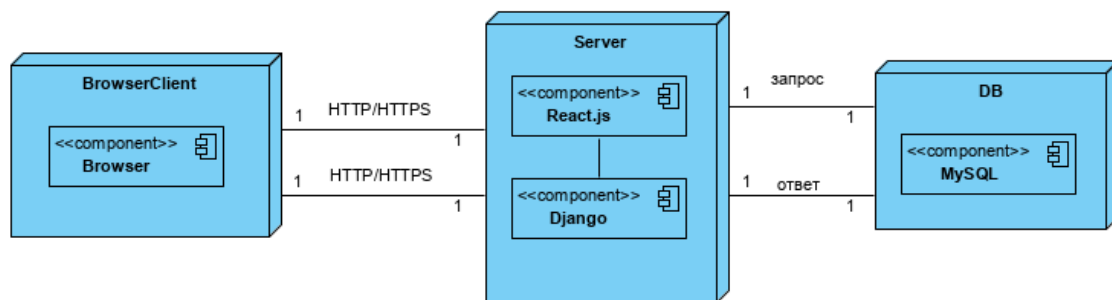


СТО

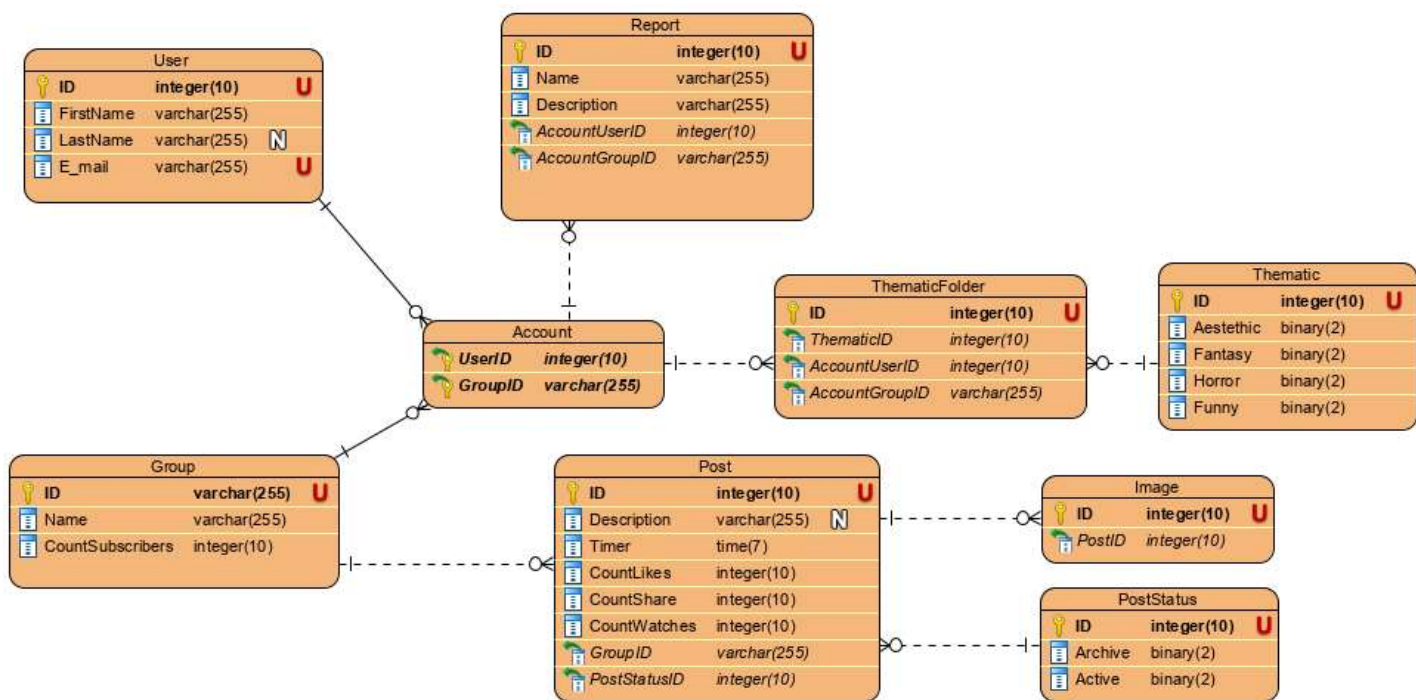
Реализация всех задумок СРО в виде продуманной архитектуры проекта

Структура проекта описана на диаграмме развертывания:

- BrowserClient - клиентская часть проекта, которая отображается в виде странички в браузере. Через HTTP/HTTPS запросы взаимодействует с серверной частью.
- Server - серверная часть проекта, которая состоит из взаимодействий React.js и Django. Через запросы общается с хранилищем данных.
- DB - часть хранилища данных, где содержатся все сущности проекта и информация. Состоит из СУБД MySQL.



Архитектура хранилища данных



Таблицы:

- User - хранит в себе необходимую информацию о пользователе, которые использует данный сервис.
- Group - хранит в себе информацию о сообществе, которым управляет пользователь.
- Account - представляет из себя аккаунт пользователя, который управляет сообществом(-ами) и, соответственно, содержащий о них всю информацию.
- Report - представляет из себя некий отчет, который включает в себя всю статистику по постам сообщества, которым управляет пользователь.
- ThematicFolder - представляет модель тематической папки, в которую можно отнести пост.
- Thematic - является перечислением тематик.
- Post - представляет из себя пост, который публикует пользователь в свое сообщество.
- Image - хранит в себе картинку, которую в дальнейшем прикрепляют к посту.

- PostStatus - является перечислением статусов поста.

Выбор технологий для проекта с обоснованиями по направлениям: backend, frontend, хранение данных, инфраструктура, API (для каждой позиции сравнить не менее 3 технологий)

Frontend:

Язык программирования: JavaScript.

Почему не PHP? После прочтения данной статьи: <https://ru.bitdegree.org/rukovodstvo/php-ili-javascript/> станет понятно, что и PHP, и JavaScript - популярные языки с низкой точкой входа, отлично подходящие для frontend разработки. Но поскольку на JS опыт работы уже имеется небольшой, то приоритет был отдан в итоге ему.

Фреймворк для разработки frontend'a: React.js

Поскольку он всегда находится в топе по мнению других разработчиков:

https://pikabu.ru/story/luchshie_frontend_freymvorki_dlya_vebrazrabotki_v_2021_godu_8000606

React разработчики могут с высокой скоростью создавать высокопроизводительные приложения, несмотря на уровень их сложности. Возможность с легкостью заново использовать уже имеющийся код повышает скорость разработки, упрощает процесс тестирования, и, как результат, понижает затраты.

Backend:

Язык программирования: Python

Почему не PHP?

Данная статья поможет определиться с тем, почему Python лучше подходит для разработки backend части: <https://zen.yandex.ru/media/futureinapps/10-prichin-pochemu-v-vebrazrabotke->

программирования на python уже имеется достаточно большой, поэтому в итоге выбор пал на него.

Фреймворк для разработки backend'a: Django

В каких случаях лучше всего подходит Django? Когда:

- Требуется разработать веб-приложение или серверную часть API.
- Требуется быстро работать, быстро разворачивать и вносить изменения в проект по ходу работы.
- В любой момент в приложении может потребоваться масштабирование: как наращивание, так и сокращение.
- В перспективе планируется интегрировать новейшие технологии, например, машинное обучение.
- Нужно использовать надежный фреймворк, который активно разрабатывается, используется многими топовыми компаниями и ведущими веб-сайтами во всем мире.
- Требуется, чтобы и веб-приложение, и серверная часть API находились в одной и той же базе кода, согласовываясь с «единым источником истины» (принцип DRY)
- Не хотите работать непосредственно с запросами к базе данных, и вам нужна поддержка ORM.
- Собираетесь пользоваться свободным ПО.
- Если застрянете – то решение придется искать самостоятельно, поэтому вам понадобится хорошая документация и отзывчивое сообщество разработчиков.

DB: MySQL

Почему не PostgreSQL?

Основные преимущества MySQL от PostgreSQL:

- **Скорость и надежность:** отказавшись от некоторых функций SQL, система MySQL сохранила легкость, отдавая приоритет скорости и надежности. Ее скорость особенно очевидна, когда речь заходит о высокопараллельных операциях без записи в базе данных (только чтение). Это отличный выбор для определенных приложений бизнес-аналитики. Но если вам нужно выполнить много сложных запросов под большой нагрузкой, то база данных PostgreSQL может справиться лучше.
- **Варианты оптимизации сервера MySQL:** Предлагается множество вариантов настройки и оптимизации вашего MySQL database server путем настройки переменных, таких как `sort_buffer_size`, `read_buffer_size`, `max_allowed_packet` и так далее.
- **Простота в использовании и популярность:** популярность базы данных MySQL означает, что будет несложно найти администраторов баз данных с большим опытом работы с этой СУБД. Пользователи говорят, что эта система проще в настройке, то есть не требует такой тонкой настройки, как другие СУБД. Настроить свою первую базу данных MySQL легко даже новичку. Установка и настройка PostgreSQL будет сложнее.

Релизы

Будет использоваться GitLab, поскольку он предлагает автоматическую интеграцию, которая означает, что каждый раз, когда в проект добавляется новый код, он перестраивает модель проекта и тестирует его проблемы учитывая нововведения. Это и называется непрерывной интеграцией или же CI.

Поскольку основная целевая аудитория на данный момент - пользователи социальной сети VK, которая проживает в России, то релизы будут выпускаться ночью ~00:00-03:00 по МСК.

Скрипт проведения собеседования

- Рассказываем о своей компании: чем занимаемся, чем гордимся, кого ищем и что от него ждем.
- Человек рассказывает о себе, в каких проектах участвовал, в чем силен, чем может пригодиться компании.
- Задаем уточняющие вопросы по резюме, если оно есть. Например, о качествах человека, опыте работы и тд.
- Определив должность, на которую претендует человек, задаем технические вопросы по теме, фиксируя ответы.
- Даем небольшие задачки, чтобы оценить уровень навыков в реальности, либо тестовое задание, которое выполнить можно у себя дома.
- Заканчиваем собеседование и анализируем собранные данные.