Geekbrains

# Вэб — разработка. Создание бекэнд — сервиса работающего с мессенджером Telegram для создания и оптимизации графика занятий или встреч, с дальнейшей возможностью экспорта в календарь с графической оболочкой.

## Программист Python

Алексанин Максим Сергеевич

2023 год

# Содержание

1. Содержание

2. Введение

2.1 Описание

2.2 Цель проекта

2.3 Задачи по реализации проекта

2.4 Состав команды

2.5 Использумеые инструменты и технологии

3. Теоретическая и практическая главы

4. Заключение

5. Список литературы

6. Приложения

# Введение

## Описание

Проект представляет собой веб-приложение, а точнее Телеграм-бот(далее приложение) работающий в изолированной среде, на удаленном сервере с внешним IP и нацелено в первую очередь на преподователей, репетиторов, ведущих онлайн — занятия или встречи. Приложение не имеет своего графического интерфейса, и общается с пользователем через мессенджер посредством API. Приложение сохраняет данные от пользователя в реляционную базу данных, работающую на том же, либо отдельном сервере. Так же прилоение сохраняет статистику для последующего просмотра пользователем.

## Цель проекта

Приложение должно оптимально рапределять время пользователя для занятий или встреч, сохранять данные участников, вести статистику, вести рассылку (уведомления) участникам. На основе обратной связи приложения пользователь должен видеть загруженность дней, недель и количество свободного времени.

## Задачи

- Выделить сущности и составить схему их взаимоотношений

- Определить форматы данных таких как время и идентификаторы

- Составить API приложения

- Составить алгоритм распределения и индикации времени

- Построить приложение

- Запуск базы данных и приложения

- Тестирование приложения

- Рефакторинг

**Состав команды**

Алексанин Максим Серегеевич(разработчик, тестировщик, DB-дизайнер)

**Инструменты и технологии**

OpenAPI, Telegram, SQLite, Git, VPS, Docker virtualization, Python

# 3. Проектирование и реализация

## 3.1 Проектирование

### 3.1.1 Основные сущности базы данных и приложения

*User —* пользователь, сущность имеющая уникальный идентификатор UUID формата base32hex описанный документе RFC 2938. Пользовател может быть как организатором так и участником события.

*Event —* событие, основная сущность приложения, которой оперирует пользовтель. Событие имеет уникальный идентификатор UUID описанный в документе RFC 2938, который может быть составлен по алгоритму описанному в документе RFC 4122 и множество других полей, самые значимые из которых является **start** и **end** — временное представление начала и конца события, на основе которого происходят вычисления загруженности и свободного времени пользователя. Данные поля имеют аттрибут **datetime** описанный в документе RFC 3339 и соответствует формату ISO 8601. События должны итерироваться, иметь индикацию длительности, при наличии поля **end**. События должны храниться в базе данных.

*Day* — день, сущность содержащая множество событий. Должна итерироваться и иметь индикацию загруженности. Может храниться в базе данных. Пользователь должен иметь возможность определить время в течении которого он доступен для событий, а так же время перерывов.

*Week* — неделя, сущность содержащая дни. Должна итерироваться и иметь индикацию загруженности. Может храниться в базе данных. На неделе могут быть определены дни, которые не доступны для событий.

### 3.1.2 Принципы и алгоритмы вычисления и индикация загруженности

Индикация загруженности дня должна вычисляться на основе оставшегося свободного времени текущего дня. Индикация загруженности недели может основываться на загруженности дней. Загруженность дня должна кэшироваться либо сохраняться в базе жанных, чтобы избежать повторных вычислений.