

**Требования к ИС «Походы выходного дня»**

**Составитель:**Шушарин А.А.

# Функциональное назначение системы

Система предназначена для организации походов выходного дняреализуется в видесайта. Пользователи могут регистрироваться в системе как организаторы и размещать информацию о походах. Прочие пользователи получают информацию из ленты и системы поиска,в случае заинтересованности могут получить подробную информацию о походе и подать заявки на участии в событии. Организатор может либо отклонить, либо подтвердить заявку. Регистрация организаторов премодерируемая (для исключения мошенничества). Пользователи могут выставить оценку и написать отзыв организатору. Есть элементы соцсети - возможность дружить с другими пользователями и просматривать список событий в которых они участвуют. Система должна поддерживать масштабируемость (в т.ч. потенциальную разработку мобильного приложения).

# Термины и сокращения

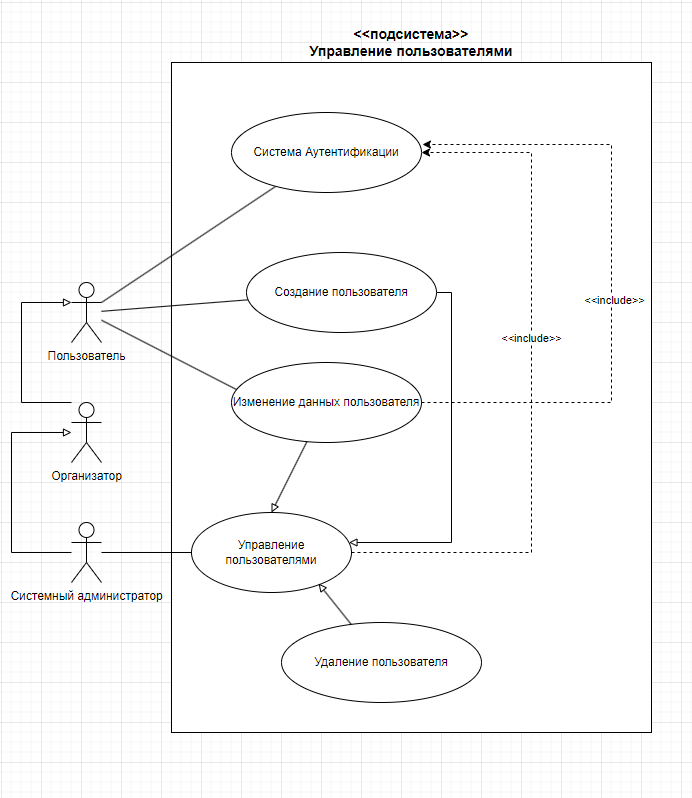
|  |  |
| --- | --- |
| **Организатор/организация** | Организация (в т.ч. физическое лицо), проводящая события |
| **Событие** | Поход выходного дня или аналогичные мероприятия |
| **Премодерация** | Сохранение изменений в системе (добавление, изменение, удаление) только после проверки администратором. Администратор либо подтверждает изменения, либо отклоняет. |
| **Карточка события** | Страница с подробной информацией о событии |
| **Карточка организатора** | Страница с подробной информацией об организаторе |
| **Личный кабинет организатора** | Раздел сайта позволяющий организатору управлять своими событиями |

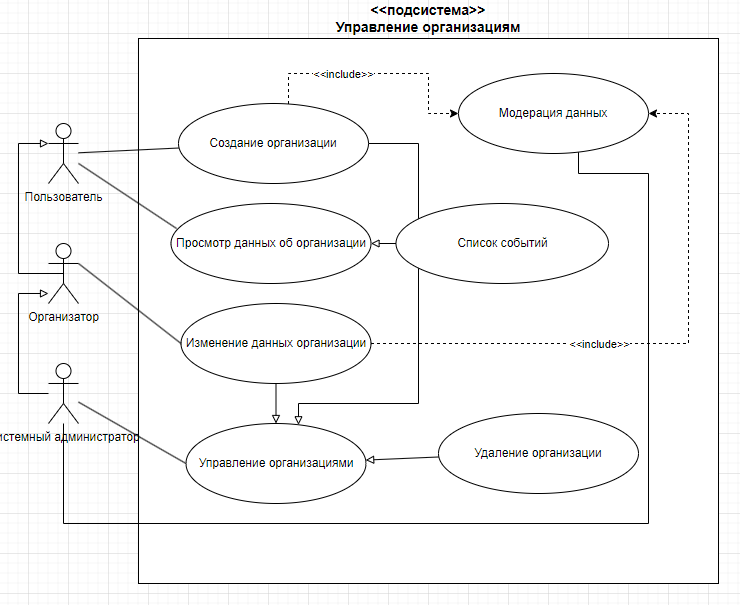
# Функциональные требования

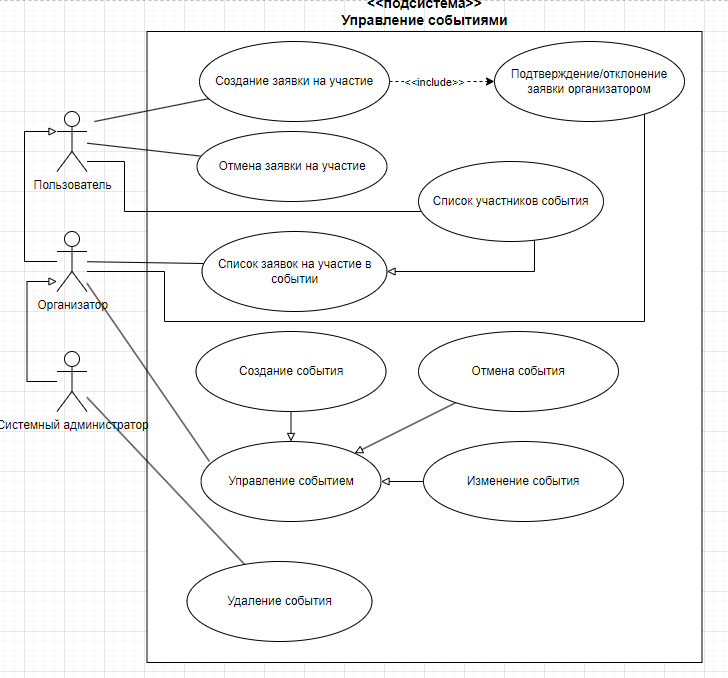
## Список требований

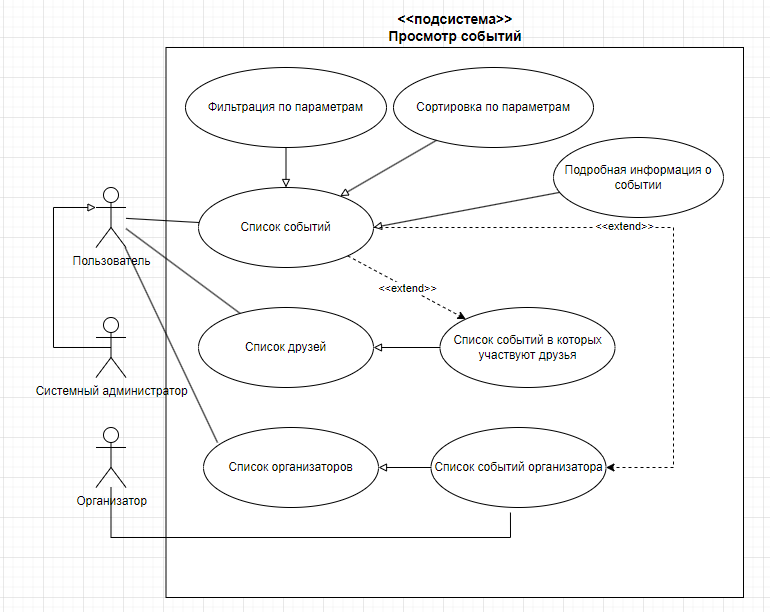
|  |  |
| --- | --- |
| **FR-01** | **Система должна предоставлять функционал аутентификации пользователей и управления пользователями** |
| **FR-01-01** | **Система должна предоставить возможность авторизации пользователей** |
| **FR-01-02** | **Система должна предоставить возможность пользователю зарегистрироваться в системе и управлять своей записью** |
| FR-01-02-01 | Пользователь должен иметь возможность зарегистрироваться |
| FR-01-02-02 | Пользователь должен иметь возможность просмотреть свои персональные данные |
| FR-01-02-03 | Пользователь должен иметь возможность изменить свои персональные данные |
| **FR-02** | **Система должна предоставлять пользователю возможность регистрации и управления организацией** |
| **FR-02-01** | **Система должна предоставить пользователю возможность зарегистрировать организацию** |
| FR-02-01-01 | Пользователь должен иметь возможностьзарегистрировать организацию |
| FR-02-01-02 | Система должна предоставить администратору возможность премодерации новых организаций |
| **FR-02-02** | **Система должна предоставить пользователю возможность изменить данные о своей организацию** |
| FR-02-02-01 | Пользователь должен иметь возможностьизменить данные своей организации |
| FR-02-02-02 | Система должна предоставить администратору возможность премодерации изменения данных организации |
| **FR-02-03** | **Система должна предоставить организатору возможность просматривать информацию о своей организации** |
| FR-02-03-01 | Организатор должен иметь возможность просмотреть информацию о своей организации |
| FR-02-03-02 | Организатор должен иметь возможность просмотреть список своих событий в хронологическом порядке |
| FR-02-03-02 | Организатор должен иметь возможность просмотреть все оценки организации с указанием пользователей |
| **FR-03** | **Система должна предоставлять организации возможностьуправления собственными событиями** |
| **FR-03-01** | Система должна обеспечить организатору возможность управлениями событием |
| FR-03-01-01 | Организатор должен иметь возможность создать событие |
| FR-03-01-02 | Организатор должен иметь возможность изменить информацию о событие |
| FR-03-01-03 | Организатор должен иметь возможность отменить событие |
| FR-03-01-04 | Организатор должен иметь возможность просмотра события |
| **FR-03-02** | Система должна обеспечить организатору возможность управлениям заявками на участие в событии |
| FR-03-02-01 | Организатор должен иметь возможность просмотра заявок на участие в событии |
| FR-03-02-02 | Организатор должен иметь возможность отклонить заявку на участие в событии |
| FR-03-02-03 | Организатор должен иметь возможность подтвердить заявку на участие в событии |
| FR-03-02-04 | Организатор должен иметь возможность посмотреть подтвержденные заявки на участие (участников мероприятия) |
| **FR-04** | **Система должна предоставлять возможность пользователям взаимодействовать с событиями** |
| **FR-04-01** | **Система должна предоставить пользователю возможность просмотра событий** |
| FR-04-01-01 | Пользователь должен иметь возможность просмотреть подробную информацию о событии |
| FR-04-01-02 | Пользователь должен иметь возможность просмотреть ленту событий, публикуемую в хронологическом порядке |
| FR-04-01-03 | Пользователь должен иметь возможность просмотреть отфильтрованный список событий |
| FR-04-01-03-01 | Пользователь должен иметь возможность отфильтровать события по дате |
| FR-04-01-03-02 | Пользователь должен иметь возможность отфильтровать события по стоимости |
| FR-04-01-03-03 | Пользователь должен иметь возможность отфильтровать события по количеству участников |
| FR-04-01-03-04 | Пользователь должен иметь возможность отфильтровать события по продолжительности похода |
| FR-04-01-03-05 | Пользователь должен иметь возможность отфильтровать события по рейтингу организатора |
| FR-04-01-03-06 | Пользователь должен иметь возможность отфильтровать события по наличию свободных мест |
| FR-04-01-03-07 | Пользователь должен иметь возможность отфильтровать события по организатору |
| FR-04-01-04 | Пользователь должен иметь возможность отсортировать события по любому полю |
| FR-04-01-04-01 | Пользователь должен иметь возможность отсортировать события по дате |
| FR-04-01-04-02 | Пользователь должен иметь возможность отсортировать события по стоимости |
| FR-04-01-04-03 | Пользователь должен иметь возможность отсортировать события по количеству участников |
| FR-04-01-04-04 | Пользователь должен иметь возможность отсортировать события по продолжительности похода |
| FR-04-01-04-05 | Пользователь должен иметь возможность отсортировать события по рейтингу организатора |
| **FR-04-02** | **Система должна предоставить пользователю возможность принять участие в событии** |
| FR-04-02-01 | Пользовательдолжен иметь возможность зарегистрироваться в событии |
| FR-04-02-02 | Система должна осуществлять проверку количества зарегистрировавшихся в событии участников и накладывать ограничение на регистрацию новых участников если достигнут максимум. |
| FR-04-02-03 | Пользователь должен иметь возможность отметить свою регистрацию в событии |
| FR-04-02-04 | Система должна отправлять уведомление пользователю о подтверждении или отклонения заявки на участие организатором |
| **FR-05** | **Система должна предоставлять возможность пользователям «дружить»** |
| **FR-05-01** | **Система должна предоставить пользователю возможность управлять дружбой** |
| FR-05-01-01 | Пользователь должен иметь возможность подать заявку дружбы |
| FR-05-01-02 | Пользователь должен иметь возможность отозвать заявку дружбы |
| FR-05-01-03 | Пользователь должен иметь возможность подтвердить заявку в друзья |
| FR-05-01-04 | Пользователь должен иметь возможность отклонить заявку в друзья |
| FR-05-01-05 | Пользователь должен иметь возможность удалить другого пользователя из друзей |
| FR-05-01-06 | Пользователь должен иметь возможность просмотреть список друзей |
| **FR-05-02** | **Система должна предоставить пользователю возможность посмотреть события друзей** |
| FR-05-02-01 | Пользователь должен иметь возможность просмотреть ленту, в которой будут указаны события, в которых зарегистрировались друзья |
| FR-05-02-02 | Пользователь должен иметь возможность просмотреть список друзей зарегистрированных в событии при просмотре события |
| **FR-06** | **Система должна предоставлять возможность оценки организаторов** |
| **FR-06-01** | **Система должна предоставить пользователю написать отзыв на организатора** |
| FR-06-01-01 | Пользователь должен иметь возможность добавить отзыв на организатора |
| FR-06-01-02 | Пользователь должен иметь возможность исправить свой отзыв на организатора |
| FR-06-01-02 | Пользователь должен иметь возможность удалить свой отзыв на организатора |
| FR-06-01-02 | Пользователь должен иметь возможность увидеть отзывы других пользователей при просмотре информации об организаторе |
| **FR-06-02** | **Система должна предоставить пользователю возможность поставить оценку организатору** |
| FR-06-02-01 | Пользователь должен иметь возможность поставить оценку организатору (по 10 бальной шкале) |
| FR-06-02-02 | Пользователь должен иметь возможность увидеть среднюю оценку организатора |

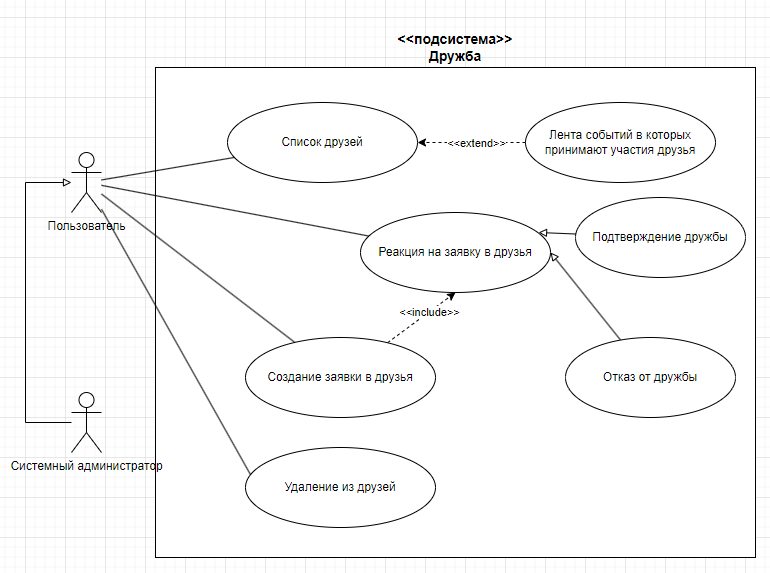
## Функциональная схема

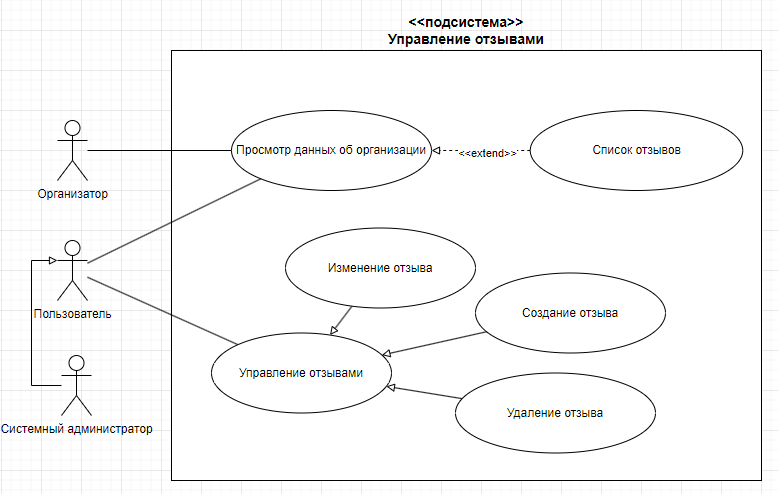


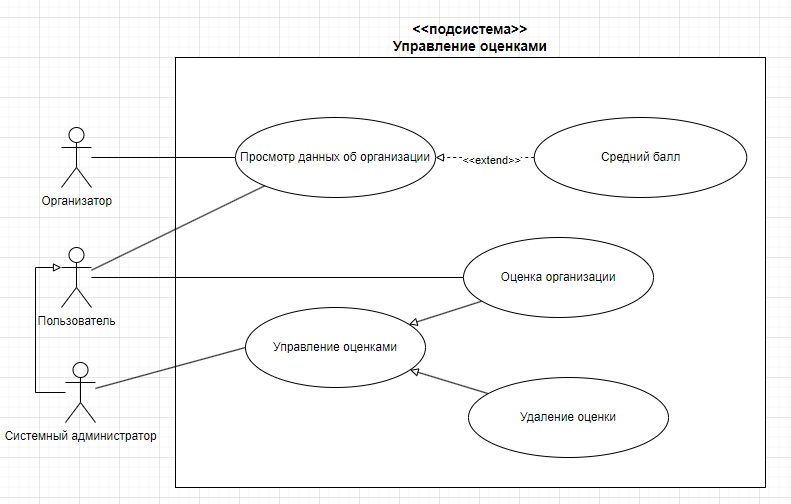












# Нефункциональные требования

|  |  |
| --- | --- |
| **NFR-01** | Система реализуется в виде веб-сайта |
| **NFR-02** | Система должна предоставлять API для использования функционала системы сторонними приложениями |
| **NFR-03** | Система организует сбор и хранение персональных данных в соответствии с требования 152 ФЗ |
| **NFR-04** | Система организует защиту персональных данных в соответствии с требования 21 приказа ФСТЭК |
| **NFR-05** | Язык системы русский |
| **NFR-06** | Время безотказной работы системы 99.5% в год |
| **NFR-07** | Время генерации страницы сайта системы не более 2 секунд |
| **NFR-08** | Время отклика API системы не более 1.5 секунд |
| **NFR-09** | Система должна выполнять требования NFR-05, NFR-06,NFR-07, NFR-08 при нагрузке до 1000 запросов в 1 секунду включительно. |

# Ролевая модель

|  |  |
| --- | --- |
| **Роль** | **Разрешения** |
| Гость | регистрация в системе  просмотр ленты событий  просмотр подробной информации о событии (без списка участников с отзывами и средней оценкой) |
| Пользователь (авторизованный) | **права гостя +**  изменение данных своей учетной записи  просмотр подробной информации о событии (со списком друзей участвующих в событии)  просмотр ленты событий друзей  регистрация/отмена регистрации в событии  свои отзывы на организатора (CRUD)  выставление оценки организатору  подача/отмена запроса дружбы  подтверждение/отказ запроса дружбы от другого пользователя  регистрация организации |
| Организатор (пользователь владелец организации) | **права пользователя (а) +**  создание организации (премодерируемое событие)  изменение данных об организации (премодерируемое событие)  создание события  изменение события  отмена события  просмотр заявок на участие в своем событии  подтверждение/отклонение заявок на участие в событии  просмотр оценок своей организации |
| Администратор системы | **права организатора+**  пользователи CRUD  организации CRUD  отзывыCRUD  оценкиCRUD  события CRUD |

# Сценарии использования

## UC-01 Создание заявки на участие в событии

|  |  |
| --- | --- |
| **Участники** | Пользователь (авторизованный), Организатор, Система |
| **Предусловия** | Событие создано организатором |
| **Результат** | Пользователь зарегистрирован в событии |
| **Связанные ФТ** | FR-04-02-01, FR-04-02-02, FR-04-02-04 |
| **Пользовательская история** | Я, как пользователь хочу зарегистрироваться на участие в определенном событии, чтобы принять в нем участие |
| **Триггер** | Пользователь нажимает кнопку «принять участие» в карточке события |

### Основной сценарий

|  |  |  |
| --- | --- | --- |
| **№ шага** | **Описание шага** | **Переходы** |
| B1 | Пользователь переходит в карточку события | => Е1 |
| B2 | Пользователь нажимает кнопку «принять участие» |  |
| B3 | Система регистрирует заявку пользователя на участие в событии со статусом «WAITING» |  |
| B4 | Система отправляет запрос организатору события на подтверждение участия пользователя | =>Е5  => E6 |
| B5 | Система меняет статус заявки пользователя на «APPROVED» |  |
| B6 | Система, посредством электронной почты, отправляет уведомление пользователю об успешной регистрации в событии |  |
| B7 | Сценарий завершается успешно |  |

### Сценарии исключения

#### Е1. Отсутствие свободных мест

|  |  |  |
| --- | --- | --- |
| **№ шага** | **Описание шага** | **Переходы** |
| Е1.1 | Система проверяет наличие свободных мест в событии и в случае отсутствия свободных мест, система отображает кнопку «принять участие» неактивной |  |
| Е1.2 | Сценарий завершается исключением |  |

#### Е5. Отказ организатора

|  |  |  |
| --- | --- | --- |
| **№ шага** | **Описание шага** | **Переходы** |
| А5.1 | Организатор отклоняет заявку пользователя |  |
| А5.2 | Система переводит статус заявки в состояние «REJECTED» |  |
| А5.3 | Система, посредством электронной почты, отправляет уведомление пользователю о том, что его запрос на участие в событии отклонен | => R5 |
| A5.4 | Сценарий завершается исключением |  |

#### E6. Отсутствие свободных мест

|  |  |  |
| --- | --- | --- |
| **№ шага** | **Описание шага** | **Переходы** |
| E6.1 | Система выполняет проверку на наличие свободных |  |
| E6.2 | Система выводит сообщение об ошибке, в случае отсутствия свободных мест | => R6 |
| E6.3 | Сценарий завершается с исключением |  |

### Сценарии восстановления

#### R5.Восстановление из сценариев исключения E5

|  |  |  |
| --- | --- | --- |
| **№ шага** | **Описание шага** | **Переходы** |
| R5.3.2 | Система предлагает пользователю выбрать событие из списка доступных к регистрации событий с аналогичной датой старта (+/- 1 день) |  |
| R5.3.3 | Процесс продолжается по базовому сценарию | =>B1 |

#### R6.Восстановление из сценариев исключения Е6

|  |  |  |
| --- | --- | --- |
| **№ шага** | **Описание шага** | **Переходы** |
| R6.2.1 | Система предлагает пользователю выбрать событие из списка доступных к регистрации событий с аналогичной датой старта (+/- 1 день) |  |
| R6.2.2 | Процесс продолжается по базовому сценарию | =>B1 |

## UC-02Создание события

|  |  |
| --- | --- |
| **Участники** | Организатор, Система |
| **Предусловия** | Статус организатора «APPROVED» |
| **Результат** | Событие создано |
| **Связанные ФТ** | FR-03-01-01 |
| **Пользовательская история** | Я, как организатор хочу создать новое событие, чтобы другие пользователи могли в нем поучаствовать |
| **Триггер** | Организатор нажимает кнопку «создать событие» в личном кабинете организатора |

### Основной сценарий

|  |  |  |
| --- | --- | --- |
| **№ шага** | **Описание шага** | **Переходы** |
| B1 | Организатор переходит в личный кабинет организатора | =>E1 |
| B2 | Организатор нажимает кнопку «создать событие» |  |
| B3 | Система отображает форму создания события |  |
| B4 | Организатор заполняет данные о событии |  |
| B5 | Организатор нажимает кнопку «сохранить событие» |  |
| B6 | Система проверяет, что все поля карточки события заполнены | =>E6 |
| B6 | Системасохраняет новое событие |  |
| B7 | Сценарий завершается успешно |  |

### Сценарии исключения

#### E1. Отсутствие свободных мест

|  |  |  |
| --- | --- | --- |
| **№ шага** | **Описание шага** | **Переходы** |
| E1.1 | Если статус организатора отличен от «APPROVED», система блокирует весь функционал кабинета, кроме редактирования персональных данных. |  |
| E1.2 | Сценарий завершается исключением |  |

#### E6. Некорректное заполнение данных события

|  |  |  |
| --- | --- | --- |
| **№ шага** | **Описание шага** | **Переходы** |
| E6.1 | Система выводит сообщение об ошибке, если одно из полей формы создания события не заполнено | => R6 |
| E6.2 | Сценарий завершается с исключением |  |

### Сценарии восстановления

#### R5.Восстановление из сценария исключения E5

|  |  |  |
| --- | --- | --- |
| **№ шага** | **Описание шага** | **Переходы** |
| R6.1.1 | Система возвращает пользователя на страницу заполнения формы создания события |  |
| R6.1.2 | Процесс продолжается по базовому сценарию | =>B4 |

## UC-03Создание отзыва

|  |  |
| --- | --- |
| **Участники** | Пользователь, Система |
| **Предусловия** | Пользователь авторизован |
| **Результат** | Отзыв на организатора создан |
| **Связанные ФТ** | FR-06-01-01 |
| **Пользовательская история** | Я, как пользователь, хочу иметь возможность добавить отзыв на организатора, чтобы помочь другим пользователям сделать выбор участвовать в события данного организатора или нет. |
| **Триггер** | Пользователь нажимает кнопку «добавить отзыв» в карточке организатора |

### Основной сценарий

|  |  |  |
| --- | --- | --- |
| **№ шага** | **Описание шага** | **Переходы** |
| B1 | Пользователь переходит в карточку организатора |  |
| B2 | Пользователь нажимает кнопку «добавить отзыв» |  |
| В3 | Система проверяет, участвовал ли пользователь хотя бы в одном событии организатора | =>Е3 |
| B4 | Система отображает форму создания отзыва |  |
| B5 | Пользователь заполняет форму отзыва |  |
| B6 | Пользователь нажимает кнопку «сохранить отзыв» |  |
| B7 | Система проверяет, что пользователь выбрал в каком событии он участвовал у данного организатора | => Е7 |
| В8 | Система проверяет, что пользователь поставил оценку организатору | => Е8 |
| B9 | Система сохраняет новое событие |  |
| B10 | Сценарий завершается успешно |  |

### Сценарии исключения

#### E3.Некорректное заполнение данных события

|  |  |  |
| --- | --- | --- |
| **№ шага** | **Описание шага** | **Переходы** |
| E3.1 | Система выводит сообщение об ошибке, если пользователь не участвовал в событиях организатора |  |
| E3.2 | Сценарий завершается ошибкой |  |

#### E7. Не выбрано событие, в котором участвовал пользователь

|  |  |  |
| --- | --- | --- |
| **№ шага** | **Описание шага** | **Переходы** |
| E7.1 | Система выводит сообщение об ошибке, если пользователь не выбрал в каком событии участвовал у данного организатора | =>R7 |
| E7.3 | Сценарий завершается ошибкой |  |

#### E8. Отсутствие оценки организатору

|  |  |  |
| --- | --- | --- |
| **№ шага** | **Описание шага** | **Переходы** |
| E8.1 | Система выводит сообщение об ошибке, если пользователь не поставил оценку организатору | =>R5 |
| E8.2 | Сценарий завершается ошибкой |  |

### Сценарии восстановления

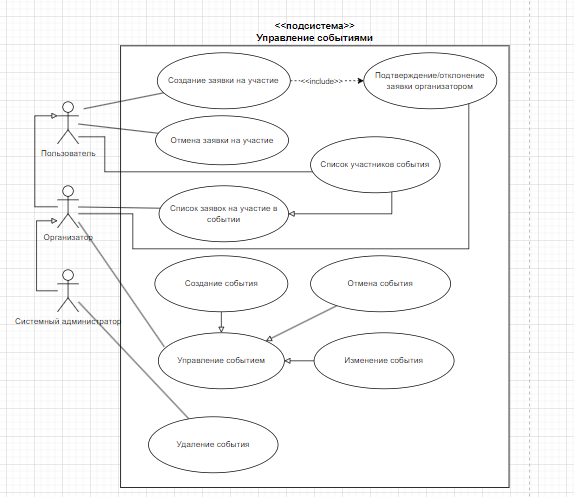
#### R7.Восстановление из сценария исключения E7

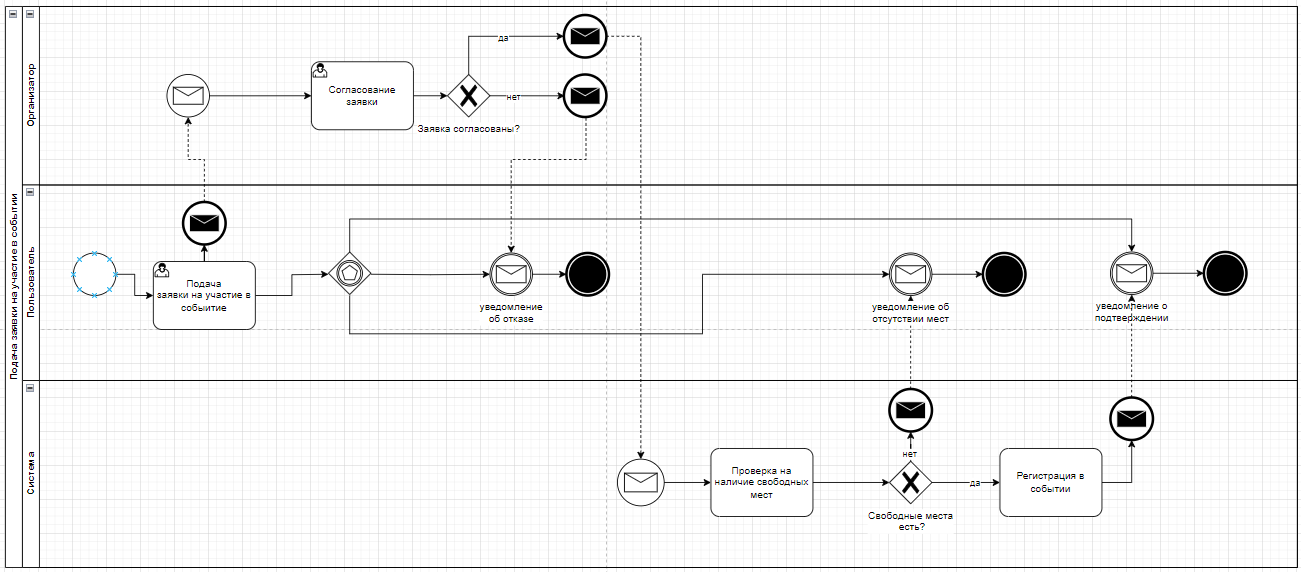
|  |  |  |
| --- | --- | --- |
| **№ шага** | **Описание шага** | **Переходы** |
| R7.1 | Система возвращает пользователя на страницу заполнения формы отзыва |  |
| R7.2 | Процесс продолжается по базовому сценарию | =>B4 |

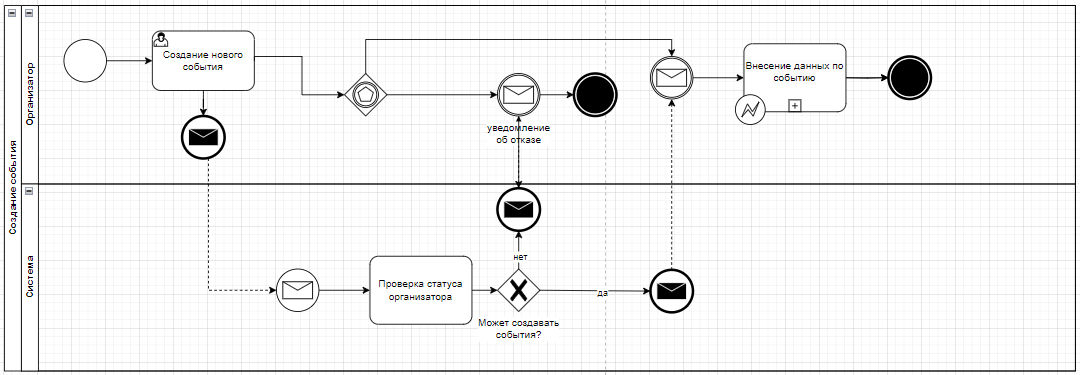
#### R8. Восстановление из сценария исключения E8

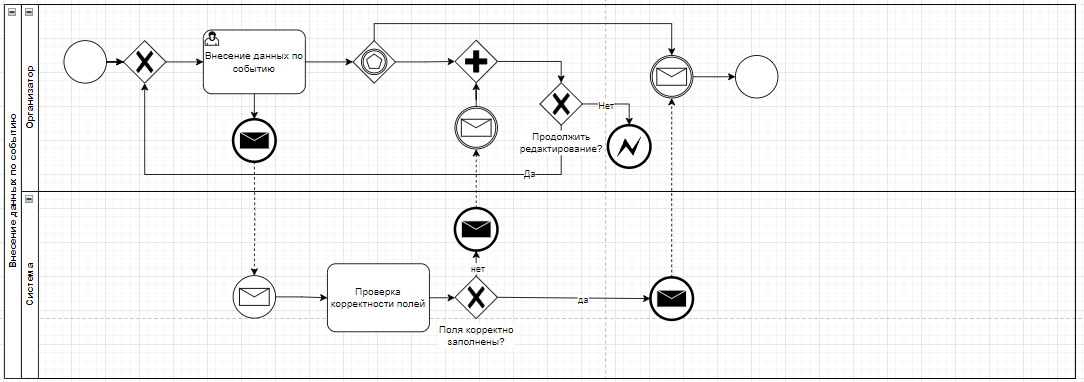
|  |  |  |
| --- | --- | --- |
| **№ шага** | **Описание шага** | **Переходы** |
| R8.1 | Система возвращает пользователя на страницу заполнения формы отзыва |  |
| R8.2 | Процесс продолжается по базовому сценарию | =>B4 |

# Схема процессов (BPMN)



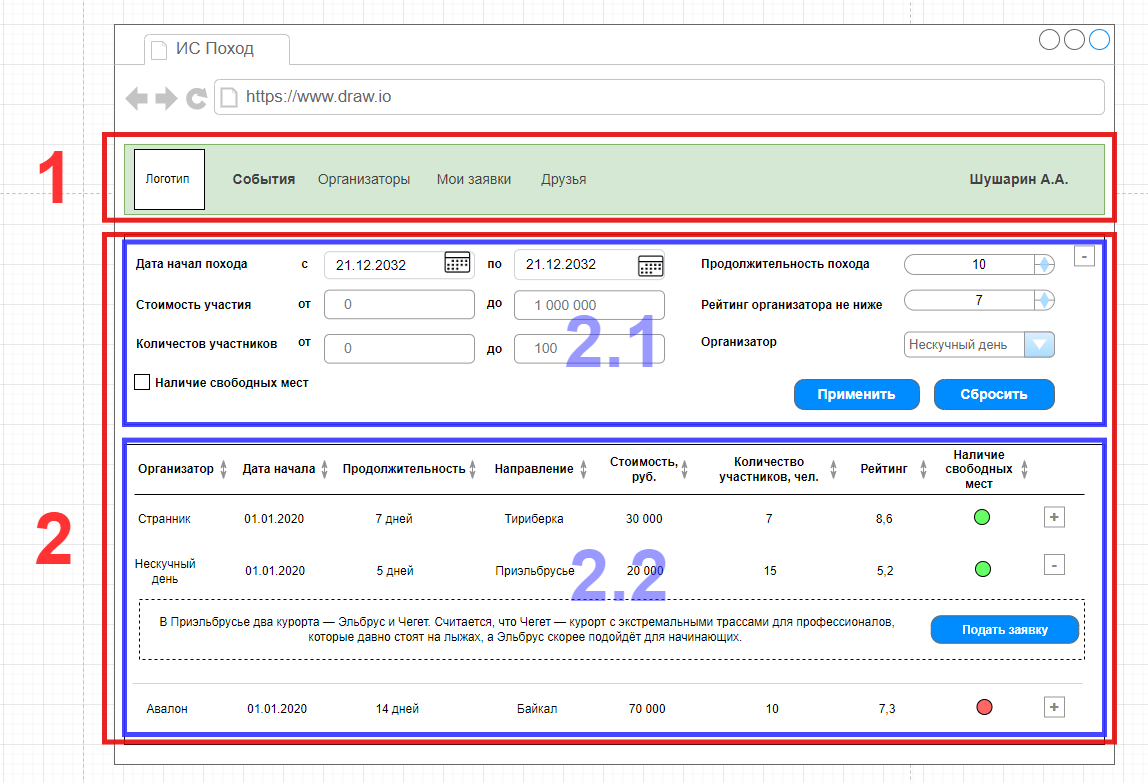






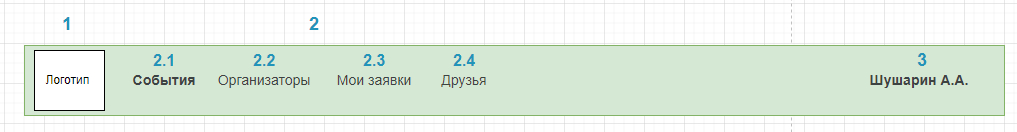
# Пользовательский интерфейс (UI)

## ЭФ-01 Основной экран



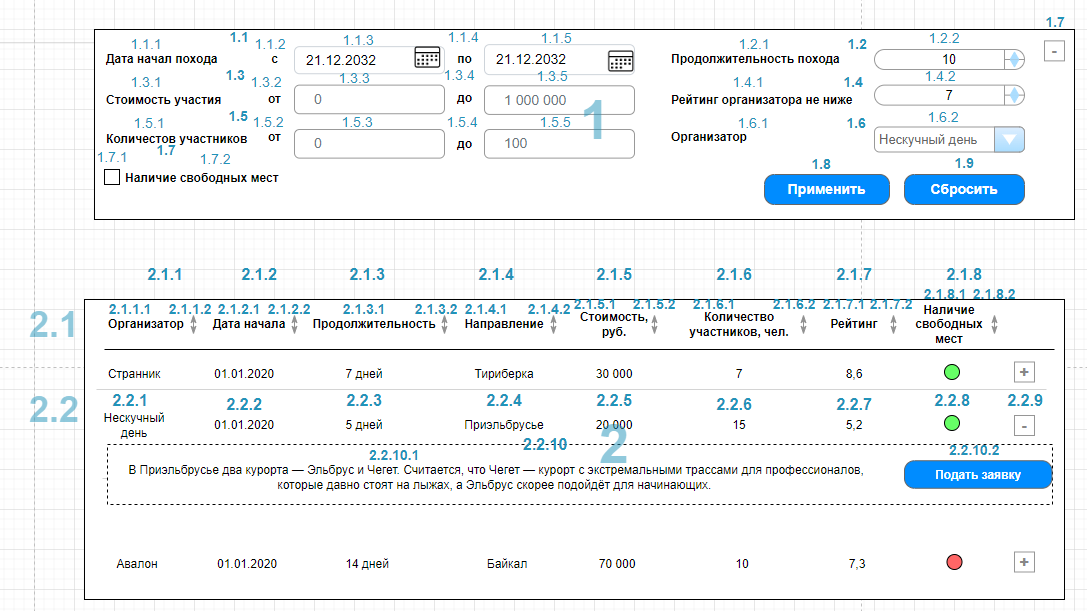
|  |  |  |  |
| --- | --- | --- | --- |
| **№ на**  **схеме** | **Тип** | **Назначение** | **Примечания** |
| 1 | Контейнер | Отображение строки меню (ЭФ-02) |  |
| 2 | Контейнер | Отображение ЭФ, реализующих основные функции приложения |  |
| 2.1 | Контейнер | Отображение ЭФ настроек фильтрации событий |  |
| 2.2 | Контейнер | Отображение ЭФ списка событий |  |

## ЭФ-02 Строка меню



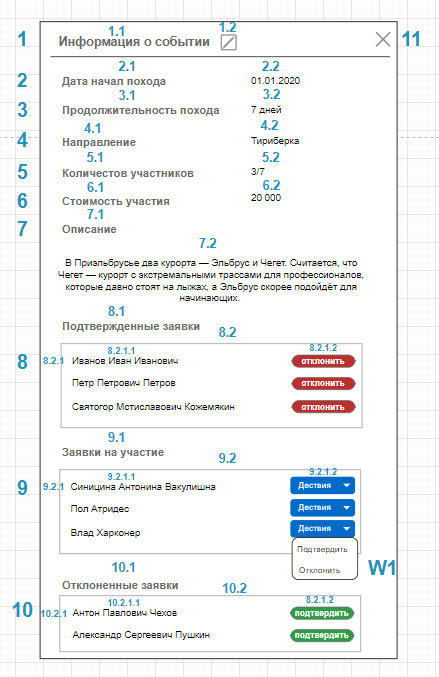
| **№ насхеме** | **Тип** | **Назначение** | **ФЛК** | | **Связьсданными** | **Обработкасобытий** | **Примечания** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Обяз.** | **Формат** |
| **1** | **Изображение/Гиперссылка** | **Логотип приложения переход к ЭФ-11** |  |  |  | **Щелчок:**  ***Отображение ЭФ-11 в области 2 ЭФ-01*** |  |
| **2** | **Группа элементов** |  |  |  |  |  |  |
| 2.1 | Гиперссылка | Открытие ЭФ-11 «События» |  |  |  | Щелчок:  Отображение ЭФ-11 в области 2 ЭФ-01 |  |
| 2.2 | Гиперссылка | Открытие ЭФ-12 «Организаторы» |  |  |  | Щелчок:  Отображение ЭФ-12 в области 2 ЭФ-01 |  |
| 2.3 | Гиперссылка | Открытие ЭФ-13 «Мои заявки» |  |  |  | Щелчок:  Отображение ЭФ-13 в области 2 ЭФ-01 |  |
| 2.4 | Гиперссылка | Открытие ЭФ-14 «Друзья» |  |  |  | Щелчок:  Отображение ЭФ-14 в области 2 ЭФ-01 |  |
| **3** | **Гиперссылка** | **Отображение имени пользования**  **переход к ЭФ-21** |  |  |  | **Щелчок:**  **Отображение ЭФ-21 (карточка пользователя) в области 2 ЭФ-01** |  |

## ЭФ-02 Список событий



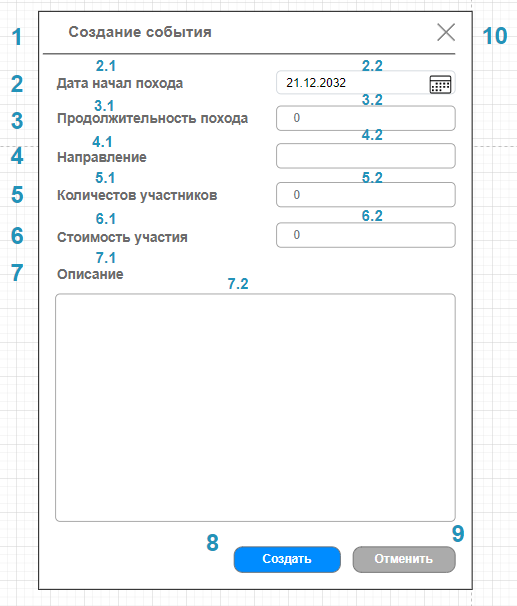
| **№ насхеме** | **Тип** | **Назначение** | **ФЛК** | | **Связьсданными** | **Обработкасобытий** | **Примечания** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Обяз.** | **Формат** |
| **1** | **Группаэлементов** | **Область настройки фильтрации событий** |  |  |  |  |  |
| **1.1** | **Группа элементов** |  |  |  |  |  |  |
| 1.1.1 | Надпись | Подпись к ЭУ 1.1.3 и 1.1.5 |  |  |  |  |  |
| 1.1.2 | Надпись | Подпись к ЭУ 1.1.3 |  |  |  |  |  |
| 1.1.3 | Поле ввода даты | Ввод значения нижней границы интервала дат для фильтрации по дате начала события |  | ДД.ММ.ГГГГ |  |  |  |
| 1.1.4 | Надпись | Подпись к ЭУ 1.1.5 |  |  |  |  |  |
| 1.1.5 | Поле ввода даты | Ввод значения верхней границы интервала дат для фильтрации по дате начала события |  | ДД.ММ.ГГГГ |  |  |  |
| **1.2** | **Группаэлементов** |  |  |  |  |  |  |
| 1.2.1 | Надпись | Подпись к ЭУ 1.2.2 |  |  |  |  |  |
| 1.2.2 | Spinner | Ввод количества дней продолжительности похода |  | 1<=N<=90 |  |  |  |
| **1.3** | **Группа элементов** |  |  |  |  |  |  |
| 1.3.1 | Надпись | Подпись к ЭУ 1.3.3 и 1.3.5 |  |  |  |  |  |
| 1.3.2 | Надпись | Подпись к ЭУ 1.3.3 |  |  |  |  |  |
| 1.3.3 | Поле ввода | Ввод значения нижней границы интервала дат для фильтрации по стоимости участия |  | 1<=N  <= 1 000 000 |  |  |  |
| 1.3.4 | Надпись | Подпись к ЭУ 1.3.5 |  |  |  |  |  |
| 1.3.5 | Поле ввода | Ввод значения верхней границы интервала дат для фильтрации по стоимости участия |  | 1<=N  <= 1 000 000 |  |  |  |
| **1.4** | **Группаэлементов** |  |  |  |  |  |  |
| 1.4.1 | Надпись | Подпись к ЭУ 1.4.2 |  |  |  |  |  |
| 1.4.2 | Spinner | Ввод значения минимального рейтинга организатора |  | 0<=R<= 10 |  |  | Инкремент кнопками производится с шагом 0,1 |
| **1.5** | **Группа элементов** |  |  |  |  |  |  |
| 1.5.1 | Надпись | Подпись к ЭУ 1.5.3 и 1.5.5 |  |  |  |  |  |
| 1.5.2 | Надпись | Подпись к ЭУ 1.5.3 |  |  |  |  |  |
| 1.5.3 | Поле ввода | Ввод значения нижней границы интервала для фильтрации помаксимальномуколичеству участников |  | 0 < N < 50 |  |  |  |
| 1.5.4 | Надпись | Подпись к ЭУ 1.5.5 |  |  |  |  |  |
| 1.5.5 | Поле ввода | Ввод значения верхней границы интервала для фильтрации по максимальномуколичеству участников |  | 1< N< 50 |  |  |  |
| **1.6** | **Группаэлементов** |  |  |  |  |  |  |
| 1.6.1 | Надпись | Подпись к ЭУ 1.6.2 |  |  |  |  |  |
| 1.6.2 | Выпадающий список | Выбор организатора |  |  |  | Щелчок:  открывается список организаторов |  |
| ***1.7*** | **Группаэлементов** |  |  |  |  |  |  |
| 1.7.1 | Чекбокс | При нажатом чекбоксе выбираются только те события где есть свободные места |  |  |  |  |  |
| 1.7.2 | Надпись | Подпись к ЭУ 1.7.1 |  |  |  |  |  |
| **1.8** | **Кнопка** | **Применить параметры фильтрации** |  |  |  | Щелчок:  Отображение в ЭФ 2 событий отвечающих критериям фильтров ЭУ 1.1—1.7 |  |
| **1.9** | **Кнопка** | **Сброс фильтров** |  |  |  | Щелчок:  Сброс значений в ЭУ 1.1-1.7 Отображение в ЭФ 2 всех событий |  |
| **1.10** | **Кнопка** | **Скрыть настройки фильтрации** |  |  |  | Щелчок:  Скрывает группу элементов ЭУ 1 |  |
| **2** | **Группаэлементов** | **Область отображения событий** |  |  |  |  |  |
| **2.1** | **Группа элементов** | **Шапка таблицы событий** |  |  |  |  |  |
| **2.1.1** | **Группаэлементов** |  |  |  |  |  |  |
| 2.1.1.1 | Надпись | Подпись к ЭУ 2.2.1 |  |  |  |  |  |
| 2.1.1.2 | Гиперссылка | Сортировка событий в ЭФ 2.2. |  |  |  | Щелчок:  Сортировка событий по полю «организатор» (по алфавитупо возрастанию) | Порядок сортировки при повторном щелчке меняется на противоположный |
| **2.1.2** | **Группаэлементов** |  |  |  |  |  |  |
| 2.1.1.1 | Надпись | Подпись к ЭУ 2.2.2 |  |  |  |  |  |
| 2.1.1.2 | Гиперссылка | Сортировка событий в ЭФ 2.2. |  |  |  | Щелчок:  Сортировка событий по полю «дата начала» от более ранних к более поздним | Порядок сортировки при повторном щелчке меняется на противоположный |
| **2.1.3** | **Группаэлементов** |  |  |  |  |  |  |
| 2.1.3.1 | Надпись | Подпись к ЭУ 2.2.3 |  |  |  |  |  |
| 2.1.3.2 | Гиперссылка | Сортировка событий в ЭФ 2.2. |  |  |  | Щелчок:  Сортировка событий по полю «продолжительность» по убыванию | Порядок сортировки при повторном щелчке меняется на противоположный |
| **2.1.4** | **Группаэлементов** |  |  |  |  |  |  |
| 2.1.4.1 | Надпись | Подпись к ЭУ 2.2.2 |  |  |  |  |  |
| 2.1.4.2 | Гиперссылка | Сортировка событий в ЭУ 2.2. |  |  |  | Щелчок:  Сортировка событий по полю «направление» по алфавитупо возрастанию | Порядок сортировки при повторном щелчке меняется на противоположный |
| **2.1.5** | **Группаэлементов** |  |  |  |  |  |  |
| 2.1.5.1 | Надпись | Подпись к ЭУ 2.2.5 |  |  |  |  |  |
| 2.1.5.2 | Гиперссылка | Сортировка событий в ЭУ 2.2. |  |  |  | Щелчок:  Сортировка событий по полю «стоимость» по убыванию | Порядок сортировки при повторном щелчке меняется на противоположный |
| **2.1.6** | **Группаэлементов** |  |  |  |  |  |  |
| 2.1.6.1 | Надпись | Подпись к ЭУ 2.2.6 |  |  |  |  |  |
| 2.1.6.2 | Гиперссылка | Сортировка событий в ЭУ 2.2. |  |  |  | Щелчок:  Сортировка событий по полю количество участников по убыванию | Порядок сортировки при повторном щелчке меняется на противоположный |
| **2.1.7** | **Группаэлементов** |  |  |  |  |  |  |
| 2.1.7.1 | Надпись | Подпись к ЭУ 2.2.7 |  |  |  |  |  |
| 2.1.7.2 | Гиперссылка | Сортировка событий в ЭУ 2.2. |  |  |  | Щелчок:  Сортировка событий по рейтингу организатора события по убыванию | Порядок сортировки при повторном щелчке меняется на противоположный |
| **2.1.8** | **Группаэлементов** |  |  |  |  |  |  |
| 2.1.8.1 | Надпись | Подпись к ЭУ 2.2.8 |  |  |  |  |  |
| 2.1.8.2 | Гиперссылка | Сортировка событий в ЭУ 2.2. |  |  |  | Щелчок:  Сортировка событий по наличию свободных мест (в начале события с имеющие в наличии свободные места) | Порядок сортировки при повторном щелчке меняется на противоположный |
| **2.2** | **Группа элементов** | **Область показа событий** |  |  |  |  |  |
| 2.2.1 | Надпись | Организатор события |  |  |  |  |  |
| 2.2.2 | Надпись | Дата начала события |  |  |  |  |  |
| 2.2.3 | Надпись | Продолжительность события |  |  |  |  |  |
| 2.2.4 | Надпись | Направление события |  |  |  |  |  |
| 2.2.5 | Надпись | Стоимость события |  |  |  |  |  |
| 2.2.6 | Надпись | Количество участников события |  |  |  |  |  |
| 2.2.7 | Надпись | Рейтинг организатора события |  |  |  |  |  |
| 2.2.8 | Рисунок | Отображает наличие свободных мест в событии |  |  |  |  | Зеленый цвет места есть, красный цвет мест нет |
| 2.2.9 | Кнопка | Открывает подробную информацию о событии |  |  |  | Щелчок:  Отображение ЭУ 2.2.10 | При повторном нажатии скрывает подробную информацию |
| **2.2.10** | **Группаэлементов** |  |  |  |  |  |  |
| *2.2.10.1* | *Надпись* | *Подробное описание событие* |  |  |  |  |  |
| *2.2.10.2* | *Кнопка* | *Регистрация пользователя в событии* |  |  |  | Щелчок:  Регистрация пользователя в событии | Элемент активен, только при наличии свободных мест и отсутствии регистрации и пользователя на данное событие  После регистрации кнопка становится неактивной |

## ЭФ-20 Карточка события (Режим просмотра, Роль – Организатор)



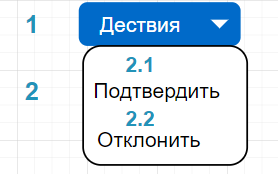
| **№ насхеме** | **Тип** | **Назначение** | **ФЛК** | | **Связьсданными** | **Обработкасобытий** | **Примечания** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Обяз.** | **Формат** |
| **1** | **Группаэлементов** |  |  |  |  |  |  |
| 1.1 | Надпись | Заголовок ЭФ |  |  |  |  |  |
| 1.2 | Гиперссылка | Переход в режим редактирования заявки, отображение ЭФ-22 |  |  |  | Щелчок:  Закрытие текущей ЭФ, открытие ЭФ-22 |  |
| **2** | **Группаэлементов** |  |  |  |  |  |  |
| 2.1 | Надпись | Подпись к ЭУ 2.1 |  |  |  |  |  |
| 2.2 | Надпись | Отображение даты начала похода |  |  |  |  |  |
| **3** | **Группаэлементов** |  |  |  |  |  |  |
| 3.1 | Надпись | Подпись к ЭУ 3.1 |  |  |  |  |  |
| 3.2 | Надпись | Отображение продолжительности похода |  |  |  |  |  |
| **4** | **Группаэлементов** |  |  |  |  |  |  |
| 4.1 | Надпись | Подпись к ЭУ 4.1 |  |  |  |  |  |
| 4.2 | Надпись | Отображение направления похода |  |  |  |  |  |
| **5** | **Группаэлементов** |  |  |  |  |  |  |
| 5.1 | Надпись | Подпись к ЭУ 5.1 |  |  |  |  |  |
| 5.2 | Надпись | Отображение количества подтвержденных участников / максимального количества участников |  |  |  |  |  |
| **6** | **Группаэлементов** |  |  |  |  |  |  |
| 6.1 | Надпись | Подпись к ЭУ 6.1 |  |  |  |  |  |
| 6.2 | Надпись | Отображение стоимости участия |  |  |  |  |  |
| **7** | **Группаэлементов** |  |  |  |  |  |  |
| 7.1 | Надпись | Подпись к ЭУ 7.1 |  |  |  |  |  |
| 7.2 | Надпись | Отображение подробного описания события |  |  |  |  |  |
| **8** | **Группаэлементов** |  |  |  |  |  |  |
| 8.1 | Надпись | Подпись к ЭУ 8.2 |  |  |  |  |  |
| 8.2 | Группа элементов | Областьпоказаподтвержденныхзаявок |  |  |  |  |  |
| 8.2.1 | Группа элементов | Область показа завки |  |  |  |  |  |
| 8.2.1.1 | Надпись | ФИО пользователя |  |  |  |  |  |
| 8.2.1.2 | Кнопка | Отмена подтвержденной заявки |  |  |  | Щелчок:  Отмена подтверждения заявки пользователя  Обновлений значений в ЭУ 5.1  Обновление данных ЭУ 8  Обновление данных ЭУ 9 |  |
| **9** | **Группаэлементов** |  |  |  |  |  |  |
| 9.1 | Надпись | Подпись к ЭУ 9.2 |  |  |  |  |  |
| 9.2 | Группа элементов | Область показа заявок на рассмотрении |  |  |  |  |  |
| 9.2.1 | Группа элементов | Область показа заявки |  |  |  |  |  |
| 9.2.1.1 | Надпись | ФИО пользователя |  |  |  |  |  |
| 9.2.1.2 | Кнопка | Обработка заявки (отказ или подтверждение) |  |  |  | См. спецификацию виджета W1  Обновлений значений в ЭУ 5.1  Обновление данных ЭУ 8  Обновление данных ЭУ 9  Обновление данных ЭУ 10 |  |
| **10** | **Группаэлементов** |  |  |  |  |  |  |
| 10.1 | Надпись | Подпись к ЭУ 10.2 |  |  |  |  |  |
| 10.2 | Группа элементов | Областьпоказа отклоненных заявок |  |  |  |  |  |
| 9.2.1 | Группа элементов | Область показа заявки |  |  |  |  |  |
| 10.2.1.1 | Надпись | ФИО пользователя |  |  |  |  |  |
| 10.2.1.2 | Кнопка | Подтверждение отклоненной заявки |  |  |  | Щелчок:  Отмена подтверждения заявки пользователя  Обновлений значений в ЭУ 5.1  Обновление данных ЭУ 8  Обновление данных ЭУ 9 |  |
| **11** | **Кнопка** | Закрытие карточки события |  |  |  | Щелчок:  Закрытие текущей ЭФ |  |

## ЭФ-21 Карточка события (Режим создания, Роль - Организатор)



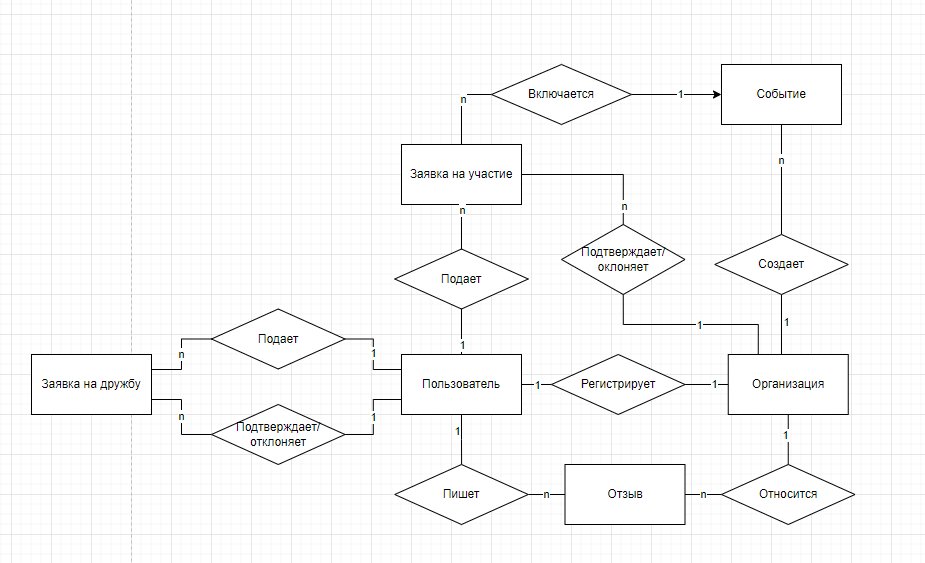
| **№ насхеме** | **Тип** | **Назначение** | **ФЛК** | | **Связьсданными** | **Обработкасобытий** | **Примечания** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Обяз.** | **Формат** |
| 1 | Надпись | Заголовок ЭФ |  |  |  |  |  |
| **2** | **Группаэлементов** |  |  |  |  |  |  |
| 2.1 | Надпись | Подпись к ЭУ 2.1 |  |  |  |  |  |
| 2.2 | Поле ввода даты | Ввод даты начала события | Да | ДД.ММ.ГГГ |  |  |  |
| **3** | **Группаэлементов** |  |  |  |  |  |  |
| 3.1 | Надпись | Подпись к ЭУ 3.1 |  |  |  |  |  |
| 3.2 | Поле ввода | Ввод продолжительности похода | Да | 1 <N <90 |  |  |  |
| **4** | **Группаэлементов** |  |  |  |  |  |  |
| 4.1 | Надпись | Подпись к ЭУ 4.1 |  |  |  |  |  |
| 4.2 | Поле ввода | Ввод направления похода | Да | Текст, неболее64символов |  |  |  |
| **5** | **Группаэлементов** |  |  |  |  |  |  |
| 5.1 | Надпись | Подпись к ЭУ 5.1 |  |  |  |  |  |
| 5.2 | Поле ввода | Ввод максимального количества участников | Да | 1 <N <50 |  |  |  |
| **6** | **Группаэлементов** |  |  |  |  |  |  |
| 6.1 | Надпись | Подпись к ЭУ 6.1 |  |  |  |  |  |
| 6.2 | Поле ввода | Ввод стоимости участия в событии | Да | 0 <= N  < 1 000 000 |  |  |  |
| **7** | **Группаэлементов** |  |  |  |  |  |  |
| 7.1 | Надпись | Подпись к ЭУ 7.1 |  |  |  |  |  |
| 7.2 | Многострочноеполе ввода | Ввод подробного описания события | Да | Не более 2048 символов |  |  |  |
| **8** | **Кнопка** | Сохранение созданной заявки |  |  |  | Щелчок:  Создание заявки  Закрытие текущей ЭФ  Открытие ЭФ-20 |  |
| **9** | **Кнопка** | Отмена создания заявки |  |  |  | Щелчок:  Закрытие текущей ЭФ |  |
| **10** | **Кнопка** | Закрытие карточки события |  |  |  | Щелчок:  Закрытие текущей ЭФ |  |

## W1 Виджет действия с заявкой

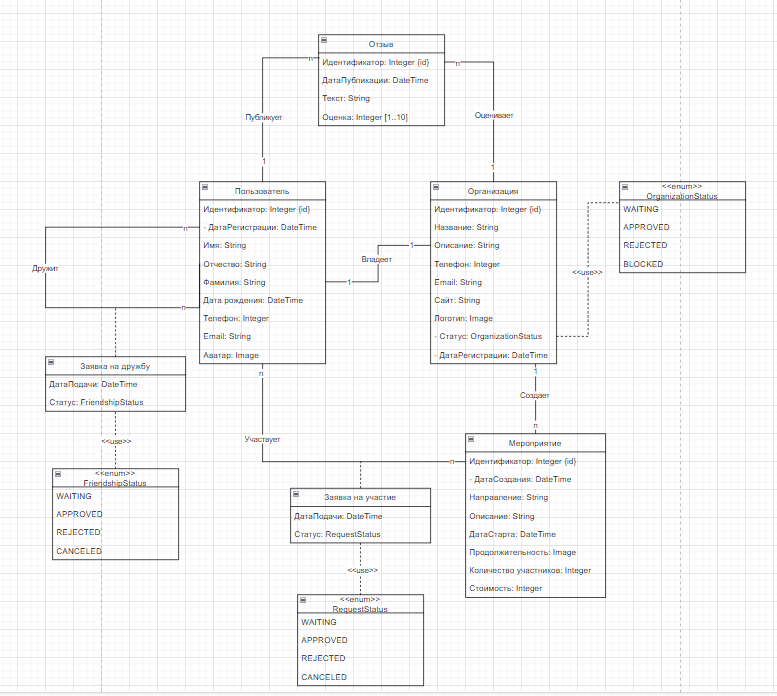


|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№ насхеме** | **Тип** | **Назначение** | **ФЛК** | | **Связьсданными** | **Обработкасобытий** | **Примечания** |
| **Обяз.** | **Формат** |
| 1 | Кнопка | Вывод выпадающего меню |  |  |  | Щелчок:  Отображение ЭУ 2 |  |
| 2 | Выпадающее меню | Изменение статуса заявки |  |  |  | Щелчок по элементу меню:  Установка соответствующего статуса заявки |  |

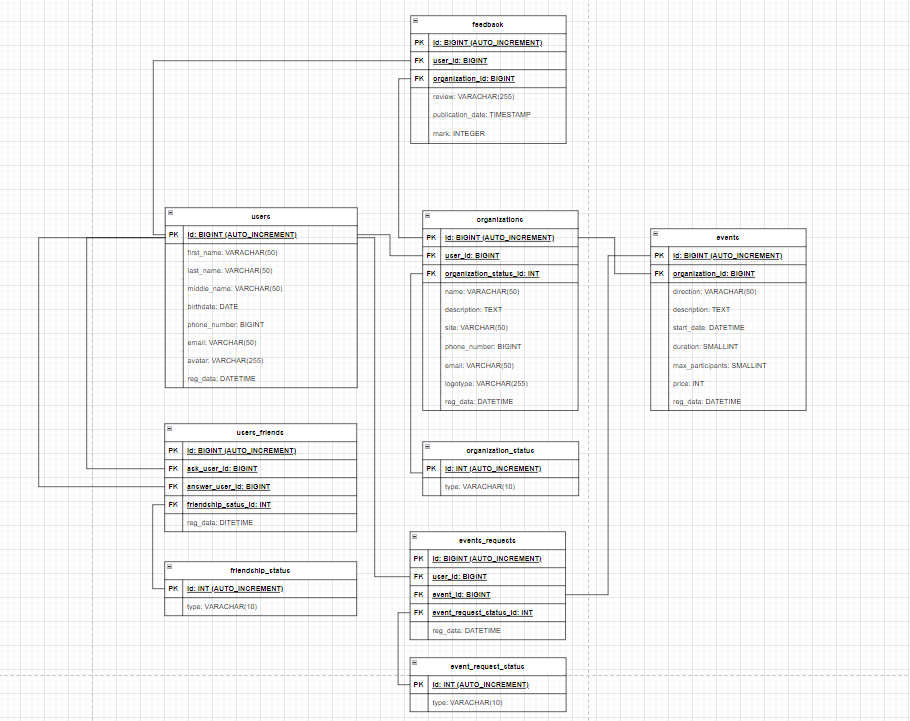
# Концептуальная модель



# Диаграмма классов



# Схема базы данных



## **users**

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Тип** | **Ограничения** | **Назначение** |
| id | BIGINT | PRIMARY KEY  AUTO\_INCREMENT | Идентификатор пользователя |
| first\_name | VARCHAR(50) | NOT NULL | Имя |
| last\_name | VARCHAR(50) | NOT NULL | Фамилия |
| middle\_name | VARCHAR(50) | NOT NULL | Отчество |
| birthdate | DATE | NOT NULL | Дата рождения |
| phone\_number | BIGINT | UNIQUE  NOT NULL | Телефон с кодом страны |
| email | VARCHAR(50) | NOT NULL | Электронная почта |
| avatar | VARCHAR(255) | - | Изображение (в базе хранится ссылка на местоположение файла на сервере) |
| reg\_data | DATETIME | NOT NULL | Дата регистрации (заполняется автоматически) |

## **organizations**

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Тип** | **Ограничения** | **Назначение** |
| id | BIGINT | PRIMARY KEY  AUTO\_INCREMENT | Идентификатор заявки |
| user\_id | BIGINT | FOREIGN\_KEY REFERENCES users(id)  NOT NULL | Владелец организации (ссылка на таблицу users) |
| organization\_status\_id | INT | FOREIGN\_KEY REFERENCES organization\_status(id)  NOT NULL | Статус организации (устанавливается администратором) |
| name | VARCHAR(50) | NOT NULL | Название |
| description | TEXT | NULLABLE | Описание |
| site | VARCHAR(50) | NULLABLE | Ссылка на сайт организации |
| phone\_number | BIGINT | NOT NULL | Телефон с кодом страны |
| email | VARCHAR(50) | NOT NULL | Электронная почта |
| logotype | VARCHAR(255) | NULLABLE | Изображение (в базе хранится ссылка на местоположение файла на сервере) |
| reg\_data | DATETIME | NOT NULL | Дата регистрации (заполняется автоматически) |

## **events**

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Тип** | **Ограничения** | **Назначение** |
| id | BIGINT | PRIMARY KEY  AUTO\_INCREMENT | Идентификатор заявки |
| organization\_id | BIGINT | FOREIGN\_KEY REFERENCES organizations(id)  NOT NULL | Организатор события (ссылка на таблицу organizations) |
| direction | VARCHAR(50) | NOT NULL | Направление похода |
| description | TEXT | NOT NULL | Подробное описание |
| start\_date | DATETIME | NOT NULL | Дата и время начала похода (события) |
| duration | SMALLINT | NOT NULL | Длительность в часах (до 90 дней) |
| max\_participants | SMALLINT | NOT NULL | Максимальное количество участников (до 100 чел) |
| price | INT | NOT NULL | Цена в рублях (точность до 100 руб.) |
| reg\_data | DATETIME | NOT NULL | Дата регистрации (заполняется автоматически) |

**[ЭФ-20]**

**--Наполнениеэкраннойформы**

SELECT e.direction, e.description, e.start\_date, e.duration,

CONCAT(COUNT(er.id), '/', e.max\_participants) AS participants\_count, e.price

FROM events as e

LEFT JOIN events\_requestsAS er ON e.id = er.event\_id

    AND er.event\_request\_status\_id =

      (SELECT ers.id FROM event\_request\_status AS ers WHERE ers.type = 'WAITING')

WHERE e.id = {{id события}}

GROUP BY e.id

**[ЭФ-20: ЭУ-8.2.1.2, ЭУ-9.2.1.2,ЭУ-10.2.1.2]**

**--Изменениестатусазаявкиксобытию**

UPDATE events\_requests SET event\_request\_status\_id =

(SELECT id FROM event\_request\_status WHERE type = {{Запрашиваемыйстатус}})

WHERE id = {{id текущего события}}

**[ЭФ-21]**

**--Добавлениесобытия**

INSERT INTO events (organization\_id, direction, description, start\_date, duration,  max\_participants, price, reg\_data)

    VALUES(

{{idорганизации}},

{{текст из ЭУ-4.2 ЭФ-21}},

{{текст из ЭУ-7.2 ЭФ-21}},

{{текст из ЭУ-2.2 ЭФ-21}},

{{текст из ЭУ-3.2 ЭФ-21}},

{{текст из ЭУ-5.2 ЭФ-21}},

{{текст из ЭУ-6.2 ЭФ-21}},

NOW(),

)

**[ЭФ-02, ЭУ-2.2]**

**--Наполнениеэкраннойформы**

SELECT o.name, e.start\_date, e.duration, e.direction, e.price, e.max\_participants,e.description,

(SELECT SUM(fb.mark)/COUNT(fb.id) FROM feedback as fb

    WHERE fb.organization\_id = e.organization\_id) AS rating,

(COUNT(er.id) <e.max\_participants) AS free\_slots

FROM events as e

JOIN organizations AS o ON e.organization\_id = o.id

LEFT JOIN events\_requestsAS er ON e.id = er.event\_id

    AND er.event\_request\_status\_id =

        (SELECT ers.id FROM event\_request\_statusASersWHEREers.type = 'APPROVED')

*--параметры фильтрации добавляются опциональна при заполнении полей ЭУ-1*

WHERE

  e.organization\_id = {{idорганизации из ЭУ-1.6.2 ЭФ-02}}

  ANDe.start\_date>= {{1.1.3}}ANDe.start\_date<= {{текстиз 1.1.5 ЭФ-02}}

    ANDe.duration = {{текстизЭУ-1.2.2 ЭФ-02}}

    ANDe.price>= {{текстизЭУ-1.3.3 ЭФ-02}}ANDe.price<= {{текстизЭУ-1.3.5 ЭФ-02}}

    ANDe.max\_participants>= {{текстизЭУ-1.5.3 ЭФ-02}}AND e.max\_participants <= {{текстизЭУ-1.5.5 ЭФ-02}}

    AND rating >{{текстизЭУ-1.4.2 ЭФ-02}}

  ANDfree\_slots = {{булево значение изЭУ-1.7.1 ЭФ-02}}

GROUPBYe.id

*--параметры сортировки добавляются опциональна по клику на ЭУ-2.1.1.2 … ЭУ-2.1.8.2, по повторному клику переключается (ASC/DESC)*

ORDERBY

    o.name,

    e.start\_date,

    e.duration,

    e.direction,

    e.price,

    e.max\_participants,

    rating,

    free\_slots

**[ЭФ-02, ЭУ-2.2.10.1]**

**--Описаниесобытия**

SELECT e.description

FROM events as e

WHERE

  e.id = 1

**[ЭФ-02, ЭУ-2.2.10.2]**

**--Подача заявки на участие**

INSERT INTO events\_requests (user\_id, event\_id, event\_request\_status\_id, reg\_data)

    VALUES(

{{idорганизации}},

{{idпросматриваемого события}},

SELECT ers.id FROMevent\_request\_statusASersWHEREers.type = 'WAITING',

CURTIME()

)

## **events\_requests**

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Тип** | **Ограничения** | **Назначение** |
| id | BIGINT | PRIMARY KEY  AUTO\_INCREMENT | Идентификатор заявки |
| user\_id | BIGINT | FOREIGN\_KEY REFERENCES users(id)  NOT NULL | Автор заявки на участие в событии (ссылка на таблицу users) |
| event\_id | BIGINT | FOREIGN\_KEY REFERENCES events(id)  NOT NULL | Событие на участие в котором подана заявка (ссылка на таблицу events) |
| event\_request\_status\_id | INT | FOREIGN\_KEY REFERENCES event\_request\_status(id)  NOT NULL | Статус заявки |
| reg\_data | DATETIME | NOT NULL | Дата подачи заявки(заполняется автоматически) |

**[ЭФ-20: ЭУ-8.2, ЭУ-9.2, ЭУ-10.2]**

**--Вывод пользователей подавших заявку на участие в событии**

SELECT u.id, CONCAT(u.first\_name, ' ' ,u.last\_name) AS participant

FROM users AS u

JOIN events\_requestsAS er ONer.user\_id = u.id

JOIN event\_request\_statusASersON ers.id = er.event\_request\_status\_id

WHERE er.event\_id = {{id события}}ANDers.type = {{статусзаявки}}

## **feedback**

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Тип** | **Ограничения** | **Назначение** |
| id | BIGINT | PRIMARY KEY  AUTO\_INCREMENT | Идентификатор заявки |
| user\_id | BIGINT | FOREIGN\_KEY REFERENCES users(id)  NOT NULL | Автор отзыва на участие в событии (ссылка на таблицу users) |
| organization\_id | BIGINT | FOREIGN\_KEY REFERENCES organization (id)  NOT NULL | Организатор на которого публикуется отзыв (ссылка на таблицу organizations) |
| review | TEXT | - | Отзыв |
| publication\_date | DATETIME | NOT NULL | Дата публикации (заполняется автоматически) |
| mark | INT | NOT NULL | Оценка |

## **users\_friends**

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Тип** | **Ограничения** | **Назначение** |
| id | BIGINT | PRIMARY KEY  AUTO\_INCREMENT | Идентификатор связи |
| ask\_user\_id | BIGINT | FOREIGN\_KEY REFERENCES users(id)  NOT NULL | Автор запроса на дружбу (ссылка на таблицу users) |
| answer\_user\_id | BIGINT | FOREIGN\_KEY REFERENCES users(id)  NOT NULL | Получатель запроса на дружбу (ссылка на таблицу users) |
| friendship\_status\_id | INT | FOREIGN\_KEY REFERENCES friendship\_status(id)  NOT NULL | Статус дружбы |
| reg\_data | DATETIME | NOT NULL | Дата подачи заявки(заполняется автоматически) |

## **friendship\_status**

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Тип** | **Ограничения** | **Назначение** |
| id | INT | PRIMARY KEY  AUTO\_INCREMENT | Идентификатор связи |
| type | VARCHAR(10) | UNIQUE NOT NULL | Справочниксостоянийдружбы  {WAITING, APPROVED,REJECTED, CANCELED} |

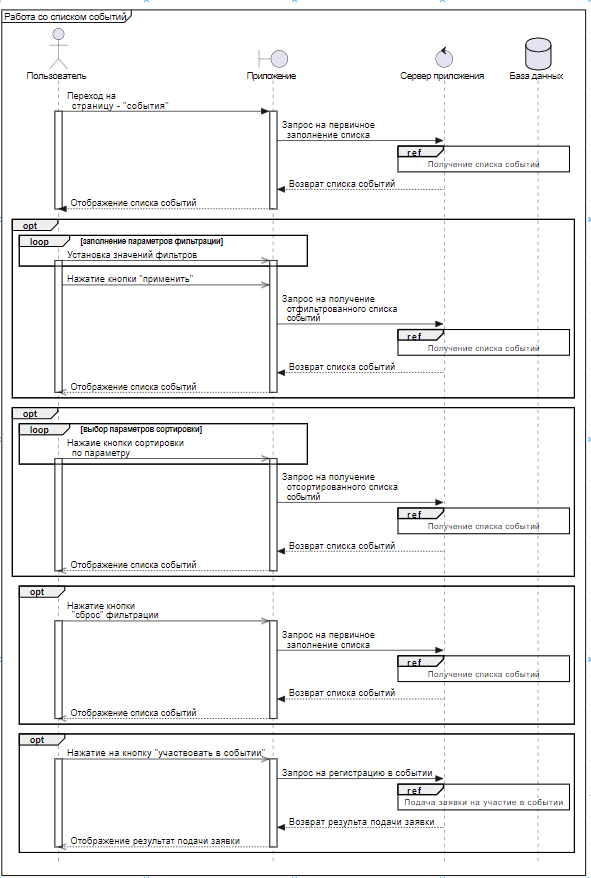
## **event\_request\_status**

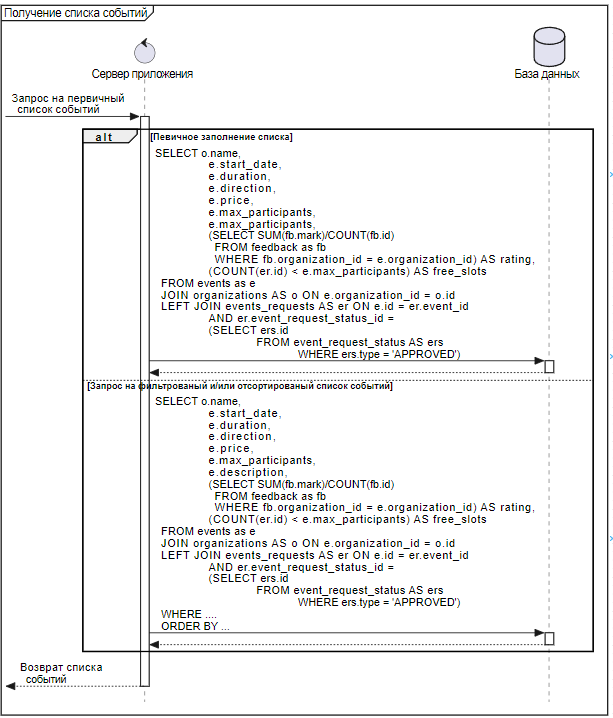
|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Тип** | **Ограничения** | **Назначение** |
| id | INT | PRIMARY KEY  AUTO\_INCREMENT | Идентификатор связи |
| type | VARCHAR(10) | UNIQUE NOT NULL | Справочник состояний заявки на участие {WAITING, APPROVED, REJECTED, CANCELED} |

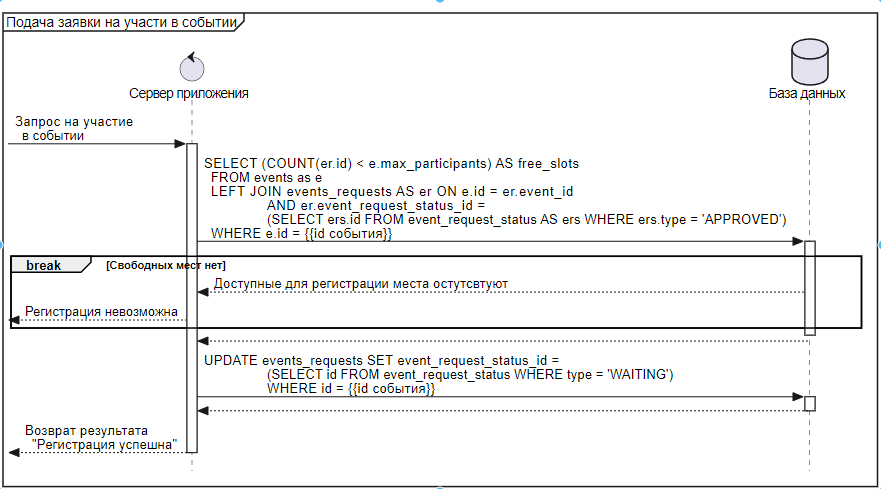
## **organization\_status**

|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Тип** | **Ограничения** | **Назначение** |
| id | INT | PRIMARY KEY  AUTO\_INCREMENT | Идентификатор связи |
| type | VARCHAR(10) | UNIQUE NOT NULL | Справочник состояний организаций  {WAITING, APPROVED,REJECTED, BLOCKED} |

Диаграмма использования







# JSONschemas

## EventListDto.schema.json *Схема JSON для списка событий*

{

  "$schema":"https://json-schema.org/draft/2020-12/schema",

  "$id": "http://mydomain.com/schemas/EventList.schema.json",

  "title": "Списоксобытий",

  "type": "array",

  "items":

  {

    "type": "object",

    "properties": {

      "id": {"type": "integer", "minimum": 1},

      "organization": {

        "type": "object",

        "properties": {

          "id": {"type": "integer", "minimum": 1},

          "name": {"type": "string", "minLength": 1}

        },

        "required": ["id", "name"]

      },

      "start\_date": {"type": "string", "format": "date"},

      "duration": {"type": "integer", "minimum": 1},

      "direction": {"type": "string", "minLength": 1},

      "price": { "type": "integer", "minimum": 0},

      "max\_participants": { "type": "integer", "minimum": 1},

      "rating": {"type": ["number", "null"], "minimum": 1, "maximum": 10},

      "free\_slots": {"type": "boolean"},

      "description": {"type": "string", "minLength": 1}

    },

    "required": [

      "id",

      "organization",

      "start\_date",

      "duration",

      "direction",

      "price",

      "max\_participants",

      "rating",

      "free\_slots",

      "description"

    ]

  }

}

## PostEventDto.schema.json *Схема JSON для добавления события*

{

  "$schema":"https://json-schema.org/draft/2020-12/schema",

  "$id": "http://mydomain.com/schemas/PostEvent.schema.json",

  "title": "Списоксобытий",

   "type": "object",

    "properties": {

      "organization\_id": {"type": "integer", "minimum": 1},

      "start\_date": {"type": "string", "format": "date"},

      "duration": {"type": "integer", "minimum": 1},

      "direction": {"type": "string", "minLength": 1},

      "price": {"type": "integer", "minimum": 0},

      "max\_participants": {"type": "integer", "minimum": 1},

      "description": {"type": "string", "minLength": 1}

    },

    "required": [

      "organization\_id",

      "start\_date",

      "duration",

      "direction",

      "price",

      "max\_participants",

      "description"

    ]

}

## PutDto.schema.json *Схема JSON для изменения события*

{

 "$schema":"https://json-schema.org/draft/2020-12/schema",

 "$id": "http://mydomain.com/schemas/PostEvent.schema.json",

  "title": "Списоксобытий",

   "type": "object",

    "properties": {

      "id": {"type": "integer", "minimum": 1},

"organization\_id": {"type": "integer", "minimum": 1},

      "start\_date": {"type": "string", "format": "date"},

      "duration": {"type": "integer", "minimum": 1},

      "direction": {"type": "string", "minLength": 1},

      "price": {"type": "integer", "minimum": 0},

      "max\_participants": {"type": "integer", "minimum": 1},

      "description": {"type": "string", "minLength": 1}

    },

    "required": [

"id",

"organization\_id",

      "start\_date",

      "duration",

      "direction",

      "price",

      "max\_participants",

      "description"

    ]

}

# API endpoint (/events)

## GET /events *Получение массива событий, удовлетворяющих фильтрам*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| organization-id | Параметр запроса | Нет | Integer | Идентификаторорганизатора |
| start-date-after | Параметр запроса | Нет | Дата в формате ГГГГ-ММ-ДД | Нижняя граница интервала начала события |
| start-date-before | Параметр запроса | Нет | Дата в формате ГГГГ-ММ-ДД | Верхняя граница интервала начала события |
| duration | Параметр запроса | Нет | Integer | Продолжительность события в днях |
| price-min | Параметр запроса | Нет | Integer | Минимальная цена события |
| price-max | Параметр запроса | Нет | Integer | Максимальная цена события |
| min-participants-amount | Параметр запроса | Нет | Integer | Минимальное количество участников |
| max-participants-amount | Параметр запроса | Нет | Integer | Максимальное количество участников |
| min-rating | Параметр запроса | Нет | Integer | Минимальный рейтинг |
| free-slots | Параметр запроса | Нет | Boolean | Наличие свободных мест |
| organization-name | Параметр запроса | Нет | ENUM (“asc”, “desc”) | Сортировка событий по названию организатора |
| order-start-date | Параметр запроса | Нет | ENUM (“asc”, “desc”) | Сортировка событий по дате события |
| order-duration | Параметр запроса | Нет | ENUM (“asc”, “desc”) | Сортировка событий по продолжительности |
| order-direction | Параметр запроса | Нет | ENUM (“asc”, “desc”) | Сортировка событий по направлению |
| order-price | Параметр запроса | Нет | ENUM (“asc”, “desc”) | Сортировка событий по цене |
| order-max-participants | Параметр запроса | Нет | ENUM (“asc”, “desc”) | Сортировка событий по названию организатора |
| order-rating | Параметр запроса | Нет | ENUM (“asc”, “desc”) | Сортировка событий по рейтингу |
| order-free-slots | Параметр запроса | Нет | ENUM (“asc”, “desc”) | Сортировка событий по наличию свободных мест |

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| GET events/?price\_min=10000&order\_price=asc | HTTP 200 + JSON |

## GET /events/{event-id} *Получение данных конкретного события*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| event-id | Параметр пути | Да | Integer | Идентификатор запрашиваемогособытия. |

### Примеры запроса

|  |  |
| --- | --- |
| **Пример** | **Ответ** |
| GET events/1 | HTTP 200 + JSON: |

## GET /events/{event-id}/requests (role: owner) *Получение данных конкретного события для «владельца события» c информацией о заявках*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| event-id | Параметр пути | Да | Integer | Идентификатор запрашиваемого события. |

### Примеры запроса

|  |  |
| --- | --- |
| **Пример** | **Ответ** |
| GET events/1/requests | HTTP 200 + JSON: |

## POST /events *Создание нового события*

### Параметры запроса

Отсутствуют

### Тело запроса

|  |  |
| --- | --- |
| **Наличие/обязательность** | Обязательно |
| **Описание** | JSONcданными создаваемой заявки |
| **Схема объекта данных** | см. PostEventDto.schema.json |

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| POST /events + телозапроса: | HTTP 201 + JSON: |

## DELETE /events/{event-id} *Удаление события*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| event-id | Параметр пути | Да | Integer | Идентификатор запрашиваемого события. |

### Тело запроса

Отсутствует

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| DELETE /events/1 | HTTP 200 |

## PUT /events/{event-id} *Изменение события*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| event-id | Параметр пути | Да | Integer | Идентификатор запрашиваемого события. |

### Тело запроса

|  |  |
| --- | --- |
| **Наличие/обязательность** | Обязательно |
| **Описание** | JSONcданными создаваемой заявки |
| **Схема объекта данных** | см. PutEventDto.schema.json |

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| PUT /events/1 + телозапроса: | HTTP 200 |

# API endpoint (/requests)

## GET /requests *Получение массива заявок, удовлетворяющих фильтрам*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| event-id | Параметр запроса | Нет | Integer | Идентификаторсобытия на участие в котором подана заявка |
| user-id | Параметр запроса | Нет | Integer | Идентификаторпользователя подавшего заявку |
| status | Параметр запроса | Нет | ENUM({WAITING, APPROVED, REJECTED, CANCELED}) | Статус заявки |

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| GET /request/?event-id =1 | HTTP 200 + JSON |

## GET /requests/{request-id} *Получение данных конкретной заявки на участие в событии*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| request-id | Параметр пути | Да | Integer | Идентификатор запрашиваемой заявки |

### Примеры запроса

|  |  |
| --- | --- |
| **Пример** | **Ответ** |
| GET /requests/1 | HTTP 200 + JSON: |

## POST /requests *Создание новой заявки*

### Параметры запроса

Отсутствуют

### Тело запроса

|  |  |
| --- | --- |
| **Наличие/обязательность** | Обязательно |
| **Описание** | JSONcданными создаваемоuj заявки |
| **Схема объекта данных** | см. PostRequestDto.schema.json |

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| POST /requests + телозапроса (JSON): | HTTP 201 + JSON: |

## DELETE /requests/{request-id} *Удаление заявки*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| request-id | Параметр пути | Да | Integer | Идентификатор запрашиваемой заявки |

### Тело запроса

Отсутствует

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| DELETE /requests/1 | HTTP 200 |

## PUT /requests/{request-id} *Изменение заявки*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| request-id | Параметр пути | Да | Integer | Идентификатор запрашиваемой заявки |

### Тело запроса

|  |  |
| --- | --- |
| **Наличие/обязательность** | Обязательно |
| **Описание** | JSON c данными изменяемой заявки |
| **Схема объекта данных** | см. PutRequestDto.schema.json |

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| PUT /requests/1 + телозапроса (JSON): | HTTP 200 |

# API endpoint (/organization)

## GET /organizations *Получение массива организаций удовлетворяющей фильтрам*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| event-id | Параметр запроса | Нет | Integer | Идентификатор события владельцем которого является организатор |
| user-id | Параметр запроса | Нет | Integer | Идентификатор пользователя который является владельцем организации |

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| GET organization/?event-id=1 | HTTP 200 + JSON |

## GET /organizations/{organization-id} *Получение данных конкретной организации*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| organization-id | Параметр пути | Да | Integer | Идентификатор запрашиваемой организации |

### Примеры запроса

|  |  |
| --- | --- |
| **Пример** | **Ответ** |
| GET organizations/1 | HTTP 200 + JSON: |

## POST /organizations *Создание новой организации*

### Параметры запроса

Отсутствуют

### Тело запроса

|  |  |
| --- | --- |
| **Наличие/обязательность** | Обязательно |
| **Описание** | JSONcданными создаваемойорганизации |
| **Схема объекта данных** | см. PostOrganizationtDto.schema.json |

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| POST /organizations + телозапроса (JSON): | HTTP 201 + JSON: |

## DELETE /organizations/{organization-id} *Удаление организации*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| organization-id | Параметр пути | Да | Integer | Идентификатор удаляемой организации |

### Тело запроса

Отсутствует

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| DELETE /organizations/1 | HTTP 200 |

## PUT /organizations/{organization-id} *Изменение организации*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| organization-id | Параметр пути | Да | Integer | Идентификатор запрашиваемой организации |

### Тело запроса

|  |  |
| --- | --- |
| **Наличие/обязательность** | Обязательно |
| **Описание** | JSON c данными создаваемой организации |
| **Схема объекта данных** | см. PutOrganizationDto.schema.json |

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| PUT /organizations/1 + телозапроса (JSON): | HTTP 200 |

# API endpoint (/users)

## GET /users *Получение массива пользователей удовлетворяющих фильтрам*

### Параметры

Отсутствуют

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| GET users/ | HTTP 200 + JSON |

## GET /users/{user-id} *Получение данных конкретного пользователя*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| user-id | Параметр пути | Да | Integer | Идентификатор запрашиваемого пользователя |

### Примеры запроса

|  |  |
| --- | --- |
| **Пример** | **Ответ** |
| GET users/1 | HTTP 200 + JSON: |

## POST /users *Создание нового пользователя*

### Параметры запроса

Отсутствуют

### Тело запроса

|  |  |
| --- | --- |
| **Наличие/обязательность** | Обязательно |
| **Описание** | JSONcданными создаваемого пользователя |
| **Схема объекта данных** | см. PostUserDto.schema.json |

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| POST /users + тело запроса (JSON): | HTTP 201 + JSON: |

## DELETE /users/{user-id} *Удаление пользователя*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| user-id | Параметр пути | Да | Integer | Идентификатор удаляемогопользователя |

### Тело запроса

Отсутствует

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| DELETE /user/1 | HTTP 200 |

## PUT /users/{user-id} *Изменение пользователя*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| user-id | Параметр пути | Да | Integer | Идентификатор запрашиваемого пользователя |

### Тело запроса

|  |  |
| --- | --- |
| **Наличие/обязательность** | Обязательно |
| **Описание** | JSON c данными изменяемого пользователя |
| **Схема объекта данных** | см. PutUserDto.schema.json |

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| PUT/users/1 + тело запроса (JSON): | HTTP 200 |

# API endpoint (/friendships)

## GET /friendships *Получение массива заявок на дружбу удовлетворяющую фильтрам*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| user-id | Параметр запроса | Нет | Integer | Идентификаторпользователя, для которого ищутся друзья |
| friend-id | Параметр запроса | Нет | Integer | Идентификаторпользователя-друга |
| status | Параметр запроса | Нет | ENUM({WAITING, APPROVED, REJECTED}) | Статус заявки |

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| GET /friendships | HTTP 200 + JSON |

## GET /friendships/{friendship-id} *Получение данных конкретного запроса на дружбу*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| friendship-id | Параметр пути | Да | Integer | Идентификатор запрашиваемой заявки на дружбу |

### Примеры запроса

|  |  |
| --- | --- |
| **Пример** | **Ответ** |
| GET friendships/1 | HTTP 200 + JSON: |

## POST /friendships *Создание заявки на дружбу*

### Параметры запроса

Отсутствуют

### Тело запроса

|  |  |
| --- | --- |
| **Наличие/обязательность** | Обязательно |
| **Описание** | JSONcданными создаваемой заявки |
| **Схема объекта данных** | см. PostFriendshipDto.schema.json |

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| POST /friendships + телозапроса (JSON): | HTTP 201 + JSON: |

## DELETE /friendships/{friendship-id} *Удаление заявки на дружбу*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| friendship-id | Параметр пути | Да | Integer | Идентификатор удаляемого пользователя |

### Тело запроса

Отсутствует

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| DELETE /friendships/1 | HTTP 200 |

## PUT /friendships/{friendship-id} *Изменение статуса заявки на дружбу*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| friendship -id | Параметр пути | Да | Integer | Идентификатор заявки на дружбу |

### Тело запроса

|  |  |
| --- | --- |
| **Наличие/обязательность** | Обязательно |
| **Описание** | JSON c данными изменяемой заявки на дружбу |
| **Схема объекта данных** | см. PutFriendshipDto.schema.json |

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| PUT/friendships/1 + телозапроса (JSON): | HTTP 200 + JSON |

# APIendpoint (/feedback)

## GET /feedbacks *Получение массива отзывов удовлетворяющей фильтрам*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| user-id | Параметр запроса | Нет | Integer | Идентификаторпользователя, который писал отзывы |
| organization-id | Параметр запроса | Нет | Integer | Идентификатор организации, для которой составлен отзыв |
| min-mark | Параметр запроса | Нет | Integer | Минимальная оценка |
| max-mark | Параметр запроса | Нет | Integer | Максимальная оценка |

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| GET /feedbacks | HTTP 200 + JSON |

## GET /feedbacks/{feedback-id} *Получение данных конкретного отзыва*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| feedback-id | Параметр пути | Да | Integer | Идентификатор запрашиваемого пользователя |

### Примеры запроса

|  |  |
| --- | --- |
| **Пример** | **Ответ** |
| GET /feedbacks/1 | HTTP 200 + JSON: |

## POST /feedbacks *Создание нового отзыва*

### Параметры запроса

Отсутствуют

### Тело запроса

|  |  |
| --- | --- |
| **Наличие/обязательность** | Обязательно |
| **Описание** | JSONcданными создаваемого пользователя |
| **Схема объекта данных** | см. PostFeedbackDto.schema.json |

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| POST /feedbacks + телозапроса (JSON): | HTTP 201 + JSON: |

## DELETE /feedbacks/{feedback-id} *Удаление отзыва*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| feedback-id | Параметр пути | Да | Integer | Идентификатор удаляемого пользователя |

### Тело запроса

Отсутствует

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| DELETE /feedbacks/1 | HTTP 200 |

## PUT /feedbacks/{feedback-id} *Изменение отзыва*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| feedback-id | Параметр пути | Да | Integer | Идентификатор запрашиваемого отзыва |

### Тело запроса

|  |  |
| --- | --- |
| **Наличие/обязательность** | Обязательно |
| **Описание** | JSON изменяемого отзыва |
| **Схема объекта данных** | см. PutFeedbackDto.schema.json |

### Примеры запроса

|  |  |
| --- | --- |
| **Запрос** | **Ответ** |
| PUT/feedbacks/1 + телозапроса (JSON): | HTTP 200 + JSON |

# Развернутая спецификация API

## GET /events/{event-id} *Получение данных конкретного события*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| event-id | Параметр пути | Да | Integer | Идентификатор запрашиваемого события. |

#### Тело запроса

|  |  |
| --- | --- |
| **Наличие/обязательность** | отсутствует |
| **Описание** | ‑ |

### Заголовки запроса

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| Authorization | Да | Bearer + JWT в Base64 | Токен аутентификации |

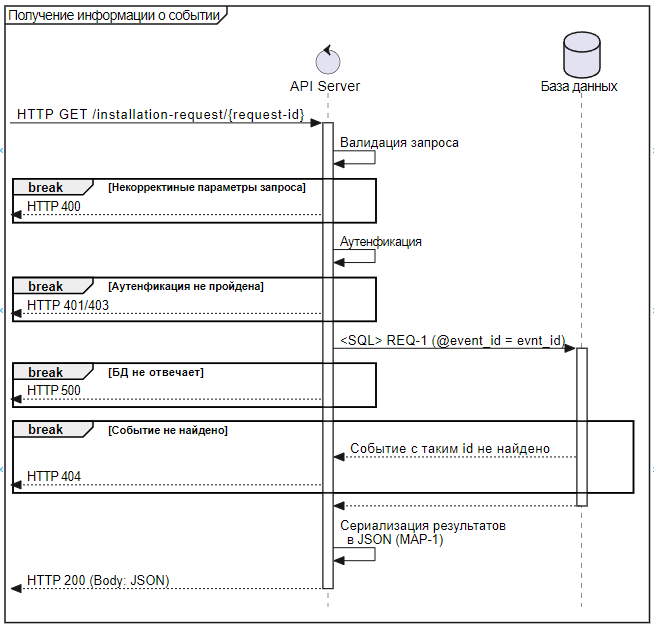
### Примеры запроса

|  |  |
| --- | --- |
| **Пример** | **Ответ** |
| GET events/1 | HTTP 200 + JSON: |

#### Варианты ответа

|  |  |  |
| --- | --- | --- |
| **Код состояния** | **Описание** | |
| 200 | **Комментарий** | Событие с запрошенным ID найдено |
| **Тело ответа** | JSON-представление найденного события |
| 404 | **Комментарий** | Событие с запрошенным ID не найдена |
| **Тело ответа** | JSON с детальной информацией об ошибке |
| 400 | **Комментарий** | Запрос содержит ошибку |
| **Тело ответа** | JSON с детальной информацией об ошибке |
| 401 | **Комментарий** | Отсутствует токен в заголовке Authorization |
| **Тело ответа** | JSON с детальной информацией об ошибке |
| 403 | **Комментарий** | Ошибка аутентификации или недостаточно прав |
| **Тело ответа** | JSON с детальной информацией об ошибке |
| 500 | **Комментарий** | Ошибка на стороне сервера |
| **Тело ответа** | JSON с детальной информацией об ошибке |

### Сценарий информационного взаимодействия



1. Клиент отправляет запрос GET /events/{event-id};
2. Сервер осуществляет проверку корректности запроса (наличие обязательных полей запроса, корректность значений полей запроса)
   1. В случае, если запрос некорректен, возвращается ошибка 400;
3. Сервер осуществляет проверку наличия токена аутентификации в заголовке «Authorization»;
   1. При отсутствии в запросе заголовка Authrozationтокена аутентификации или наличия в нем содержимого, отличного от «Bearer + JWT в Base64», возвращается ошибка 401;
4. Сервер осуществляет верификацию токена и проверку выполнения хотя бы одного из следующих наличия у пользователя права на получения данных заявок;
   1. Если токен не верифицирован (срок действия истек либо подпись не подтверждена), либо у пользователя недостаточно прав, возвращается ошибка 403;
5. Сервер осуществляет подключение к серверу СУБД:
   1. При ошибке или невозможности подключения к базе данных возвращается код состояния 500 (InternalServerError)
6. Сервер осуществляет запрос (REQ-1) к базе данных:

SELECT

e.id,

o.id

o.name,

e.start\_date,

e.duration,

e.direction,

e.price,

e.max\_participants,

e.description,

(SELECT SUM(fb.mark)/COUNT(fb.id) FROM feedback as fb

WHERE fb.organization\_id = e.organization\_id) AS rating,

(COUNT(er.id) < e.max\_participants) AS free\_slots

FROM events as e

JOIN organizations AS o ON e.organization\_id = o.id

LEFT JOIN events\_requests AS er ON e.id = er.event\_id

AND er.event\_request\_status\_id =

(SELECT ers.id FROM event\_request\_status AS ers WHERE ers.type = 'APPROVED')

WHERE

e.id = {@event-id}

GROUPBYe.id

* 1. При отсутствии в БД заявки с указанным {event-id} в базе данных возвращается код состояния 404 (NotFound)

1. Сервер формирует JSON-представление заявки (MAP-1) на основе результата запроса (REQ-1) согласно таблице ниже:

|  |  |  |
| --- | --- | --- |
| **МаппингMAP-1 (REQ-1.1) ->GetEventDto** | | |
| **Выходной объект** | **Источник** | **Комментарий** |
| **REQ-1** |  |
| id | e.id |  |
| organization.id | o.id |  |
| organization.name | o.name |  |
| start\_date | e.start\_date |  |
| duration | e.duration |  |
| direction | e.direction |  |
| price | e.price |  |
| max\_participants | e.max\_participants |  |
| rating | rating |  |
| free\_slots | free\_slots |  |
| description | e.description |  |

1. Сервер возвращает код состояния 200 и передает в теле ответа JSON, сформированный на шаге (7).

## GET /events/{event-id}/requests (role: owner) *Получение данных конкретного события для «владельца события» c информацией о заявках*

### Параметры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Вид параметра** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| event-id | Параметр пути | Да | Integer | Идентификатор запрашиваемого события. |

### Заголовки запроса

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование** | **Обяз.** | **Тип/формат значения** | **Назначение/комментарий** |
| Authorization | Да | Bearer + JWT в Base64 | Токен аутентификации |

### Тело запроса

|  |  |
| --- | --- |
| **Наличие/обязательность** | отсутствует |
| **Описание** | ‑ |

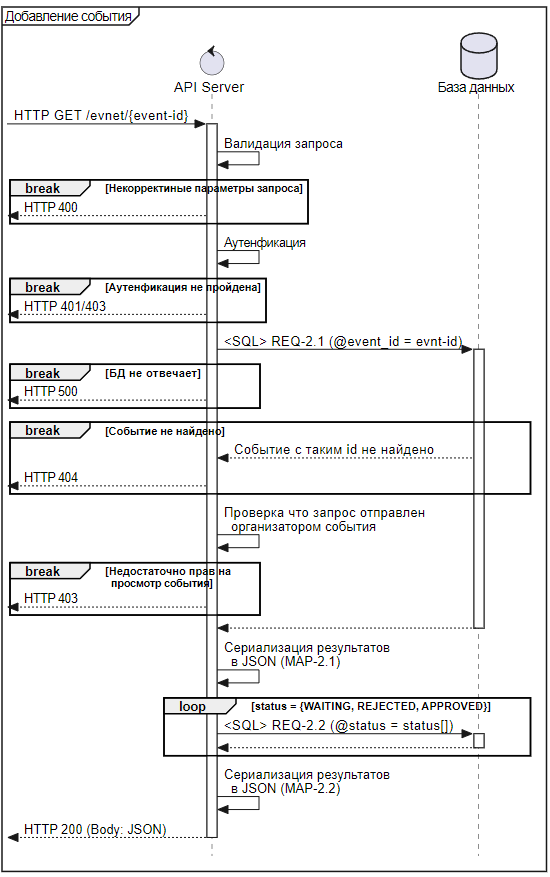
### Примеры запроса

|  |  |
| --- | --- |
| **Пример** | **Ответ** |
| GET event/1/request | HTTP 200 + JSON: |

#### Варианты ответа

|  |  |  |
| --- | --- | --- |
| **Код состояния** | **Описание** | |
| 200 | **Комментарий** | Событие с запрошенным ID найдено |
| **Тело ответа** | JSON-представление найденной заявки |
| 404 | **Комментарий** | Заявка с запрошенным IDне найдена |
| **Тело ответа** | JSON с детальной информацией об ошибке |
| 400 | **Комментарий** | Запрос содержит ошибку |
| **Тело ответа** | JSON с детальной информацией об ошибке |
| 401 | **Комментарий** | Отсутствует токен в заголовке Authorization |
| **Тело ответа** | JSON с детальной информацией об ошибке |
| 403 | **Комментарий** | Ошибка аутентификации или недостаточно прав |
| **Тело ответа** | JSON с детальной информацией об ошибке |
| 500 | **Комментарий** | Ошибка на стороне сервера |

### Сценарий информационного взаимодействия



1. Клиент отправляет запрос GET /event/{event-id}/request;
2. Сервер осуществляет проверку корректности запроса (наличие обязательных полей запроса, корректность значений полей запроса)
   1. В случае, если запрос некорректен, возвращается ошибка 400;
3. Сервер осуществляет проверку наличия токена аутентификации в заголовке «Authorization»;
   1. При отсутствии в запросе заголовка Authrozation токена аутентификации или наличия в нем содержимого, отличного от «Bearer + JWT в Base64», возвращается ошибка 401;
4. Сервер осуществляет верификацию токена и проверку выполнения хотя бы одного из следующих наличия у пользователя права на получения данных заявок;
   1. Если токен не верифицирован (срок действия истек либо подпись не подтверждена), либо у пользователя недостаточно прав, возвращается ошибка 403;
5. Сервер осуществляет подключение к серверу СУБД:
   1. При ошибке или невозможности подключения к базе данных возвращается код состояния 500 (InternalServerError)
6. Сервер осуществляет запрос (REQ-2.1) к базе данных:

SELECT

e.direction,

e.description,

e.start\_date,

e.duration,

COUNT(er.id)ASparticipants\_count,

e.max\_participants,

e.price

FROM events as e

LEFT JOIN events\_requests AS er ON e.id = er.event\_id

AND er.event\_request\_status\_id =

(SELECT ers.id FROM event\_request\_status AS ers WHERE ers.type = 'APPROVED')

WHERE e.id = {@event\_id}

GROUP BY e.id

* 1. При отсутствии в БД заявки с указанным {request-id} в базе данных возвращается код состояния 404 (NotFound)

1. Сервер формирует JSON-представление события (MAP-2.1) на основе результата запроса (REQ-2.1) согласно таблице ниже:

|  |  |  |
| --- | --- | --- |
| **Маппинг MAP-1 (REQ-2.1) ->GetEventWhithRequestDto** | | |
| **Выходной объект** | **Источник REQ-1** | **Комментарий** |
| id | e.id |  |
| start\_date | e.start\_date |  |
| duration | e.duration |  |
| direction | e.direction |  |
| price | e.price |  |
| participants\_count | participants\_count |  |
| max\_participants | e.max\_participants |  |
| description | e.description |  |

1. Сервер осуществляет запрос (REQ-2.2) к базе данных поочередно с разными типам переменной @status = { WAITING, REJECTED, APPROVED} :

SELECT

u.id,

CONCAT(u.first\_name, ' ' ,u.last\_name) AS participant

er.id

FROM users AS u

JOIN events\_requests AS er ONer.user\_id = u.id

JOIN event\_request\_status AS ers ON ers.id = er.event\_request\_status\_id

WHERE er.event\_id = {@event\_id}AND

ers.type = {@status}

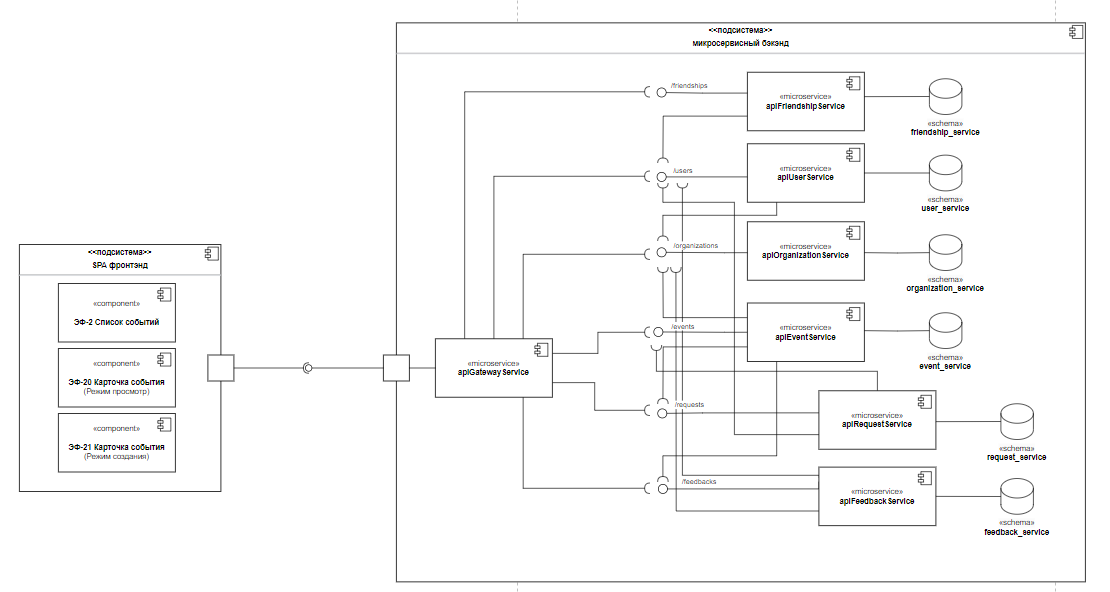
1. Сервер формирует вложенные списки запросов JSON-представление события (MAP-2.2) на основе результатов запросов (REQ-2.2) согласно таблице ниже:

|  |  |  |
| --- | --- | --- |
| **Маппинг MAP-2 (REQ-2.2) ->GetEventWhithRequestDto.{@staus}\_request[]** | | |
| **Выходной объект** | **ИсточникREQ-2.2** | **Комментарий** |
| request\_id | er.id |  |
| user\_id | u.id |  |
| fser\_fullname | participant |  |

1. Сервер возвращает код состояния 200 и передает в теле ответа JSON, сформированный на шагах(7).

Архитектура проекта

## Схема (Component Diagram)



## Диаграмма использования API GET /events/{event-id}

