

# Техническое задание: Система управления студенческими группами

## 1. Общее описание проекта

### 1.1 Назначение системы

Telegram Web App для управления студенческими группами, включающее функции организации событий, дедлайнов, выбора тем для занятий и управления очередями на защиту работ.

### 1.2 Целевая аудитория

- Студенты (участники групп)
- Старосты групп
- Помощники старост
- Администраторы системы

### 1.3 Технологический стек

- Backend: PostgreSQL
- Frontend: Telegram Web App
- Авторизация: Telegram OAuth

## 2. Система авторизации и ролей

### 2.1 Процесс авторизации

#### 2.1.1 Первый вход через Telegram-бота

1. Авторизация через Telegram
2. Ввод ФИО пользователем
3. Выбор группы из списка
4. Старосте выбранной группы приходит уведомление с предложением принять студента
5. После подтверждения старостой пользователь добавляется в группу

#### 2.1.2 Первый вход по ссылке-приглашению от старосты

1. Переход по уникальной ссылке-приглашению
2. Авторизация через Telegram
3. Ввод ФИО пользователем
4. Автоматическое добавление в группу

### **2.1.3 Вход администраторов**

- Осуществляется по специальной ссылке
- Доступен только команде разработчиков (3 человека)
- Фиксированные учетные записи администраторов

## **2.2 Система ролей**

### **2.2.1 Администратор**

#### **Права:**

- Доступ ко всем группам в роли старосты
- Просмотр любой информации в системе
- Супер-права для управления системой

#### **Ограничения:**

- Только 3 администратора в системе
- Доступ по специальной ссылке

### **2.2.2 Староста группы**

#### **Права:**

- Принятие/отклонение заявок на вступление в группу
- Создание и отправка ссылок-приглашений
- Выгон участников из группы
- Назначение помощников старосты
- Создание, редактирование и удаление событий
- Составление списков тем для занятий
- Управление выбором тем (удаление выбора, назначение темы)
- Создание и редактирование дедлайнов
- Управление очередями на защиту
- Удаление группы

#### **Ограничения:**

- Только один староста на группу
- Может состоять только в одной группе

### **2.2.3 Помощник старосты**

**Права:**

- Создание и редактирование событий
- Создание дедлайнов и редактирование их сроков
- Составление списков тем на занятия
- Изменение количества человек на одну тему
- Редактирование количества защит на одну пару

**Ограничения:**

- Не может выгонять участников
- Не может принимать заявки в группу
- Не может назначать других помощников
- Количество помощников < количества участников группы

**2.2.4 Участник группы****Права:**

- Просмотр событий и информации группы
- Выбор тем из доступного списка
- Занятие места в очереди на защиту
- Добавление личных напоминаний
- Просмотр информации о выбранных темах других участников
- Просмотр очереди на защиту

**Ограничения:**

- Может состоять только в одной группе
- Не может создавать/редактировать события

**3. Структура базы данных****3.1 Основные таблицы****3.1.1 Users (Пользователи)**

sql

- id (PK) - уникальный идентификатор пользователя
- telegram\_id - ID пользователя в Telegram (**UNIQUE**)
- telegram\_username - никнейм в Telegram
- first\_name - имя
- last\_name - фамилия
- middle\_name - отчество (NULLABLE)
- created\_at - дата создания аккаунта
- last\_active\_at - дата последней активности

### 3.1.2 Groups (Группы)

sql

- id (PK) - уникальный идентификатор группы
- name - название группы
- description - описание группы (NULLABLE)
- created\_at - дата создания

### 3.1.3 GroupMembers (Участники групп)

sql

- id (PK) - уникальный идентификатор записи
- user\_id (FK) - ссылка на пользователя
- group\_id (FK) - ссылка на группу
- is\_leader - флаг старосты группы
- is\_assistant - флаг помощника старосты
- joined\_at - дата вступления в группу

### 3.1.4 GroupInvitations (Приглашения в группы)

sql

- id (PK) - уникальный идентификатор приглашения
- group\_id (FK) - ссылка на группу
- invited\_by\_user\_id (FK) - кто отправил приглашение
- invite\_token - уникальный токен приглашения
- expires\_at - срок действия приглашения
- is\_used - использовано ли приглашение
- created\_at - дата создания

### 3.1.5 Events (События)

sql

- id (PK) - уникальный идентификатор события
- group\_id (FK) - ссылка на группу
- created\_by\_user\_id (FK) - кто создал событие
- title - название события
- description - описание
- subject - предмет/дисциплина
- deadline\_id (FK) - ссылка на дедлайн (NULLABLE)
- order\_id (FK) - ссылка на очередь (NULLABLE)
- topics\_id (FK) - ссылка на темы (NULLABLE)
- is\_important - важное ли событие
- created\_at - дата создания

### 3.1.6 ViewedEvents (Просмотренные события)

sql

- id (PK) - уникальный идентификатор записи
- user\_id (FK) - ссылка на пользователя
- event\_id (FK) - ссылка на событие
- viewed\_at - дата просмотра

### 3.1.7 Deadlines (Дедлайны)

sql

- id (PK) - уникальный идентификатор дедлайна
- event\_id (FK) - ссылка на событие
- description - описание
- start\_date - дата начала
- end\_date - срок сдачи

### 3.1.8 Queues (Очереди)

sql

- id (PK) - уникальный идентификатор очереди
- event\_id (FK) - ссылка на событие
- title - название очереди
- description - описание
- max\_participants - максимальное количество участников
- created\_at - дата создания

### 3.1.9 QueueParticipants (Участники очередей)

sql

- id (PK) - уникальный идентификатор записи
- queue\_id (FK) - ссылка на очередь
- user\_id (FK) - ссылка на пользователя
- position - позиция в очереди
- joined\_at - дата присоединения

### 3.1.10 Topics (Темы)

sql

- id (PK) - уникальный идентификатор темы
- event\_id (FK) - ссылка на событие
- title - название темы
- description - описание
- max\_participants - максимальное количество участников
- created\_at - дата создания
- created\_by\_user\_id (FK) - кто создал тему

### 3.1.11 TopicSelections (Выбор тем)

sql

- id (PK) - уникальный идентификатор записи
- topic\_id (FK) - ссылка на тему
- user\_id (FK) - ссылка на пользователя
- selected\_at - дата выбора
- is\_confirmed - подтвержден ли выбор (старостой)
- confirmed\_by\_user\_id (FK) - кто подтвердил выбор
- confirmed\_at - дата подтверждения

## 3.2 Административные таблицы

### 3.2.1 AdminLogs (Логи действий админов)

sql

- id (PK) - уникальный идентификатор записи
- telegram\_id (FK) - ссылка на пользователя-админа
- action - действие
- details - детали действия
- performed\_at - дата выполнения
- ip\_address - IP адрес

### 3.2.2 AdminLogins (Входы админов)

sql

- id (PK) - уникальный идентификатор записи
- telegram\_id (FK) - ссылка на пользователя-админа
- login\_at - дата входа
- ip\_address - IP адрес

### 3.3 Дополнительные таблицы

#### 3.3.1 StarredGroups (Отмеченные группы)

sql

- id (PK) - уникальный идентификатор записи
- user\_id (FK) - ссылка на пользователя
- group\_id (FK) - ссылка на группу
- starred\_at - дата отметки

#### 3.3.2 Reminders (Напоминания)

sql

- user\_id (PK, FK) - ссылка на пользователя
- notify\_before\_deadline - за сколько минут предупреждать
- notify\_new\_topics - уведомления о новых темах (BOOLEAN)
- notify\_new\_queues - уведомления о новых очередях (BOOLEAN)
- notify\_new\_events - уведомления о новых событиях (BOOLEAN)
- notify\_method - способ уведомления
- updated\_at - дата последнего обновления настроек

#### 3.3.3 ChangeLog (Журнал изменений) - опциональная

sql

- id (PK) - уникальный идентификатор записи
- user\_id (FK) - кто внес изменения
- entity\_type - тип изменяемой сущности
- entity\_id - ID изменяемой сущности
- action - действие (создание, изменение, удаление)
- old\_values - предыдущие значения (JSON)
- new\_values - новые значения (JSON)
- changed\_at - дата изменения

## 4. Функциональные требования

### 4.1 Раздел "События"

#### **4.1.1 Для старост и помощников старост**

##### **Интерфейс:**

- Кнопка "Добавить событие"
- Лента событий с возможностью прокрутки
- Меню из 3 точек для каждого события (редактировать/удалить)

##### **Функционал:**

- Создание событий с текстом, фото или видео
- Редактирование существующих событий
- Удаление событий
- Отметка событий как важные (показываются сверху)
- Привязка дисциплины к событию с автопредложением

#### **4.1.2 Для участников**

##### **Интерфейс:**

- Лента событий (только просмотр)
- Кнопка "Просмотрено" для каждого события
- Подразделы: "Актуальные" и "Просмотренные"

##### **Функционал:**

- Просмотр событий
- Отметка событий как просмотренных
- Фильтрация по статусу просмотра

#### **4.2 Календарь и бронирование**

##### **4.2.1 Общий интерфейс**

- Календарная сетка
- Дни с событиями помечены цветом и цифрой (количество событий)
- При клике на день без событий: "Событий нет"
- При клике на день с событиями: список событий с деталями

##### **4.2.2 Для старост и помощников старост**

##### **Функционал:**

- Кнопка "Добавить бронь"



- Интерактивное выделение доступных дней
- Создание событий на выбранный день
- Настройка списка тем с ограничениями по количеству участников
- Прикрепление изображений к событиям
- Редактирование и удаление существующих броней
- Управление выбором тем участниками (только для старост)

#### **4.2.3 Для участников**

##### **Функционал:**

- Просмотр событий на выбранный день
- Выбор темы из выпадающего списка
- Выбор места в очереди на защиту
- Просмотр информации о выбранных темах других участников
- Просмотр очереди на защиту

### **4.3 Система уведомлений**

#### **4.3.1 Настройки уведомлений**

- Время предупреждения о дедлайнах (в минутах)
- Уведомления о новых темах (вкл/выкл)
- Уведомления о новых очередях (вкл/выкл)
- Уведомления о новых событиях (вкл/выкл)
- Способ уведомления (Telegram, email)

#### **4.3.2 Типы уведомлений**

- Напоминания о приближающихся дедлайнах
- Уведомления о новых событиях в группе
- Уведомления о новых доступных темах
- Уведомления о новых очередях на защиту

### **4.4 Управление группами**

#### **4.4.1 Создание группы**

- Название группы
- Описание группы (опционально)
- Автоматическое назначение создателя старостой

#### **4.4.2 Управление участниками**

- Просмотр списка участников
- Принятие/отклонение заявок на вступление
- Исключение участников (только староста)
- Назначение помощников старосты
- Создание ссылок-приглашений

#### **4.4.3 Ссылки-приглашения**

- Уникальный токен для каждой ссылки
- Срок действия ссылки
- Одноразовое использование
- Отслеживание использования

### **5. Нефункциональные требования**

#### **5.1 Безопасность**

- Авторизация через Telegram OAuth
- Уникальные токены для приглашений
- Логирование действий администраторов
- Контроль доступа на основе ролей

#### **5.2 Производительность**

- Автоматическое удаление неактивных аккаунтов (1 год неактивности)
- Оптимизация запросов к базе данных
- Кэширование часто запрашиваемых данных

#### **5.3 Удобство использования**

- Интуитивный интерфейс
- Адаптивная верстка для мобильных устройств
- Быстрая загрузка страниц
- Отзывчивый интерфейс

#### **5.4 Масштабируемость**

- Поддержка множественных групп
- Возможность расширения функционала
- Модульная архитектура

## **6. Ограничения и правила**

### **6.1 Ограничения системы**

- Пользователь может состоять только в одной группе
- Только один староста на группу
- Количество помощников старосты < количества участников
- Срок действия ссылок-приглашений ограничен
- Одноразовое использование ссылок-приглашений

### **6.2 Бизнес-правила**

- Автоматическое удаление неактивных аккаунтов через 1 год
- Важные события отображаются сверху ленты
- Подтверждение выбора темы старостой (опционально)
- Возможность отмены выбора темы старостой

## **7. Этапы разработки**

### **7.1 Первая очередь (MVP)**

1. Система авторизации и ролей
2. Основные таблицы базы данных
3. Управление группами и участниками
4. Базовый функционал событий
5. Простой календарь

### **7.2 Вторая очередь**

1. Система уведомлений
2. Выбор тем и очереди
3. Административная панель
4. Расширенный функционал событий

### **7.3 Третья очередь (опционально)**

1. Система файлов
2. Журналирование изменений
3. Расширенная аналитика
4. Дополнительные функции

## **8. Технические детали**

### **8.1 Архитектура**

- Клиент-серверная архитектура
- RESTful API
- Telegram Web App как фронтенд
- PostgreSQL как основная база данных

### **8.2 Интеграции**

- Telegram Bot API для авторизации
- Telegram Web App API для интерфейса
- Система уведомлений через Telegram

### **8.3 Развертывание**

- Контейнеризация приложения
- Автоматические бэкапы базы данных
- Мониторинг производительности
- Логирование ошибок

## **9. Риски и митигация**

### **9.1 Технические риски**

- Ограничения Telegram Web App API
- Производительность при большом количестве пользователей
- Безопасность данных пользователей

### **9.2 Пользовательские риски**

- Сложность интерфейса для неопытных пользователей
- Конфликты между участниками группы
- Злоупотребление правами старост

### **9.3 Митигация**

- Тщательное тестирование всех функций
- Реализация системы логирования и аудита
- Создание подробной документации для пользователей
- Возможность отката критических изменений

