**ПРОЕКТИРОВАНИЕ ГЛАВНОГО СЕРВЕРА**

Главный сервер представляет собой стек из двух контейнеров – бекэнд приложения ASP.NET и базы данных PostgreSQL.

Теперь перейдем к бекэнд приложению. Исходя из стоящих перед приложением задач, оно должно иметь следующие контроллеры: сущности пользователя, аутентификации, статуса серверов, проверки прав, подключения к VPN-серверу. Контроллер сущности пользователя должен поддерживать CRUD операции, а равно действия GET (получить информацию о пользователе), PATCH (отредактировать информацию о пользователе), PUT (добавить нового пользователя), DELETE (удалить пользователя). Контроллер аутентификации должен поддерживать действия GET (проверка действительности JWT-токена), POST (аутентификация пользователя с получение JWT-токена), PATCH (получить новый JWT-токен на основании действительного). На этапе проектирования не принято решение об использовании refresh-токенов, таковое будет принято на этапе прототипирования, однако пока-что будем закладывать таковую возможность.

Рассмотрим вопрос структуры базы данных. Поскольку для приложения предполагается использование EntityFramework, в действительности, мы будем рассматривать структуру моделей данных бекэнд приложения. Сущность User представляется следующими полями: id (long, PK), email\_address (varchar), email\_confirmed (bool, AK), tariff (enum int), paid\_until (timestamp), refresh\_tokens (Array<refresh\_token> = stuct[] {token\_key (varchar), valid\_until (timestamp)}), trusted\_ips (Array<int>). Сущность AdminUser представляется следующим полем: id (long, PK, FK from User.id). Сущность LastSendedEmail представляется полями: id (long, PK, FK from User.id), lastEmailSendedTime (timestamp). Также возможно создание сущности AnonymousUser, использующей вместо адреса почты уникальный идентификатор, по которому пользователь может выполнить вход в аккаунт. Утеря такого идентификатора означает утрату доступа к аккаунту.