia, datę, id stacji początkowej i id stacji końcowej (jako stację początkową i końcową rozumiemy stacje, na których dany bilet jest ważny), id użytkownika, id zniżki, id wagonu, id miejsca oraz cenę.
  Kluczami obcymi są: id stacji początkowej i końcowej, id połączenia, id użytkownika i id zniżki.
- Discounts - rekordy reprezentują zniżki, każda zniżka ma swoje id, nazwę oraz wartość
- Classes - rekordy reprezentują klasy, każda klasa ma swoje id, nazwę oraz współczynnik ceny

Wykorzystane funkcje:

- timestamp_diff - funkcja przyjmuje czas początkowy i czas końcowy w formacie TIMESTAMP i zwraca różnicę między nimi w minutach.
- travel_time - funkcja przyjmuje id połączenia, id stacji początkowej i id stacji końcowej (wszystkie w formacie NUMBER) i zwraca czas przejazdu między stacjami w minutach według danego połączenia.

Wykorzystane triggery:

- tg_check_name_station - trigger przed dodaniem nowej stacji sprawdza, czy stacja o tej nazwie już istnieje w bazie, jeżeli tak, rzuca błąd
- tg_check_station_in_stops - trigger przed usunięciem istniejącej stacji sprawdza, czy ze stacją powiązane są przystanki, jeżeli tak, rzuca błąd
- tg_del_stops_and_tickets - trigger przed usunięciem istniejącego połączenia usuwa powiązane z nim przystanki i bilety
- tg_del_user_tickets - trigger przed usunięciem użytkownika usuwa powiązane z nim bilety

Wykorzystana procedura:

- print_station_connections - procedura przyjmuje id stacji, a następnie wyświetla możliwe połączenia z tej stacji w postaci:
  “nazwa stacji, nazwa stacji końcowej, godzina odjazdu”.

Skrypty:

Skrypty wymagane w wymaganiach dotyczących projektu są umieszczone są w repozytorium projektu na wydziałowym gitlabie.

Analiza rozwiązania:

Projekt posiada ograniczenia w postaci czasu dostępu do danych oraz złożoności obliczeniowej w algorytmie poszukiwania przesiadek obecnym bezpośrednio w aplikacji w języku Java.
Baza danych może zostać rozbudowana o dodatkowe tabele i funkcjonalności, takim jak rozbudowany system wyboru miejsc w postaci schematu wagonu wraz z oznaczeniem zajętości miejsc.

Autorzy projektu:
Filip Browarny
Rafał Budnik
Krzysztof Kluczyński
Jakub Kowalski