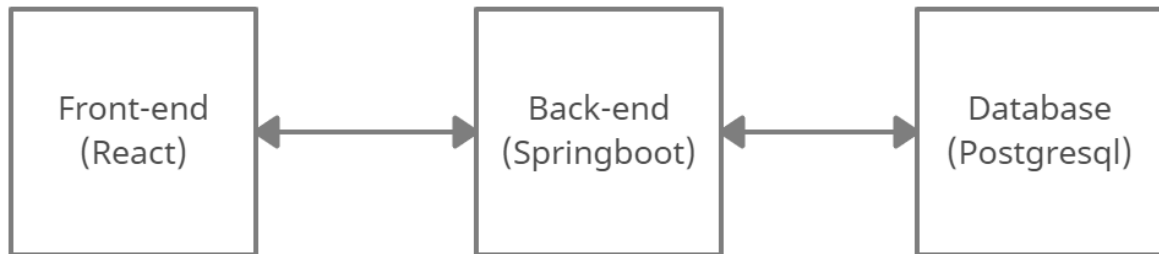


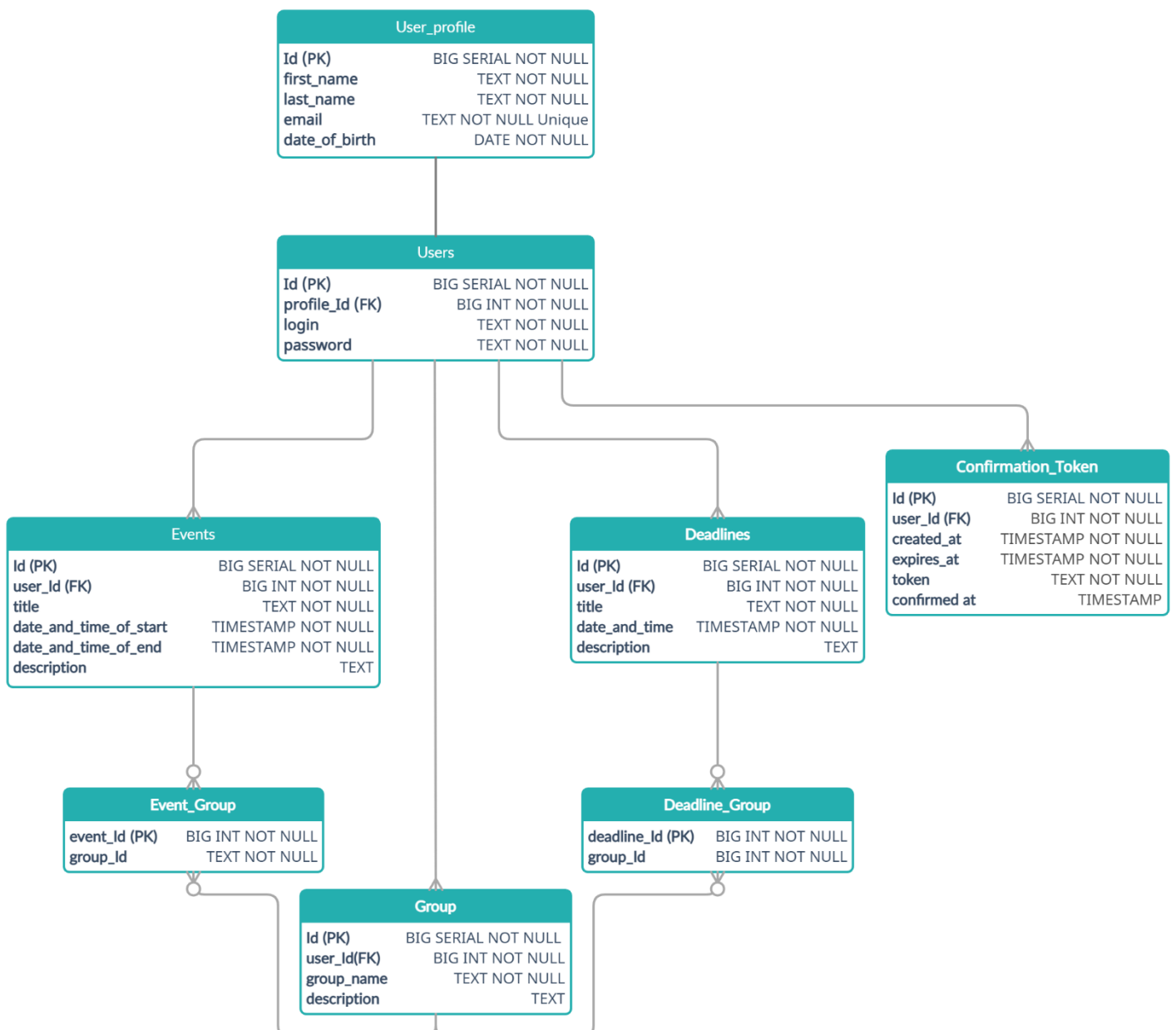
# Dokumentacja

Autorzy: Pavel Klimuk(backend), Stanislau Stankevich(frontend)

Główne warstwy aplikacji:



Struktura bazy danych:



## Wykorzystane frameworki i biblioteki:

### Backend:

- Spring Boot
- Spring Security
- Json Web Token

### Frontend

- ReactJS
- LuxonJS (biblioteka do pracy z datą i czasem)

## Wykorzystywane narzędzia:

### Backend:

- IntelliJ IDEA
- Postman

### Frontend:

- VS Code
- Chrome DevTools + Chrome React extension
- Postman

## Główne klasy (Backend):

- UserProfile
- User
- Deadline
- Event
- Group

## Główne komponenty (Frontend):

- Calendar -> Day -> DayProperties (reprezentacja kalendarzu)
- Events/Deadlines/Groups (listy zdarzeń/deadlinów/grup)
- AddNew(Event/Deadline/Group) (formy do dodawania/edycji)
- UserCreate/Login (formy do tworzenia konta/logowania)
- Header/NavBar (nawigacja, pasek górny aplikacji)

## Główne serwisy (Frontend):

- (Deadlines/Events/Groups/User)Service - http zapytania do API
- auth/headers - przechwytywanie i przechowywanie tokenu Autoryzacji

## Wymagania środowiskowe:

### Backend:

Java: 11

Spring Boot: 2.4.4 lub nowsza

Maven

PostgreSQL: 13.2 lub nowsza

### Frontend:

Node JS

## Instrukcja uruchomienia:

Sklonować repozytorium: <https://github.com/pklimuk/planner.git>

### PostgreSQL:

Uruchomić skrypt **init.sql**, by stworzyć lokalną bazę danych z odpowiednimi ustawieniami.

### Backend:

Uruchomić plik `PlannerApplication.java`.

### Frontend:

Zmienić branch na 'frontend'

W katalogu głównym wpisać **npm install** żeby zainstalować potrzebne moduły node.

W tym samym katalogu **npm start**, przeglądarka sama się uruchomi.