

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра Информационные системы**

**ЛАБОРАТОРНАЯ РАБОТА**  
**по дисциплине «Архитектура информационных систем»**  
**Тема: Модуль «Мероприятия» для ЛК ЛЭТИ**

Студентка гр. 3374

Гренадерова А.А.

Студентка гр. 3374

Владимирская А.А.

Преподаватель

Ананьева В.Я.

Санкт-Петербург

2025

# **ЗАДАНИЕ**

## **НА ЛАБОРАТОРНУЮ РАБОТУ**

Студентка Владимирская А.А.

Студентка Гренадерова А.А.

Группа 3374

Тема работы: Модуль «Мероприятия» для ЛК ЛЭТИ

Исходные данные:

Средствами ПО Enterprise Architect спроектировать сайт с мероприятиями СПбГЭТУ «ЛЭТИ». Сформировать технические требования, архитектурное описание и список тестов для проекта.

Содержание пояснительной записи:

Введение, требования, архитектурное описание, архитектурное обоснование, модели, UML описание, use case, классы, активности, размещение, тесты, заключение, список используемых источников.

Предполагаемый объем пояснительной записи:

Не менее 15 страниц.

Дата выдачи задания: 10.02.2025

**Дата сдачи реферата:**

**Дата защиты реферата:**

Студентка

Гренадерова А.А.

Студентка

Владимирская А.А.

Преподаватель

Ананьева В.Я.

# Содержание

## Оглавление

<b>Содержание .....</b>	3
<b>1. Требования к системе .....</b>	4
1.1. Преамбула .....	4
1.2. Введение .....	4
1.3. Глоссарий .....	5
1.4. Пользовательские требования .....	6
1.5. Описание системной архитектуры .....	7
1.6. Системные требования .....	8
Функциональные требования .....	8
Нефункциональные требования .....	12
1.7. Системные модели .....	13
1.8. Возможные направления эволюции .....	13
1.9. Приложения .....	13
1.10. Индексы .....	14
<b>2. Архитектурное обоснование .....</b>	15
2.1. Выбор технологического стека .....	15
2.2. Backend: баланс между скоростью и надежностью .....	15
2.3. Интеграции и безопасность .....	15
2.4. Клиент-серверная архитектура и API .....	15
2.5. Производительность и отказоустойчивость .....	16
2.6. Масштабируемость и пользовательский опыт .....	16
<b>3. Макет 0-го уровня .....</b>	17
<b>4. Архитектурное описание на UML .....</b>	24
4.1. Диаграмма вариантов использования (Use Case Diagram) .....	24
4.3 Диаграмма классов (Class Diagram) .....	38
4.3 Диаграмма деятельности (Activity diagram) .....	42
4.4 Диаграмма конечного автомата (State diagram) .....	44
<b>5. Тесты .....</b>	46

# **1. Требования к системе**

## **1.1. Преамбула**

**Документ:** Техническое задание на разработку модуля "Мероприятия" для ЛК вуза

**Версия:** 1.0

**Дата утверждения:** [текущая дата]

**Авторы:** Гренадерова А.А., Владимирская А.А.

**Статус:** на согласовании

**Целевая аудитория:**

- Разработчики программного обеспечения
- Администраторы информационных систем вуза
- Руководители студенческих организаций
- Технические специалисты вуза

**Связанные документы:**

- Техническая политика информационной безопасности вуза
- Стандарты разработки ПО вуза
- API документация ETU ID

## **1.2. Введение**

Модуль «Мероприятия» будет создан, чтобы упростить организацию мероприятий в вузе. Он помогает автоматизировать процессы, которые раньше требовали ручной работы: регистрацию участников, учет посещаемости и анализ активности студентов. Благодаря этому преподаватели и администрация смогут быстрее планировать события, а студенты — удобнее находить и записываться на интересные мероприятия.

Система работает на основе ролей, чтобы у каждого пользователя был доступ только к нужным функциям. Например, студенты видят анонсы и могут записываться, организаторы — управлять мероприятиями своих отделов и центров, а администрация — следить за статистикой. Вход осуществляется через ETU ID, поэтому не нужно запоминать новые логины и пароли.

Модуль также берет на себя рутинные задачи: отправляет уведомления о предстоящих событиях, напоминает о регистрации и собирает данные для отчетов. Это экономит время и снижает риск ошибок. В итоге вуз получает удобный инструмент, который делает мероприятия более прозрачными, а студентов — более вовлеченными.

## **Бизнес-требования**

- Централизованный учет всех мероприятий, проводимых в вузе.
- Упрощение процесса регистрации участников и автоматизация учета посещаемости.
- Проведение детального анализа активности студентов для эффективного планирования будущих мероприятий.
- Повышение вовлеченности студентов и улучшение учебного процесса.

## **1.3. Глоссарий**

### **Основные понятия**

- *Мероприятие* – событие, организованное вузом с ID, названием, описанием, датой проведения, ответственным отделом, целевой аудиторией и анонсом.
- *Участник* - пользователь, записавшийся на мероприятие, указав свои данные, в том числе данные, интегрированные из базы данных вуза. Система фиксирует посещение и сохраняет историю участия.
- *Персональные данные* - информация о пользователе, получаемая через API (ФИО, учебная группа), и вводимая пользователем (контактные данные, аватарка).
- *Интерфейс* – совокупность возможностей, способов и методов одновременного действия.
- *База данных* – совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, причем такое собрание данных, которое поддерживает одну или более областей применения.
- *Пользователь* – человек, просматривающий сайт, посетитель мероприятий или организатор.

### **Ролевая модель**

1. Guest (Гость)
  - Вход в систему
  - Регистрация в системе
  - Просмотр всех мероприятий

2. User (Пользователь)

- Просмотр всех мероприятий
- Запись на мероприятие
- Просмотр всех мероприятий, на которые записан
- Возможность записи/отмены записи
- Просмотр и редактирование профиля
- Доступ к персональной статистике

3. Editor (Организатор):

- Создание и редактирование мероприятий отдела
- Просмотр списка участников
- Назначение новых организаторов

4. Admin (Руководитель отдела):

- Создание и редактирование мероприятий отдела
- Просмотр списка участников
- Назначение и удаление Editors

5. Creator (Разработчик)

- Все вышеперечисленные права.
- Назначение руководителей отделов

#### **1.4. Пользовательские требования**

Для пользователей, которые **посещают мероприятия**:

- *Поиск мероприятий* – студенты могут легко находить события, фильтровать их в поиске по направлениям (наука, творчество, спорт, волонтерство) и дате.
- *Регистрация на мероприятие* – студент может заранее зарегистрироваться на интересующее мероприятие, затем может посмотреть в личном кабинете мероприятия, на которые он зарегистрирован.
- *Учет активности* – участие в мероприятиях фиксируется в личном кабинете (для портфолио, стипендий, рейтинга).

Для **организаторов мероприятий**:

- *Размещение мероприятий на сайте* – пользователь может загрузить на сайт информацию о мероприятии.
- *Учет участников* – организатор видит, сколько людей зарегистрировалось на мероприятие.
- *Аналитика* – организатор видит статистику по мероприятиям, количество участников прошедших и будущих мероприятий.

- Возможность добавлять других организаторов – для удобства организации мероприятий пользователь сможет назначать другим пользователям права редактора мероприятий.

## **1.5. Описание системной архитектуры**

### **1. Frontend-часть:**

- Основной интерфейс (React.js) – для динамичного и отзывчивого UI.
- Административная панель (Angular) – для удобного управления контентом и пользователями.
- Мобильная адаптация (CSS3, React Native) – поддержка смартфонов и планшетов.

### **2. Backend-часть:**

- Ядро системы (Node.js) – обработка бизнес-логики.
- API сервер (RESTful + GraphQL) – гибкий обмен данными между клиентом и сервером.
- База данных (PostgreSQL) – надежное хранение структурированных данных.
- Кэширование (Redis) – ускорение частых запросов (например, списка мероприятий).

### **3. Интеграции:**

- ETU ID (OAuth 2.0) – безопасная аутентификация через университетскую систему.
- Email-сервис (SendGrid / Nodemailer) – автоматические уведомления пользователей.
- Календарный сервис (Google Calendar API) – синхронизация событий.
- Telegram-бот – для оповещений через телеграм.

### **API-спецификация (дополненная):**

GET /api/events – получение списка мероприятий (с фильтрами).

POST /api/events – создание мероприятия (только для Editor/Admin).

GET /api/users/{id} – информация о пользователе (доступ ограничен).

POST /api/events/{id}/register – запись на мероприятие.

GET /api/stats – статистика по активности (для организаторов).

## **1.6. Системные требования**

### **Функциональные требования**

#### **1. Для посетителя мероприятий**

*Для гостя (Guest):*

- Первичная авторизация посетителя через ETU ID
  - Система отображает страницу входа с полями ввода логина и пароля ETU ID.
  - Система сравнивает введенные пользователем данные с базой данных ETU ID и выводит сообщение в случае успешной или неуспешной авторизации.
  - Если данные введены успешно, то система создает учетную запись пользователя, присваивает роль User и отображает предыдущую страницу (с которой пользователь попал на страницу регистрации).
- Если пользователь уже был авторизован ранее, то:
  - Система отображает страницу входа с полями ввода логина и пароля ETU ID
  - Система сравнивает введенные пользователем данные с базой данных ETU ID и выводит сообщение в случае успешной и неуспешной авторизации
  - Если данные введены успешно, то система загружает профиль пользователя и отображает предыдущую страницу.
- Авторизованный пользователь или посетитель сайта может просматривать список всех мероприятий вуза:
  - Список мероприятий разбит на два блока - самые популярные (слайдер из пяти афиш вверху страницы) и список всех мероприятий.
  - Список всех мероприятий можно отфильтровать по дате и по направлению (наука, волонтерство, культура, спорт).
  - Список отображается в формате карточек с краткой информацией: название, афиша, дата, направление.
  - Перейдя по карточке мероприятия (в первом блоке – кнопка "подробнее"/ во втором блоке можно нажать на афишу или на кнопку "зарегистрироваться"), пользователь переходит на страницу мероприятия с подробной информацией.

- Страница мероприятия содержит следующую информацию: основная информация (организация, которая проводит мероприятие (например, волонтерский центр), дата и время мероприятия, место проведения мероприятия; контакты для связи с организаторами (например, телефон и почта); описание (текст с описанием мероприятия); афиша мероприятия; количество зарегистрировавшихся участников; название мероприятия.

*Для пользователя (User):*

- Просмотр мероприятий.
- Запись на мероприятия - на страницах мероприятий есть кнопка "зарегистрироваться", которая отобразит блок для ввода личных данных с возможностью автозаполнения по данным из профиля. После ввода данных система выведет сообщение о успешной/неуспешной регистрации и закроет блок регистрации.
- Пользователь имеет возможность зайти в личный кабинет (в верхней части есть кнопка, по которой можно перейти в ЛК), который содержит информацию о пользователе:
  - Просмотр списка мероприятий на которых он записан, а также возможность отменить регистрацию
  - Список мероприятий, на которые пользователь записан также отображаются в формате карточек (как на главной странице сайта). Чтобы отменить запись или получить более подробную информацию о мероприятии пользователь может также перейти по карточке мероприятия.
  - Откроется страница мероприятия, где на месте кнопки "зарегистрироваться" будет кнопка "отменить регистрацию". Если пользователь ее нажмет, то откроется диалоговое окно "Вы уверены?". Если пользователь подтвердит действие, то он перестанет быть зарегистрирован на мероприятие, и оно перестанет отображаться в личном кабинете.
  - Просмотр собственной статистики посещений в формате счетчика (сколько мероприятий запланировано к посещению, сколько мероприятий уже посещено, сколько мероприятий посещено в этом году).
  - Помимо списка запланированных мероприятий, пользователь видит и список мероприятий, на которые он был записан, и они уже прошли.

есть, список посещенных мероприятий. Он отображается также в формате карточек с возможностью перехода на страницу мероприятия, но в этом случае на странице мероприятия не будет кнопок "зарегистрироваться" или "отменить регистрацию". Пользователь увидит только информацию по мероприятию.

- Профиль можно редактировать:

➤ Профиль содержит персональные данные. Если пользователь нажмет на кнопку "Редактировать профиль", то он может изменить все параметры (кроме ФИО, номера группы и роли). О пользователе содержится следующая информация: ФИО, номер группы, аватарка (ее устанавливает пользователь при желании), роль (пользователь, организатор мероприятий, руководитель отдела, разработчик), электронная почта, номер телефона.

## **2. Для организаторов мероприятий**

*Для организатора (Editor):*

- Пользователь с правами редактора (организатор мероприятий или руководитель отдела) имеет возможность управления мероприятиями своего подразделения. Для перехода на страницу управлением мероприятиями в личном кабинете есть соответствующая кнопка "Управление мероприятиями", открывающая контекстное меню с действиями: Создание мероприятия, Изменение и просмотр мероприятий, Редакторы отдела.
  - Создание мероприятия - открывается страница мероприятия в режиме создания, где можно ввести параметры мероприятия (они указаны выше, в описании страницы мероприятия). После ввода параметров можно сохранить мероприятие - оно добавляется в общую базу мероприятий. Выводится сообщение о успешной/неуспешной публикации и открывается страница мероприятия в режиме просмотра.
  - Изменение и просмотр мероприятий открывает список мероприятий отдела в формате карточек, где можно посмотреть какие мероприятия доступны к редактированию, т.е. мероприятия подразделения. Можно перейти на карточку мероприятия и откроется страница мероприятия в режиме редактирования (соответствует режиму создания,

только в полях ввода параметров уже есть значения, которые можно изменить). После внесения изменений мероприятие можно сохранить, и аналогично созданию, выводится сообщение об успешных или неуспешных внесенных изменениях, затем в базе мероприятий изменяются отредактированные параметры события.

- Также на странице мероприятия в режиме редактирования можно перейти к участникам мероприятия. При переходе на этот раздел система отображает список участников, зарегистрированных на мероприятия.
- В режиме "Редакторы отдела" можно увидеть список редакторов, принадлежащих подразделению. По кнопке "Добавить редактора" откроется поиск (идентичен поиску людей в разделе сообщений Личного кабинета ЛЭТИ), где можно найти студента или преподавателя по ФИО и добавить в редакторы отдела.

*Для руководителя отдела (Admin):*

- Руководитель отдела имеет возможности создания и редактирования мероприятий.
  - В личном кабинете, вместо мероприятий, которые он собирается посетить отображаются запланированные и прошедшие мероприятия раздела. Также в разделе статистика счётчик отображает количество мероприятий запланированных, проведенных и проведенных в этом году.
- Руководитель, помимо возможности добавления новых редакторов имеет возможность снимать с должности уже действующих редакторов. Для более удобного использования этой функции в личном кабинете есть кнопка.

*Для разработчика (Creator):*

- Разработчик имеет все вышеперечисленные возможности. Страница личного кабинета аналогична странице личного кабинета руководителя отдела, но в запланированных мероприятиях отображаются все мероприятия из базы данных.
- Разработчик может назначить и снять с должности не только редакторов, но и руководителей.

## **Нефункциональные требования**

### ***1. Производительность:***

- Время отклика системы при просмотре мероприятий не должно превышать 2 секунд.
- Система должна выдерживать до 1000 одновременных пользователей.

### ***2. Удобство использования:***

- Интерфейс должен быть понятным для всех категорий пользователей: для регистрации на мероприятие на сайте нужно провести 5 минут.
- Поддержка мобильных устройств (адаптивный дизайн).

### ***3. Надежность:***

- Система должна быть доступна 99,9% времени.
- Автоматическое сохранение данных при внезапном закрытии приложения.

### ***4. Безопасность:***

- Обязательная аутентификация для доступа к личным данным.
- Шифрование передаваемых данных (HTTPS).
- Защита от SQL-инъекций и других уязвимостей.
- Логирование действий пользователей для анализа проблем.

### ***5. Масштабируемость:***

- Возможность добавления новых функциональных модулей без переписывания кода.
- Поддержка увеличения количества пользователей и мероприятий.

### ***6. Аналитика:***

- Просмотр статистики по посещению мероприятий (количество посещенных и запланированных мероприятий отображается в профиле в формате счетчика).

## 1.7. Системные модели

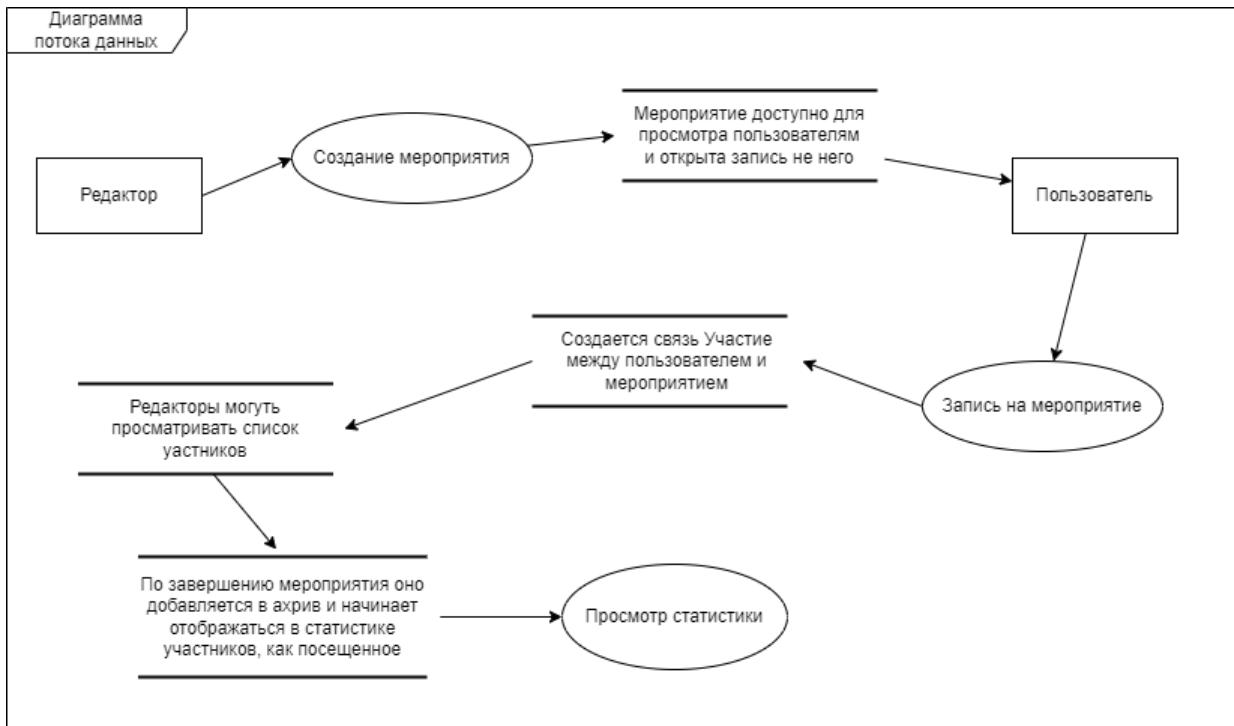


Диаграмма 1 Диаграмма потока данных

## 1.8. Возможные направления эволюции

1. Интеграция с мобильным приложением вуза
2. Система рейтинга мероприятий
3. Механизм рекомендаций на основе ИИ
4. Интеграция с системой баллов НСО
5. Оффлайн-регистрация через QR-коды

## 1.9. Приложения

### Приложение А. Примеры данных

```
{  
  "event": {  
    "id": 123,  
    "title": "Научная конференция",  
    "date": "2024-11-15T10:00:00",  
    "tag": "Научный отдел",  
    "description": "Ежегодная конференция...",  
    "image": "conf2024.jpg"  
    "status": "Завершено"  
  }  
}
```

## **Приложение Б. Политика безопасности**

1. Все запросы авторизуются через ETU ID
2. Персональные данные не хранятся локально
3. Резервное копирование ежедневно

## **1.10. Индексы**

### **1. Алфавитный указатель**

- Участник – стр. 4, 5, 6, 8, 10
- Мероприятие – стр. 5, 6, 7, 8, 9, 10
- Персональные данные – стр. 5, 9
- Интерфейс – стр. 5, 6, 11
- База данных – стр. 5, 7
- Пользователь – стр. 5, 6, 7, 8, 9

### **3. Список иллюстраций:**

<i>Рис. 1 - основная страница.....</i>	17
<i>Рис. 2 – страница входа .....</i>	18
<i>Рис. 3 – карточка мероприятия 1 .....</i>	18
<i>Рис. 4 – карточка мероприятия 2 .....</i>	19
<i>Рис. 5 – личный кабинет 1 .....</i>	20
<i>Рис. 6 – личный кабинет 2 .....</i>	21
<i>Рис. 7 – личный кабинет 3 .....</i>	22
<i>Рис. 8 – личный кабинет 4 .....</i>	23

### **2. Список диаграмм:**

<i>Диаграмма 1 Диаграмма потока данных .....</i>	13
<i>Диаграмма 2 - Диаграмма вариантов использования .....</i>	25
<i>Диаграмма 3 - Диаграмма активности .....</i>	43
<i>Диаграмма 4 - Диаграмма конечного автомата.....</i>	45

## **2. Архитектурное обоснование**

### **2.1. Выбор технологического стека**

Для фронтенда выбран React.js из-за его высокой производительности, компонентного подхода и богатой экосистемы библиотек. React позволяет создавать динамичные интерфейсы с минимальными затратами ресурсов, что критично для пользовательского опыта. Административная панель реализована на Angular, так как этот фреймворк обеспечивает строгую типизацию и удобен для сложных форм управления. Для мобильной адаптации используется CSS3 и React Native, что гарантирует корректное отображение на любых устройствах.

### **2.2. Backend: баланс между скоростью и надежностью**

В качестве backend-решения выбран Node.js, поскольку он обеспечивает высокую скорость разработки и поддерживает асинхронные операции, что важно для обработки запросов множества пользователей. PostgreSQL в качестве базы данных гарантирует надежное хранение структурированных данных и поддержку сложных запросов. Для ускорения работы системы добавлен Redis, который кэширует частые запросы (например, списки мероприятий) и снижает нагрузку на основную БД.

### **2.3. Интеграции и безопасность**

Аутентификация реализована через ETU ID (OAuth 2.0), что исключает необходимость хранения паролей и повышает безопасность. Для уведомлений используется SendGrid или Nodemailer – эти решения надежны и легко масштабируются. Интеграция с Google Calendar API позволяет пользователям синхронизировать мероприятия с личными календарями, улучшая удобство системы. Встроенный в сайт telegram-бот позволит получать удобные для пользователя уведомления прямо в популярном мессенджере.

### **2.4. Клиент-серверная архитектура и API**

Применение RESTful API и GraphQL обеспечивает гибкость: клиент может запрашивать только нужные данные, уменьшая нагрузку на сервер. Архитектура спроектирована с учетом возможного перехода на микросервисы – например, при росте нагрузки можно выделить аутентификацию, управление мероприятиями и аналитику в отдельные сервисы.

## **2.5. Производительность и отказоустойчивость**

Система использует HTTPS для шифрования данных и JWT для безопасной аутентификации API. Регулярное резервное копирование БД предотвращает потерю информации. Асинхронная обработка задач (через встроенные механизмы Node.js) ускоряет работу при пиковых нагрузках.

## **2.6. Масштабируемость и пользовательский опыт**

Контейнеризация (Docker) упрощает развертывание системы и ее масштабирование. Адаптивный интерфейс работает на любых устройствах, а push-уведомления (при интеграции с Firebase) делают взаимодействие с системой более удобным.

Данная архитектура обеспечивает:

- Гибкость (легкое добавление новых функций),
- Надежность (кэширование, резервные копии, отказоустойчивость),
- Безопасность (OAuth 2.0, JWT, HTTPS),
- Масштабируемость (готовность к росту числа пользователей).

Это делает систему стабильной, удобной для пользователей и простой в поддержке.

### 3. Макет 0-го уровня

Макет 0-го уровня представляет собой примерный макет проекта. Ниже приложены сами макеты с комментариями.

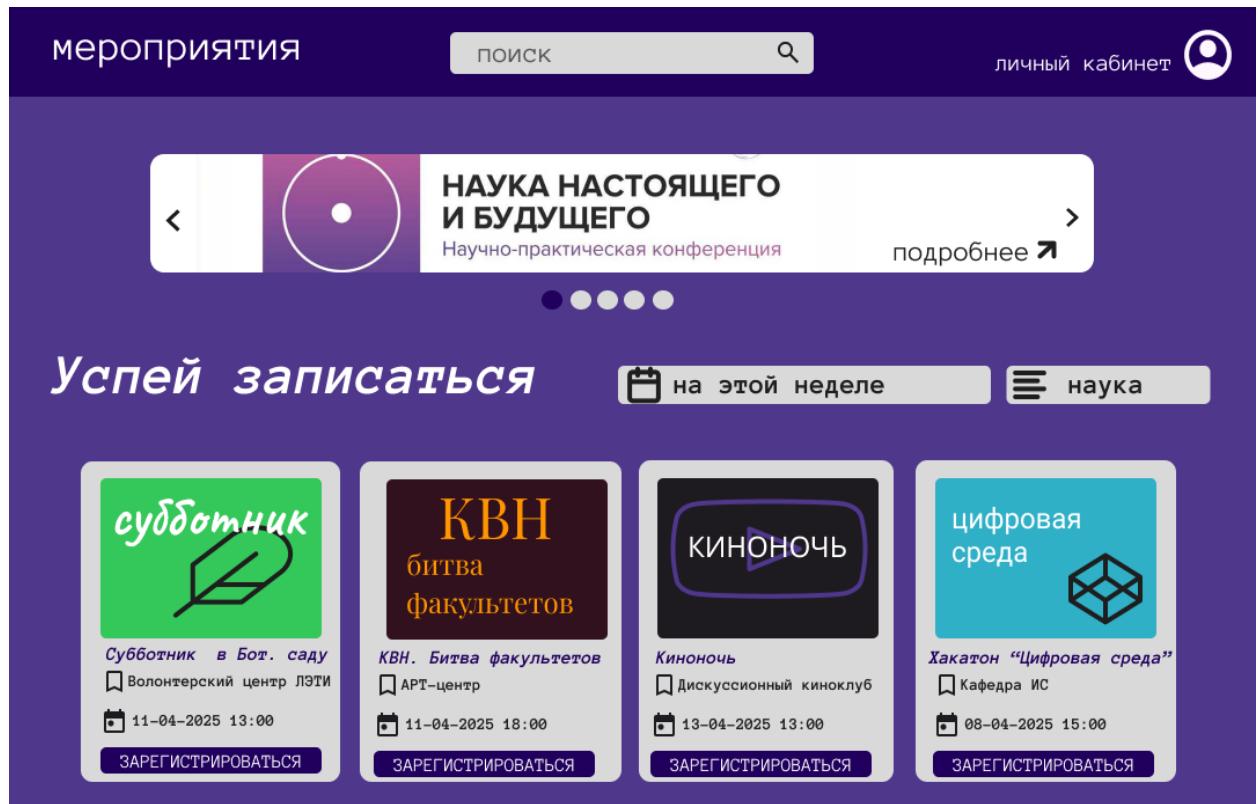
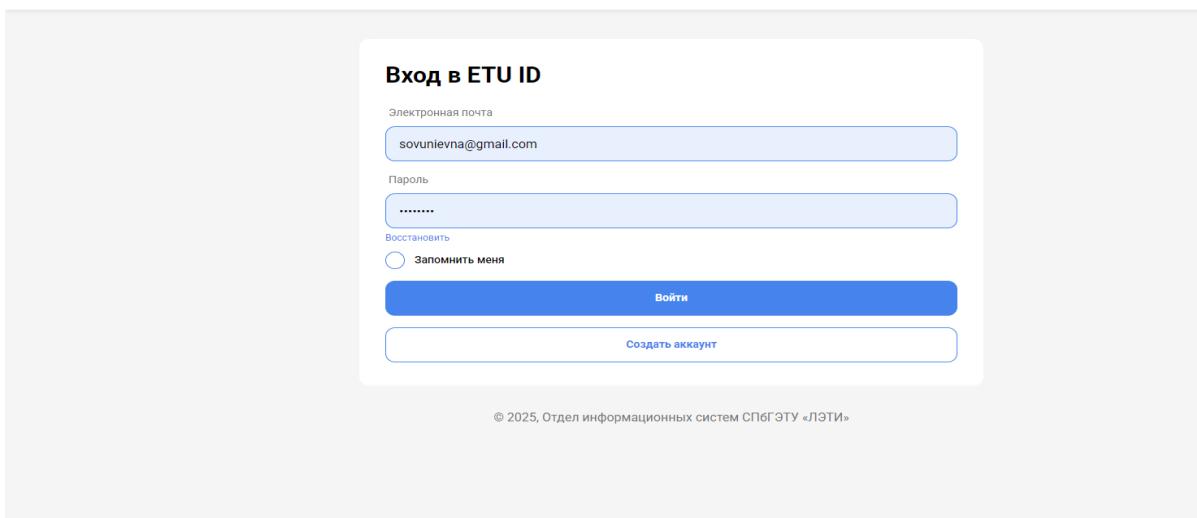


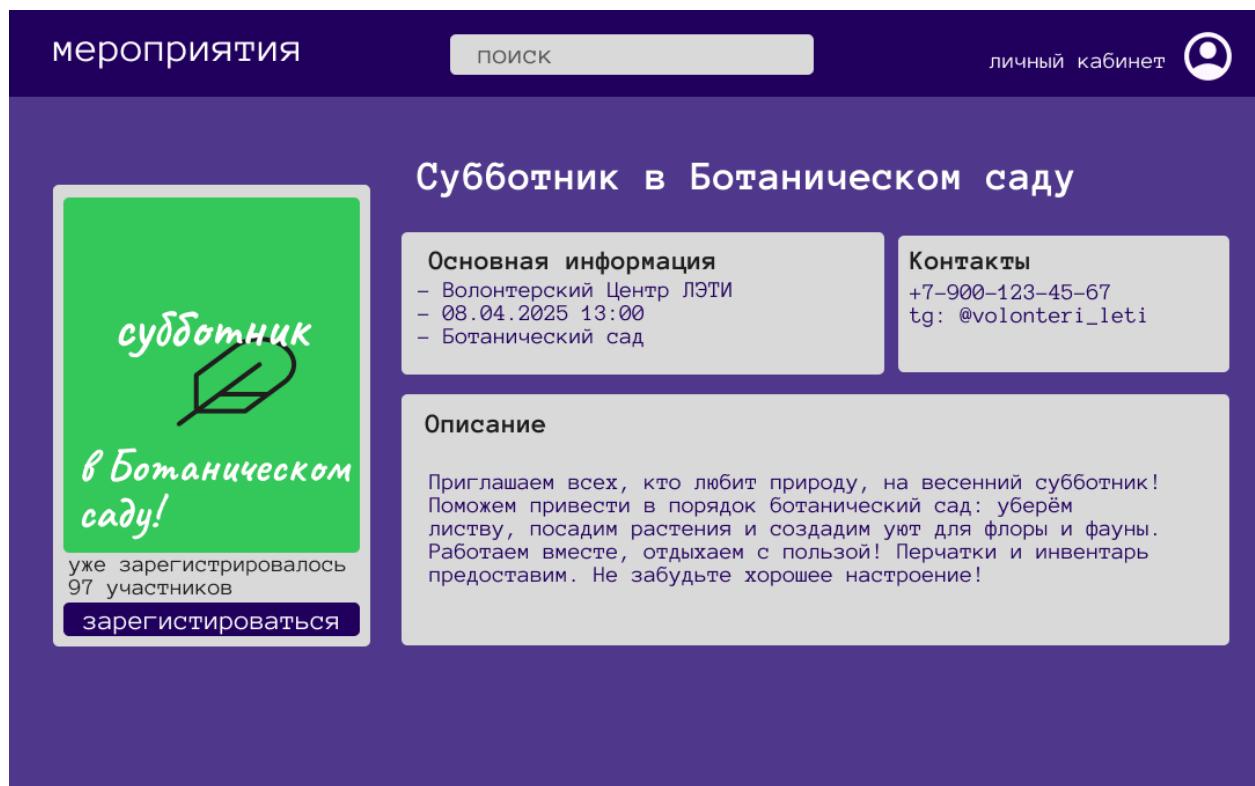
Рис. 1 - основная страница

На рис. 1 изображена основная страница сайта. На ней пользователь видит список мероприятий. В верхней части, там, где слайдер отображаются самые популярные мероприятия. Ниже списком с карточками можно увидеть список всех мероприятий. Можно воспользоваться фильтром по направлению (волонтерство, арт-центр, наука) и фильтром по дате. Пользователь может нажать кнопку «Зарегистрироваться» или сразу на кнопку входа в профиль (на верхней панели справа), и она отправит его на страницу входа.



*Рис. 2 – страница входа*

На рис. 2 изображена страница входа. Она перенаправляет пользователя на ETU ID, чтобы совершить вход через эту платформу. Когда пользователь вошел, он может зарегистрироваться на мероприятие.

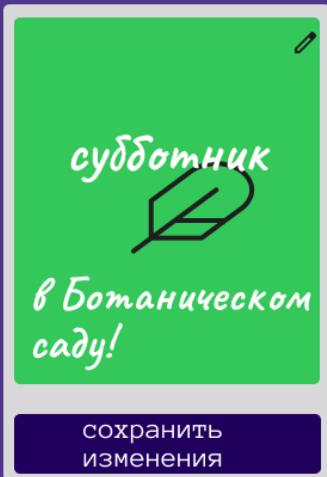


*Рис. 3 – карточка мероприятия 1*

На рис. 3 изображен макет карточки мероприятия. На нем расположена информация о мероприятии, а также доступна функция зарегистрироваться на мероприятие.



## Субботник в Ботаническом саду



сохранить изменения

## Основная информация /

Организация Волонтерский Центр ЛЭТИ  
Дата 08.04.2025 13:00  
Место Ботанический сад

## Контакты /

+7-900-123-45-67  
tg: @volonteri\_leti

## Описание /

Приглашаем всех, кто любит природу, на весенний субботник! Поможем привести в порядок ботанический сад: уберём листву, посадим растения и создадим уют для флоры и фауны. Работаем вместе, отдыхаем с пользой! Перчатки и инвентарь предоставим. Не забудьте хорошее настроение!

посмотреть список участников

Рис. 4 – карточка мероприятия 2

На рисунке 4 представлен макет страницы мероприятия в режиме редактирования. В этом режиме редактор, руководитель отдела или разработчик может изменять параметры мероприятия, а также посмотреть список участников мероприятия по соответствующей кнопке.

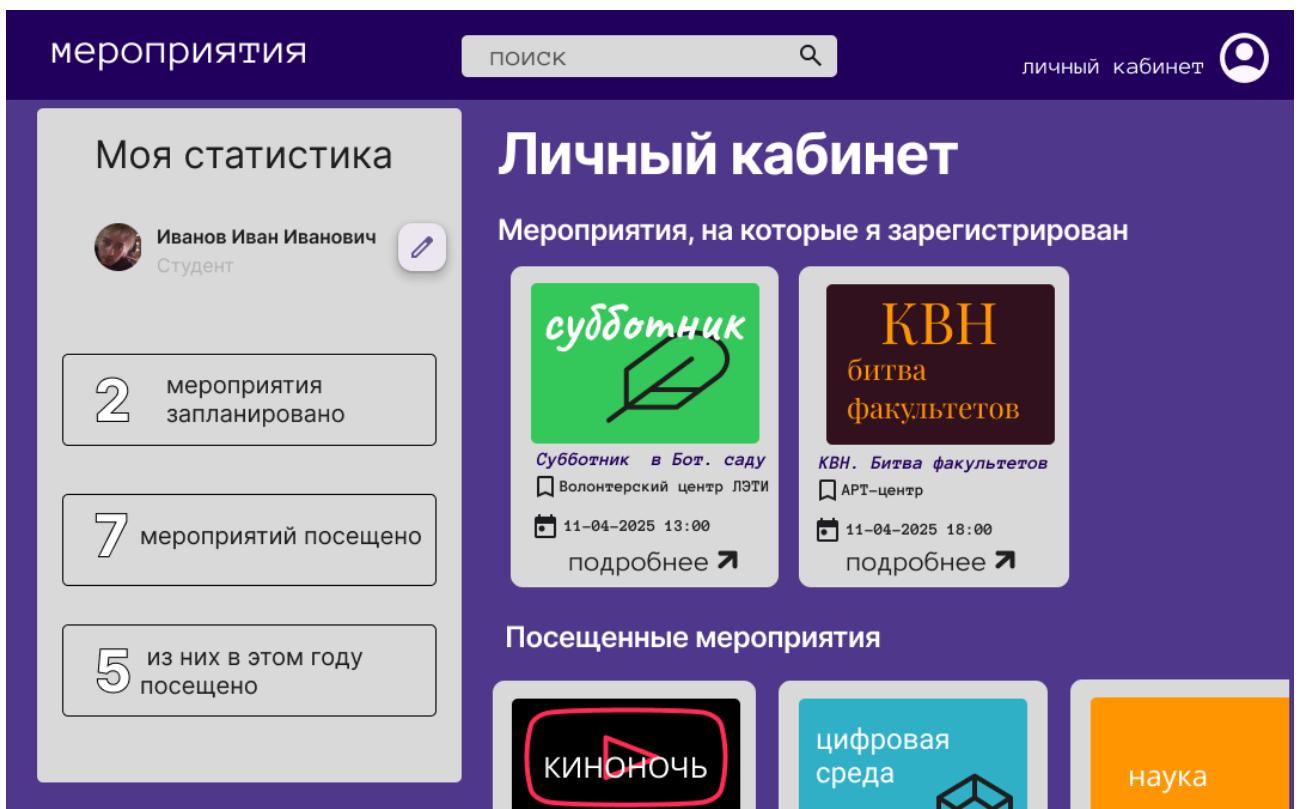


Рис. 5 – личный кабинет 1

На рис. 5 изображен макет личного кабинета пользователя. Здесь можно увидеть статистику (в левой части страницы) и отдельно мероприятия, которые пользователь посетил или планирует посетить. Также доступна функция редактировать профиль, где можно сменить аватарку и персональные данные.

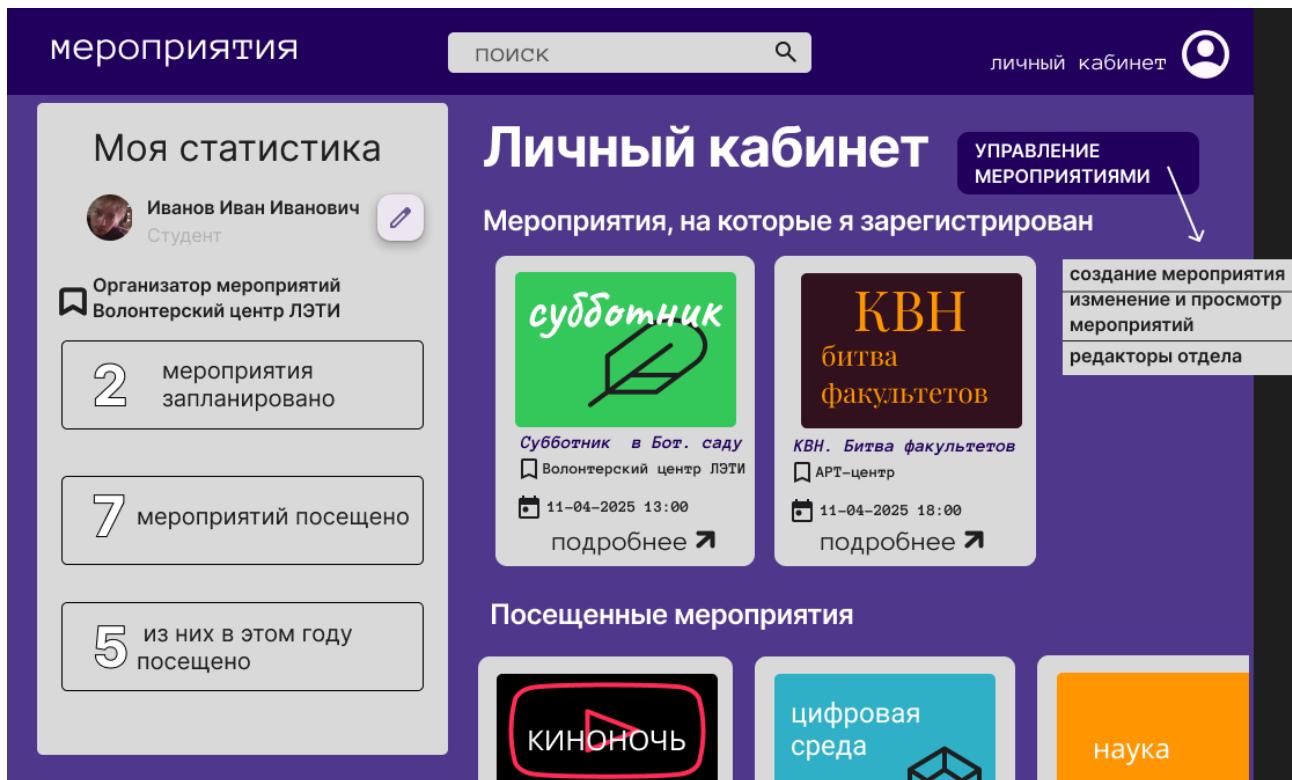


Рис. 6 – личный кабинет 2

На рисунке 6 представлен макет личного кабинета редактора отдела (организатора мероприятий). Отличие от ЛК пользователя в том, что здесь есть кнопка, открывающая контекстное меню управления мероприятий. «Создание мероприятия» открывает страницу мероприятия в режиме создание (аналогично странице мероприятия в режиме редактирования), изменение и просмотр мероприятий открывает список мероприятий отдела, а кнопка редакторы отдела открывает доступ к списку редакторов отдела, где можно помимо просмотра редакторов назначить новых. Также в блоке статистике отображается соответствующая должность и название отдела.

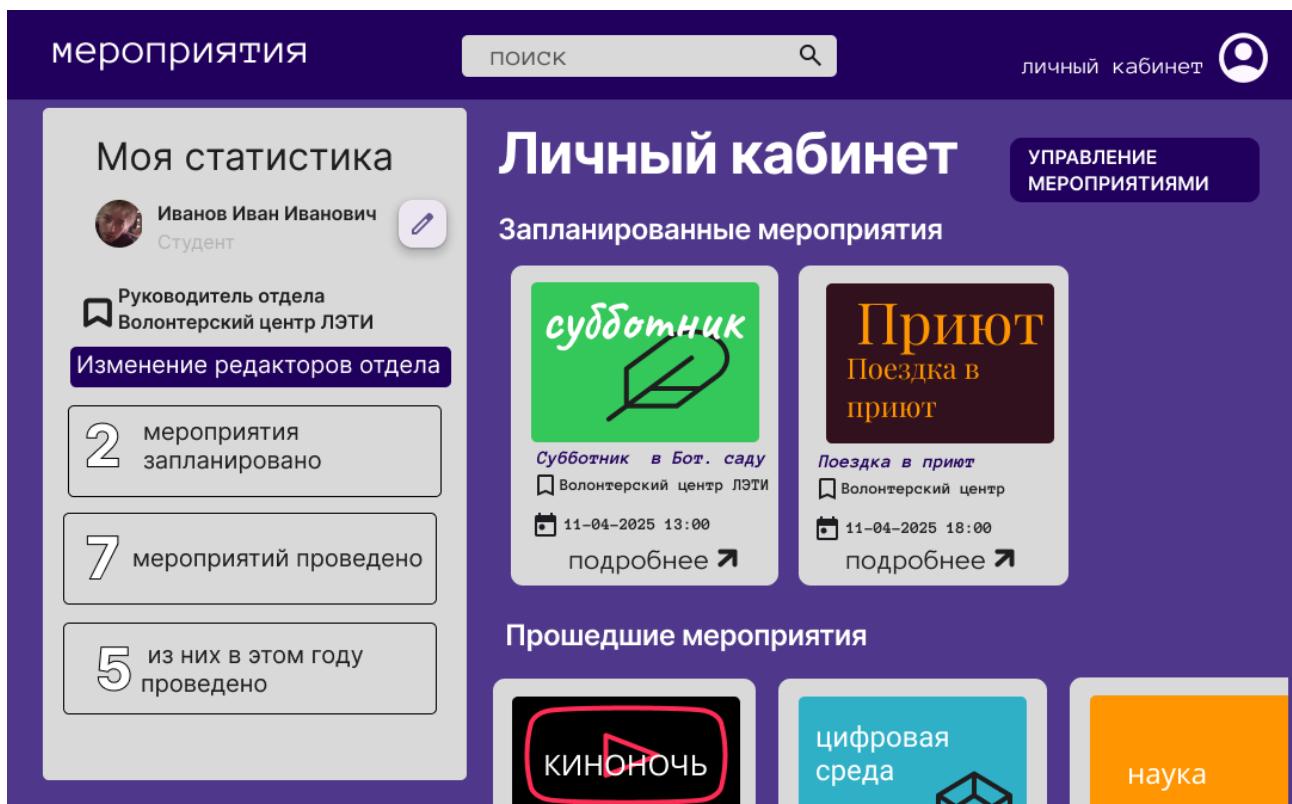


Рис. 7 – личный кабинет 3

На рисунке 7 представлен макет личного кабинета руководителя отделов. Отличие от предыдущих макетов, в том, что вместо мероприятий, планируемых к посещению, указаны мероприятия, которые планирует проводить отдел. Также отдельно выведена кнопка Изменение редакторов отдела (к этой функции можно перейти также через контекстное меню). Помимо добавления новых редакторов отдела руководитель имеет возможность снять редакторов с должности.

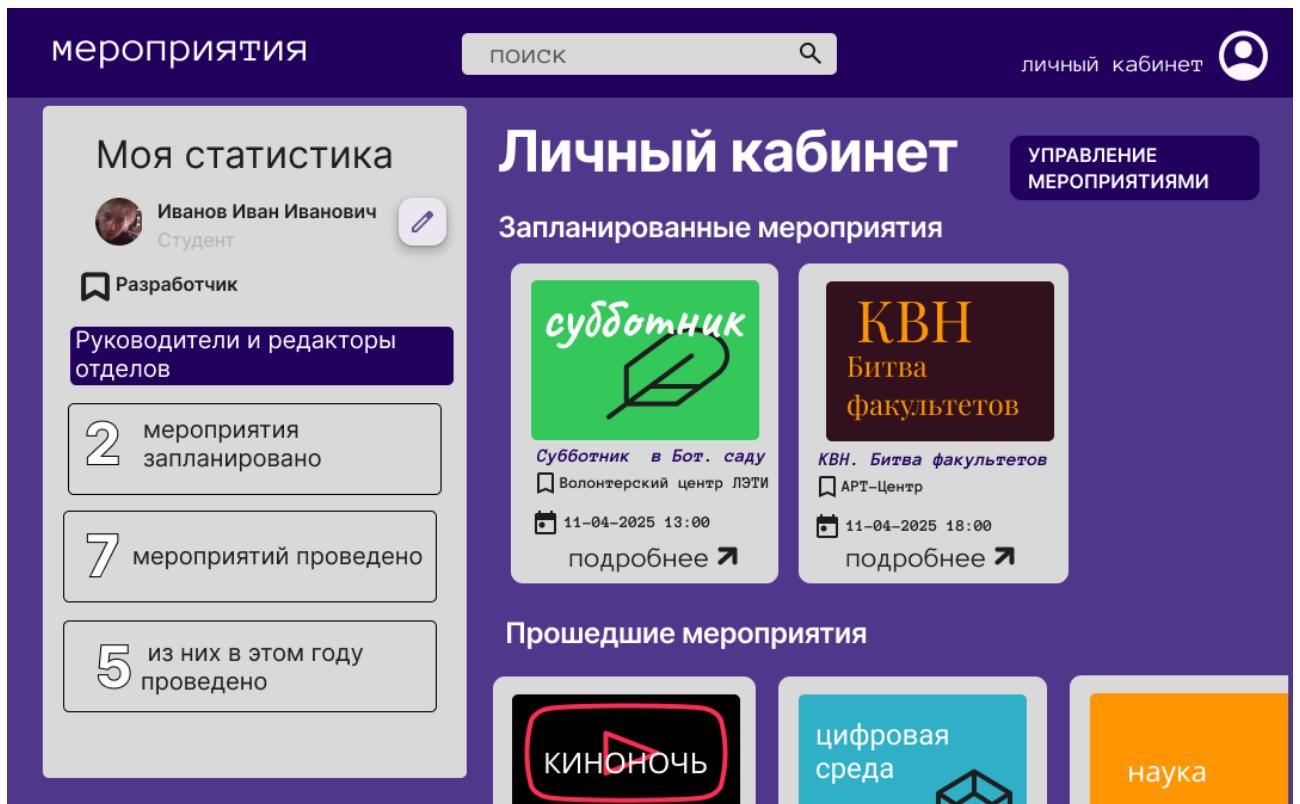


Рис. 8 – личный кабинет 4

На рисунке 4.3 представлен макет личного кабинета разработчика. Он аналогичен личному кабинету Руководителя отдела, но в этом случае вместо мероприятий отдельного раздела в блоках запланированные и прошедшие мероприятия отображаются все мероприятия из базы данных мероприятий. Также вместо кнопки Изменение редакторов отделов располагается кнопка Руководители и редакторы отделов, с аналогичными функциями, но помимо назначения и разжалования редакторов можно также назначить и разжаловать руководителей отделами.

## 4. Архитектурное описание на UML

### 4.1. Диаграмма вариантов использования (Use Case Diagram)

Пользователь – посетитель мероприятия или организатор.

Пользователи системы могут иметь одну из 5-ти ролей:

- 1) **Guest** (Гость)- неавторизированный пользователь. Варианты деятельности: Вход в систему, Регистрация в системе, Просмотр всех мероприятий.
- 2) **User** (Пользователь) - авторизированный пользователь, студент. Варианты деятельности: Просмотр всех мероприятий, Запись на мероприятие, Просмотр всех мероприятий, на которые записан, Возможность записи/отмены записи, Просмотр и редактирование профиля, Доступ к персональной статистике
- 3) **Editor** (Редактор) – организатор мероприятий. Варианты деятельности: Создание и редактирование мероприятий отдела, Просмотр списка участников, Назначение новых Editor, Все возможности User.
- 4) **Admin** (Администратор) - руководитель отдела. Варианты деятельности: Создание и редактирование мероприятий отдела, Просмотр списка участников, Назначение и удаление Editor.
- 5) **Creator** (Разработчик). Варианты деятельности: Все вышеперечисленные права, Назначение руководителей отделов

## Диаграмма вариантов использования для системы «Модуль “Мероприятия”»

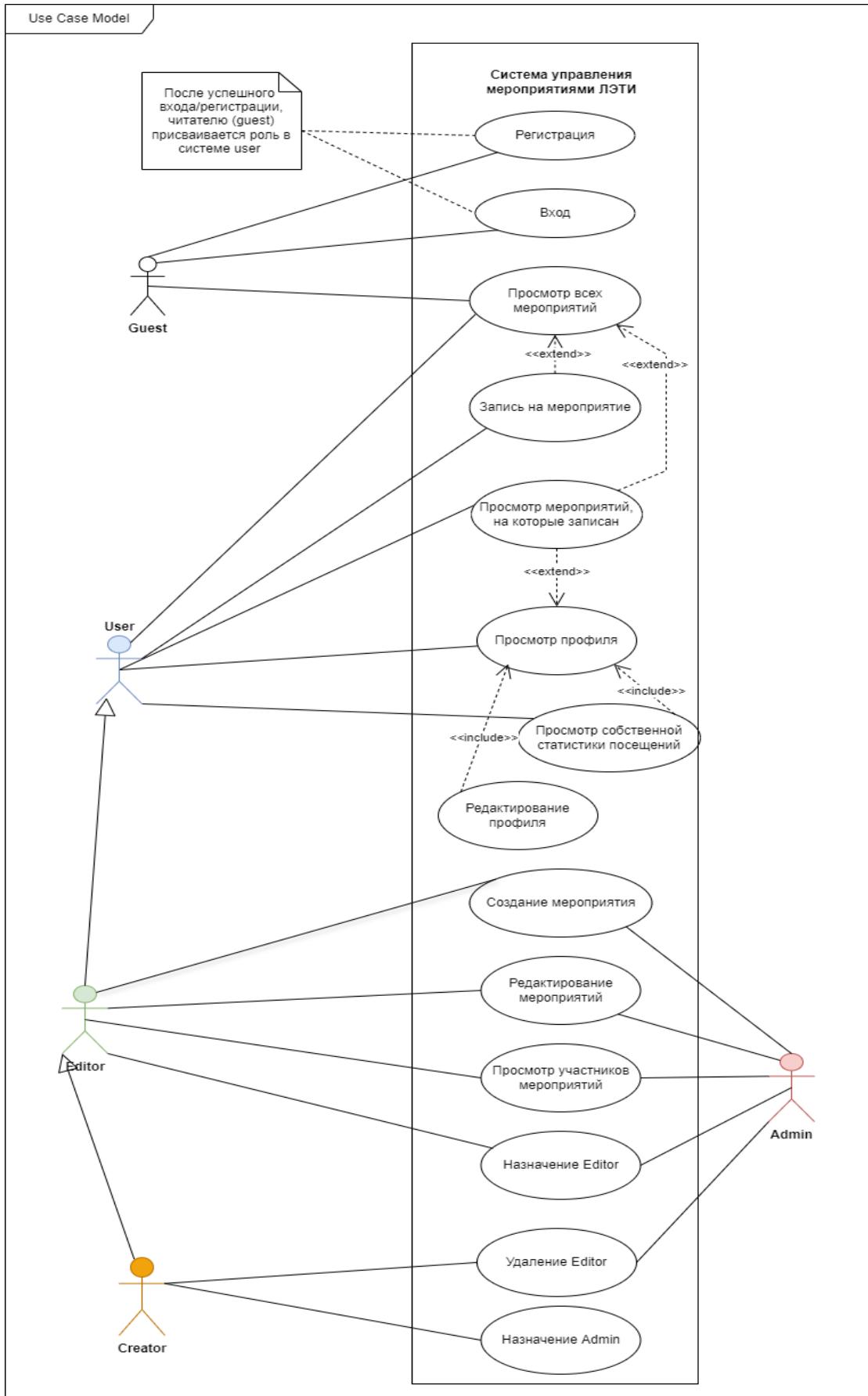


Диаграмма 2 - Диаграмма вариантов использования

## **Описание case-ов Use Case диаграммы для системы «Модуль “Мероприятия”» :**

Таблица 4.1 1 - Описание case-а «Регистрация посетителя»

Вариант использования	Регистрация посетителя
Актеры	Guest
Краткое описание	Первичная авторизация посетителя через ETU ID
Цель	Присвоить роль в системе и дать доступ к возможностям, предоставленным только авторизированным пользователям
Ссылки на другие варианты использования	Отсутствуют

Таблица 4.1 1

Таблица 4.1 2 - Ход выполнения case-а «Регистрация посетителя»

Действия актеров	Отклик системы
1. Гость нажимает на кнопку входа/регистрации или пытается совершить действие, не доступное не авторизованному пользователю	2. Система отображает страницу ввода логина и пароля ETU ID
3. Гость впервые вводит в окне авторизации свой логин и пароль ETU ID	4. Система сравнивает введенные пользователем данные с базой данных ETU ID Система выводит сообщение об успешной/не успешной авторизации пользователя В случае успешной авторизации, система создает учетную запись пользователя, присваивает роль User и отображает предыдущую страницу (с которой пользователь попал на страницу регистрации)

Таблица 4.1 2

Таблица 4.2 1 - Описание case-а «Вход посетителя»

Вариант использования	Вход посетителя
Актеры	Guest
Краткое описание	Авторизация посетителя через ETU ID
Цель	Присвоить роль в системе и дать доступ к возможностям, предоставленным только авторизированным пользователям
Ссылки на другие варианты использования	Отсутствуют

Таблица 4.2 1

Таблица 4.2 2 - Ход выполнения case-а «Вход посетителя»

Действия актеров	Отклик системы
1. Гость нажимает на кнопку входа/регистрации или пытается совершить действие, не доступное не авторизированному пользователю	2. Система отображает страницу ввода логина и пароля ETU ID
3. Гость вводит в окне авторизации свой логин и пароль ETU ID	4. Система сравнивает введенные пользователем данные с базой данных ETU ID Система выводит сообщение об успешной/не успешной авторизации пользователя В случае успешной авторизации, система загружает профиль пользователя и определяет его роль, затем отображает предыдущую страницу (с которой пользователь попал на страницу регистрации)

Таблица 4.2 2

Таблица 4.3 1 - Описание case-а «Просмотр всех мероприятий»

Вариант использования	Просмотр всех мероприятий
Актеры	Guest, User
Краткое описание	Посетитель/авторизированный пользователь просматривает список всех мероприятий вуза
Цель	Дать возможность посетителю/авторизированному пользователю ознакомиться со списком предстоящих мероприятий, проводимых различными подразделениями вуза
Ссылки на другие варианты использования	Extend: «Запись на мероприятие» расширяет возможности case-а «Просмотр всех мероприятий» Extend: «Просмотр мероприятий, на которые записан пользователь» расширяет возможности case-а «Просмотр всех мероприятий»

Таблица 4.3 1

Таблица 4.3 2 - Ход выполнения case-а «Просмотр всех мероприятий»

Действия актеров	Отклик системы
1. Гость/авторизированный пользователь просматривает список всех мероприятий вуза	2. Система отображает страницу со списком всех существующих мероприятий
3. Гость/авторизированный пользователь нажимает на какое-либо мероприятие	4. Система, в случае необходимости, выводит полную информацию по конкретному мероприятию

Таблица 4.3 2

Таблица 4.4 1 - Описание case-а «Запись на мероприятие»

Вариант использования	Запись на мероприятие
Актеры	User, Editor, Creator
Краткое описание	Авторизованный пользователь записывается на мероприятие
Цель	Дать возможность авторизированному пользователю записаться на любое из мероприятий, проводимых различными подразделениями вуза
Ссылки на другие варианты использования	Extend: «Запись на мероприятие» расширяет возможности case-а «Просмотр всех мероприятий»

Таблица 4.4 1

Таблица 4.4 2 - Ход выполнения case-а «Запись на мероприятие»

Действия актеров	Отклик системы
1. Авторизованный пользователь нажимает кнопку "записаться"	2. Система отображает блок для ввода личных данных с возможностью автозаполнения по данным из профиля
3. Авторизованный пользователь нажимает кнопку подтверждения	4. Система создает связь "Участие", привязывая пользователя к выбранному мероприятию Затем система выводит сообщение о успешной/не успешной регистрации и отображает предыдущую страницу

Таблица 4.4 2

Таблица 4.5 1 - Описание case-а «Просмотр мероприятий, на которые записан пользователь»

Вариант использования	Просмотр мероприятий, на которые записан пользователь
Актеры	User, Editor, Creator
Краткое описание	Просмотр списка мероприятий, на которые записан пользователь
Цель	Дать возможность авторизированному пользователю ознакомиться со списком мероприятий на которые он записан, а так же удалить запись
Ссылки на другие варианты использования	Extend: «Просмотр мероприятий, на которые записан пользователь» расширяет возможности case-а «Просмотр профиля» Extend: «Просмотр мероприятий, на которые записан пользователь» расширяет возможности case-а «Просмотр всех мероприятий»

Таблица 4.5 1

Таблица 4.5 2 - Ход выполнения case-а «Просмотр мероприятий, на которые записан пользователь»

Действия актеров	Отклик системы
1. Авторизованный пользователь пролистывает ниже в своем профиле или нажимает на соответствующую кнопку в разделе всех мероприятий	2. Система отображает на странице профиля список мероприятий, с которыми пользователь связан Участием
3. Пользователь просматривает список всех мероприятий, на которые записан	
4. Пользователь нажимает на интересующее его мероприятие	5. Система отображает полную информацию о мероприятии, а так же предоставляет возможность удалить запись на мероприятие
6. Если пользователь нажимает на "удалить запись"	7. Система удаляет связь Участие между пользователем и мероприятием и престает отображать данное мероприятие в профиле

Таблица 4.5 2

Таблица 4.6 1 - Описание case-а «Просмотр профиля»

Вариант использования	Просмотр профиля
Актеры	User, Editor, Creator
Краткое описание	Авторизованный пользователь переходит в раздел профиля с индивидуальной информацией
Цель	Дать возможность авторизованному пользователю ознакомиться с данными своего профиля с возможностью изменить их) и увидеть свою статистику
Ссылки на другие варианты использования	Extend: «Просмотр мероприятий, на которые записан пользователь» расширяет возможности case-а «Просмотр профиля» Include: «Просмотр собственной статистики посещений» включает в себя «Просмотр профиля» Include: «Редактирование профиля» включает в себя «Просмотр профиля»

Таблица 4.6 1

Таблица 4.6 2 - Ход выполнения case-а «Просмотр профиля»

Действия актеров	Отклик системы
1. Авторизованный пользователь переходит в раздел профиля	2. Система отображает страницу личного профиля пользователя

Таблица 4.6 2

Таблица 4.7 1 - Описание case-а «Просмотр собственной статистики посещений»

Вариант использования	Просмотр собственной статистики посещений
Актеры	User, Editor, Creator
Краткое описание	Авторизованный пользователь просматривает свою статистику
Цель	Дать возможность авторизированному пользователю просмотреть свою статистику посещений, увидеть мероприятия, которые он уже посещал ранее
Ссылки на другие варианты использования	Include: «Просмотр собственной статистики посещений» включает в себя «Просмотр профиля»

Таблица 4.7 1

Таблица 4.7 2 - Ход выполнения case-а «Просмотр собственной статистики посещений»

Действия актеров	Отклик системы
1. Авторизованный пользователь просматривает раздел статистики, пролистав вниз в профиле	2. Система отображает блок статистики пользователя
3. Пользователь нажимает кнопку развертывания архива мероприятий, которые он посещал ранее	4. Система выводит список мероприятий, со статусом "В архиве", с которыми был связан пользователь

Таблица 4.7 2

Таблица 4.8 1 - Описание case-а «Редактирование профиля»

Вариант использования	Редактирование профиля
Актеры	User, Editor, Creator
Краткое описание	Авторизованный пользователь изменяет личную информацию в своем профиле
Цель	Дать возможность авторизированному пользователю внести изменения в информацию о себе и контактные данные в своем профиле
Ссылки на другие варианты использования	Include: «Редактирование профиля» включает в себя «Просмотр профиля»

Таблица 4.8 1

Таблица 4.8 2 – «Редактирование профиля»

Действия актеров	Отклик системы
1. Авторизованный пользователь нажимает кнопку "Редактировать профиль"	2. Система дает возможность пользователю изменять свои данные (за исключением ФИО и номера группы), а так же позволяет отключить функцию отображения статистики
3. Пользователь нажимает кнопку "Сохранить изменения"	4. Система выполняет "проверку на дурака", а так же на соответствие цензуре. Система обновляет данные объекта Пользователь и впоследствии отображает в профиле обновленную информацию

Таблица 4.8 2

Таблица 4.9 1 - Описание case-а «Создание мероприятия»

Вариант использования	Создание мероприятия
Актеры	Editor, Admin, Creator
Краткое описание	Авторизованный пользователь с правами на управление мероприятиями создает новое мероприятие от лица своего подразделения
Цель	Дать возможность авторизированному пользователю с правами на управление мероприятиями создать новое мероприятие от лица своего подразделения (добавить: название, дату проведения, изображение, описание)
Ссылки на другие варианты использования	Отсутствует

Таблица 4.9 1

Таблица 4.9 2 - Ход выполнения case-а «Создание мероприятия»

Действия актеров	Отклик системы
1. Авторизованный пользователь с правами на управление мероприятиями переходит в раздел создания нового мероприятия	2. Система отображает меню, в котором редактор может выбрать действие (создание/редактирование/просмотр участников)
3. Редактор переходит в раздел создания нового мероприятия	4. Система отображает страницу, на которой можно загрузить информацию о новом мероприятии
5. Редактор вводит название мероприятия	6. Система создает объект Мероприятие. Система проверяет введенное название на соответствие типу данных String и добавляет в Мероприятие
7. Редактор вводит дату и время проведения мероприятия	8. Система проверяет на соответствие типу данных DateTime и добавляет в Мероприятие
9. Редактор вводит описание мероприятия	10. Система проверяет на соответствие типу данных String и добавляет в Мероприятие
11. Редактор загружает изображение для карточки мероприятия	12. Система проверяет на соответствие формату и максимально разрешенному размеру файла и добавляет в Мероприятие
13. Редактор проверяет введенные данные и нажимает кнопку "создать"	14. Система присваивает мероприятию индивидуальный id, а так же тег подразделения, от которого оно создается. Система добавляет в общую базу мероприятий и начинает отображать его в разделе просмотра всех мероприятий. Система выводит сообщение о успешной/не успешной публикации и отображает предыдущую страницу

Таблица 4.9 2

Таблица 4.10 1 - Описание case-а «Редактирование мероприятий»

Вариант использования	Редактирование мероприятий
Актеры	Editor, Admin, Creator
Краткое описание	Авторизованный пользователь с правами на управление мероприятиями редактирует существующие мероприятия своего подразделения
Цель	Дать возможность авторизированному пользователю с правами на управление мероприятиями изменить любую составляющую существующего объекта "мероприятие" от своего подразделения (изменить: название, дату проведения, изображение, описание)
Ссылки на другие варианты использования	Отсутствует

Таблица 4.10 1

Таблица 4.10 2 - Ход выполнения case-а «Редактирование мероприятий»

Действия актеров	Отклик системы
1. Авторизованный пользователь с правами на редактирование мероприятий переходит в раздел управления мероприятиями	2. Система отображает меню, в котором редактор может выбрать действие (создание/редактирование/просмотр участников)
3. Редактор переходит в раздел редактирования мероприятий	4. Система отображает страницу, на которой перечислены все мероприятия подразделения, к которому относится редактор (в случае Creator - полный список мероприятий всех подразделений)
5. Редактор выбирает мероприятие из списка или фильтруя по названию/дате	6. Система отображает страницу, на которой можно внести изменения в каждую из составляющих выбранного объекта "мероприятие"
7. Редактор изменяет название мероприятия	8. Система проверяет введенное название на соответствие типу данных String и добавляет в Мероприятие Система обновляет объект Мероприятие.
9. Редактор изменяет дату и время проведения мероприятия	10. Система проверяет на соответствие типу данных DateTime Система обновляет объект Мероприятие
11. Редактор изменяет описание мероприятия	12. Система проверяет на соответствие типу данных String Система обновляет объект Мероприятие
13. Редактор изменяет изображение для карточки мероприятия	14. Система проверяет на соответствие формату и максимально разрешенному размеру файла Система обновляет объект Мероприятие

15. Редактор проверяет введенные данные и нажимает кнопку "создать"	16. Система полностью обновляет объект Мероприятие, сохраняя его прежние id и тег. Система выводит сообщение о успешном/не успешном обновлении и отображает предыдущую страницу
17. Если редактор решает отменить мероприятие, он нажимает на кнопку "удалить" в разделе редактирования мероприятия	18. Система удаляет объект мероприятие, удаляя все его данные из базы данных При удалении объекта система высылает всем пользователям, зарегистрированным на данное мероприятие (имеющим связь Участие), уведомление/сообщение: "Мероприятие <название> - отменено."

Таблица 4.10 2

Таблица 4.11 1 - Описание case-а «Просмотр участников мероприятий»

Вариант использования	Просмотр участников мероприятий
Актеры	Editor, Admin, Creator
Краткое описание	Авторизированный пользователь с правами на управление мероприятия просматривает список участников существующего мероприятия своего подразделения
Цель	Дать возможность авторизированному пользователю с правами на управление мероприятиями ознакомиться со списком пользователей, зарегистрированных на участие в активном мероприятии от своего подразделения
Ссылки на другие варианты использования	Отсутствует

Таблица 4.11 1

Таблица 4.11 2 - Ход выполнения case-а «Просмотр участников мероприятий»

Действия актеров	Отклик системы
1. Авторизованный пользователь с правами на редактирование мероприятий переходит в раздел управления мероприятиями	2. Система отображает меню, в котором редактор может выбрать действие (создание/редактирование/просмотр участников)
3. Редактор переходит в раздел просмотра участников мероприятий	4. Система отображает страницу, на которой перечислены все мероприятия подразделения, к которому относится редактор (в случае Creator - полный список мероприятий всех подразделений)
5. Редактор выбирает мероприятие из списка или фильтруя по названию/дате	6. Система отображает страницу, на которой показана информация объекта Мероприятие Система отображает список всех пользователей, зарегистрированных на мероприятие (имеющие связь Участие) Система предлагает выбрать какую информацию в списке отображать (ФИО/ФИО+номер группы/телефон/почта)
7. Редактор выбирает способ отображения списка (ФИО/ФИО+номер группы/телефон/почта)	8. Система отображает список участников в соответствии с запросом редактора Система предоставляет возможность скачать список в формате txt

Таблица 4.11 2

Таблица 4.12 1 - Описание case-а «Назначение Editor»

Вариант использования	Назначение Editor
Актеры	Editor, Admin, Creator
Краткое описание	Авторизованный пользователь с правами на управление мероприятиями назначает нового Editor для своего подразделения, воспользовавшись поиском по ФИО среди всех авторизованных пользователей
Цель	Дать возможность авторизованному пользователю с правами на управление мероприятиями назначить нового Editor, и предоставить ему права на управление мероприятиями своего подразделения, воспользовавшись поиском по ФИО среди всех авторизованных пользователей
Ссылки на другие варианты использования	Отсутствует

Таблица 4.12 1

Таблица 4.12 2 - Ход выполнения case-а «Назначение Editor»

<b>Действия актеров</b>	<b>Отклик системы</b>
1. Авторизованный пользователь с правами на управление мероприятиями переходит в раздел "назначить Editor"	2. Система отображает строку поиска по ФИО и готова выполнять поиск среди всех авторизованных пользователей данной системы
3. Редактор вводит ФИО пользователя	4. Если пользователь найден - система выводит данные пользователя и кнопку "назначить" Если пользователь не найден - система выводит соответствующее сообщение
5. Редактор нажимает "назначить"	6. Система определяет подразделение редактора, который назначает нового Editor, и прикрепляет нового Editor к соответствующему подразделению (исключение - Creator, он имеет возможность выбрать, к какому подразделению прикрепить нового Editor) Система назначает выбранному пользователю роль в системе Editor и открывает все соответствующие права.

Таблица 4.12 2

Таблица 4.13 1 - Описание case-а «Удаление Editor»

<b>Вариант использования</b>	<b>Удаление Editor</b>
Актеры	Admin, Creator
Краткое описание	Администратор или разработчик разжалует Editor своего подразделения до роли User, лишая всех соответствующих прав доступа, воспользовавшись поиском по ФИО среди всех авторизованных пользователей
Цель	Дать возможность администратору или разработчику разжаловать Editor какого-либо подразделения до роли User, лишив всех соответствующих прав доступа, воспользовавшись поиском по ФИО среди всех авторизованных пользователей
Ссылки на другие варианты использования	Отсутствует

Таблица 4.13 1

Таблица 4.13 2 - Ход выполнения case-а «Удаление Editor»

<b>Действия актеров</b>	<b>Отклик системы</b>
1. Администратор или разработчик переходит в раздел "удаление Editor"	2. Система отображает строку поиска по ФИО и готова выполнять поиск среди всех пользователей с ролью Editor
3. Редактор вводит ФИО пользователя	4. Если пользователь найден и прав хватает - система выводит данные пользователя и кнопку "разжаловать" Если пользователь не найден или не хватает прав доступа (в случае, если действие выполняет Admin, а Editor прикреплен к подразделению, отличному от подразделения Admin) - система выводит соответствующее сообщение
5. Редактор нажимает кнопку "разжаловать"	6. Система обновляет роль в системе для пользователя и он из Editor становится User, лишаясь при этом всех соответствующих прав редактора

Таблица 4.13 2

Таблица 4.14 1 - Описание case-а «Назначение Admin»

<b>Вариант использования</b>	<b>Назначение Admin</b>
Актеры	Creator
Краткое описание	Creator (разработчик) назначает нового Admin для своего подразделения, воспользовавшись поиском по ФИО среди всех авторизированных пользователей
Цель	Дать возможность Creator назначить нового Admin, и предоставить ему права на управление одним из подразделений, воспользовавшись поиском по ФИО среди всех авторизированных пользователей
Ссылки на другие варианты использования	Отсутствует

Таблица 4.14 1

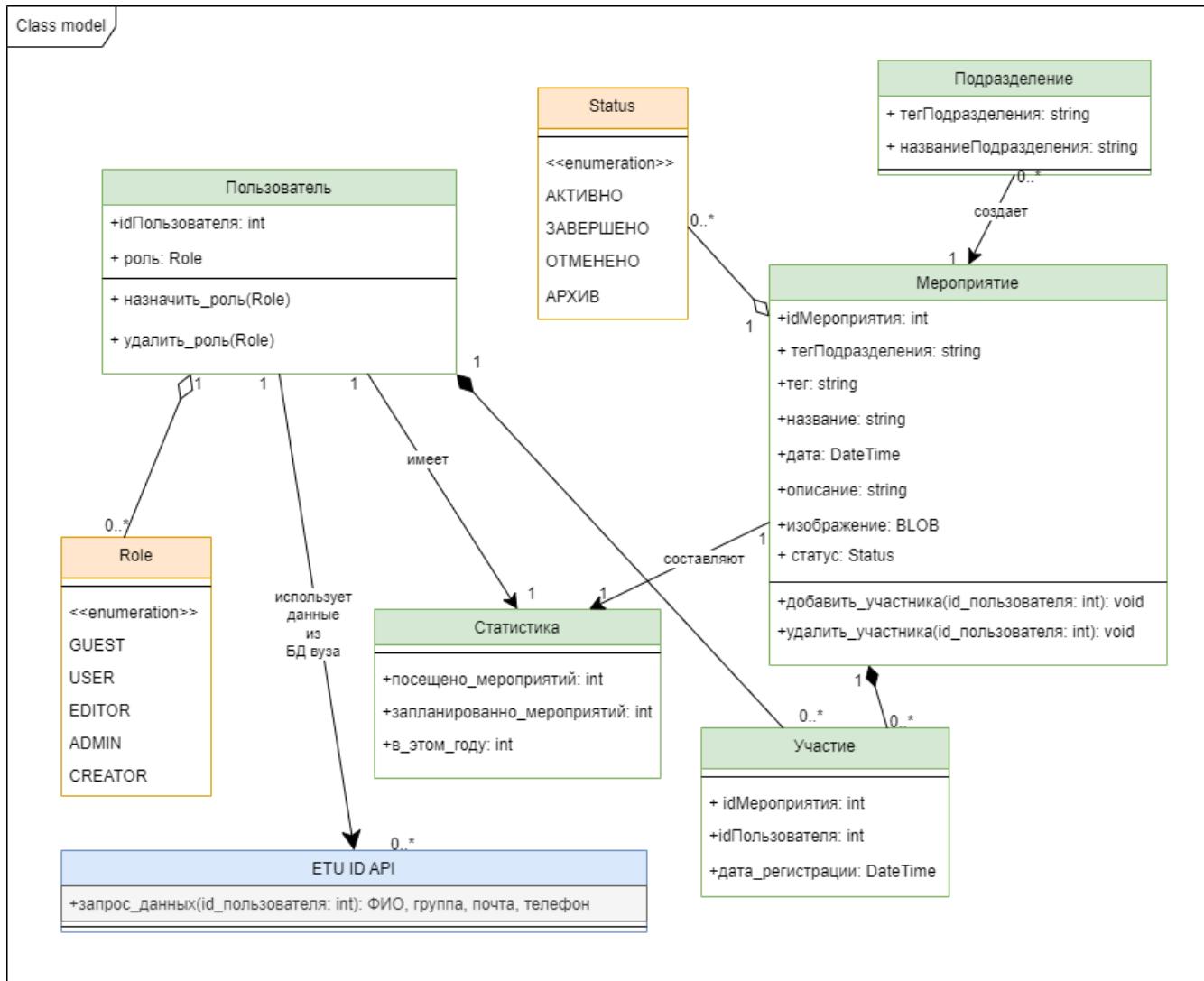
Таблица 4.14 2 - Ход выполнения case-а «Назначение Admin»

Действия актеров	Отклик системы
1. Creator (разработчик) переходит в раздел "назначить Admin"	2. Система отображает строку поиска по ФИО и готова выполнять поиск среди всех авторизированных пользователей данной системы
3. Creator вводит ФИО пользователя	4. Если пользователь найден - система выводит данные пользователя, список подразделений, к которым можно прикрепить Admin и кнопку "назначить" Если пользователь не найден - система выводит соответствующее сообщение
5. Creator выбирает нужное подразделение из списка и нажимает "назначить"	6. Система назначает выбранному пользователю роль в системе Admin и открывает все соответствующие права. Система приписывает нового Admin к соответствующему подразделению

Таблица 4.14 2

### 4.3 Диаграмма классов (Class Diagram)

Данная диаграмма содержит 6 классов: Пользователь, Мероприятие, Подразделение, Статистика, участия , и 2 класса-перечисления набора констант: Status и Role.



## Описание классов.

### 1. Класс Пользователь: атрибуты приведены в табл. 1.

Атрибут	Тип	Видимость
+idПользователя	int	Общедоступный
+роль	Role	Общедоступный

Таблица классы 1

Данный класс содержит два метода:

+ назначить\_роль(Role): void

Общедоступный метод, который присваивает пользователю новую роль из перечисления Role. Возвращает значение void.

+ удалить\_роль(Role): void

Общедоступный метод, который удаляет текущую роль пользователя  
Ничего не возвращает.

### 2. Класс Мероприятие: атрибуты приведены в табл. 2.

Атрибут	Тип	Видимость
+idМероприятия	int	Общедоступный
+тегПодразделения	string	Общедоступный
+название	string	Общедоступный
+дата	DateTime	Общедоступный
+описание	string	Общедоступный
+изображение	BLOB	Общедоступный
+статус	Status	Общедоступный

Таблица классы 2

Данный класс содержит два метода:

+ добавить\_участника(id\_пользователя: int): void

Общедоступный метод. Добавляет участника в мероприятие по его идентификатору.  
Ничего не возвращает.

+ удалить\_участника(id\_пользователя: int): void

Общедоступный метод. Удаляет участника из мероприятия по его идентификатору.

Ничего не возвращает.

### 3. Класс Подразделение: атрибуты приведены в табл. 3.

Атрибут	Тип	Видимость
+тегПодразделения	string	Общедоступный
+названиеПодразделения	string	Общедоступный

Таблица классы 3

Данный класс не содержит методов.

**4. Класс Участие: атрибуты приведены в табл. 4.**

Атрибут	Тип	Видимость
+idМероприятия	int	Общедоступный
+idПользователя	int	Общедоступный
+дата_регистрации	DateTime	Общедоступный

Таблица классы 4

+ создать\_участие(): void

Создает запись об участии. Ничего не возвращает.

**5. Класс Статистика: атрибуты приведены в табл. 5.**

Атрибут	Тип	Видимость
+посещено_мероприятий	int	Общедоступный
+активность	double	Общедоступный
+рейтинг	int	Общедоступный

Таблица классы 5

Данный класс не содержит методов.

**6. Класс ETU ID API: Данный класс не содержит атрибутов.**

+ запрос\_данных(id\_пользователя): ФИО, группа, почта, телефон  
 int (ID пользователя) Возвращает значения: string, int, string, string  
 Запрашивает данные пользователя из внешней системы

**7. Классы-перечисления Status и Role.**

Status:
АКТИВНО
ЗАВЕРШЕНО
ОТМЕНЕНО
АРХИВ

Role:
GUEST
USER
EDITOR
ADMIN
CREATOR

## Связи между классами – таблица классы 6.

Классы	Тип связи	Описание
Подразделение и Мероприятие	Зависимость	Подразделение может организовывать мероприятия.
Мероприятие и Участие	Композиция	Мероприятие содержит записи об участии.
Пользователь и Участие	Композиция	Пользователь может участвовать в мероприятиях.
Пользователь и Статистика	Ассоциация	У каждого пользователя есть своя статистика.
Пользователь и Role	Агрегация	Пользователь зависит от перечисления Role.
Мероприятие и Status	Агрегация	Мероприятие зависит от перечисления Status.
ETU ID API и Пользователь	Ассоциация	Система запрашивает данные пользователей через API из базы данных вуза.
Статистика и Мероприятие	Зависимость	Статистика может учитывать данные о мероприятиях.

Таблица классы 6

#### **4.3 Диаграмма деятельности (Activity diagram)**

Диаграмма деятельности (активности) отображает динамические аспекты поведения системы. Эта диаграмма представляет собой блоксхему, которая наглядно показывает, как поток управления переходит от одной деятельности к другой, от одного актора к другому.

На приведенной ниже диаграмме деятельности отображено взаимодействие с системой всех актеров. Показаны процессы:

1. Просмотра мероприятий и авторизации гостя.
2. Варианты взаимодействий с системой для пользователя (просмотр мероприятий, запись на мероприятия, просмотр профиля, статистики и предстоящих мероприятий, на которые зарегистрирован пользователь, а также возможность редактирования профиля)
3. Создания и редактирования мероприятий редакторами, а также просмотра списка участников и назначения новых ролей.

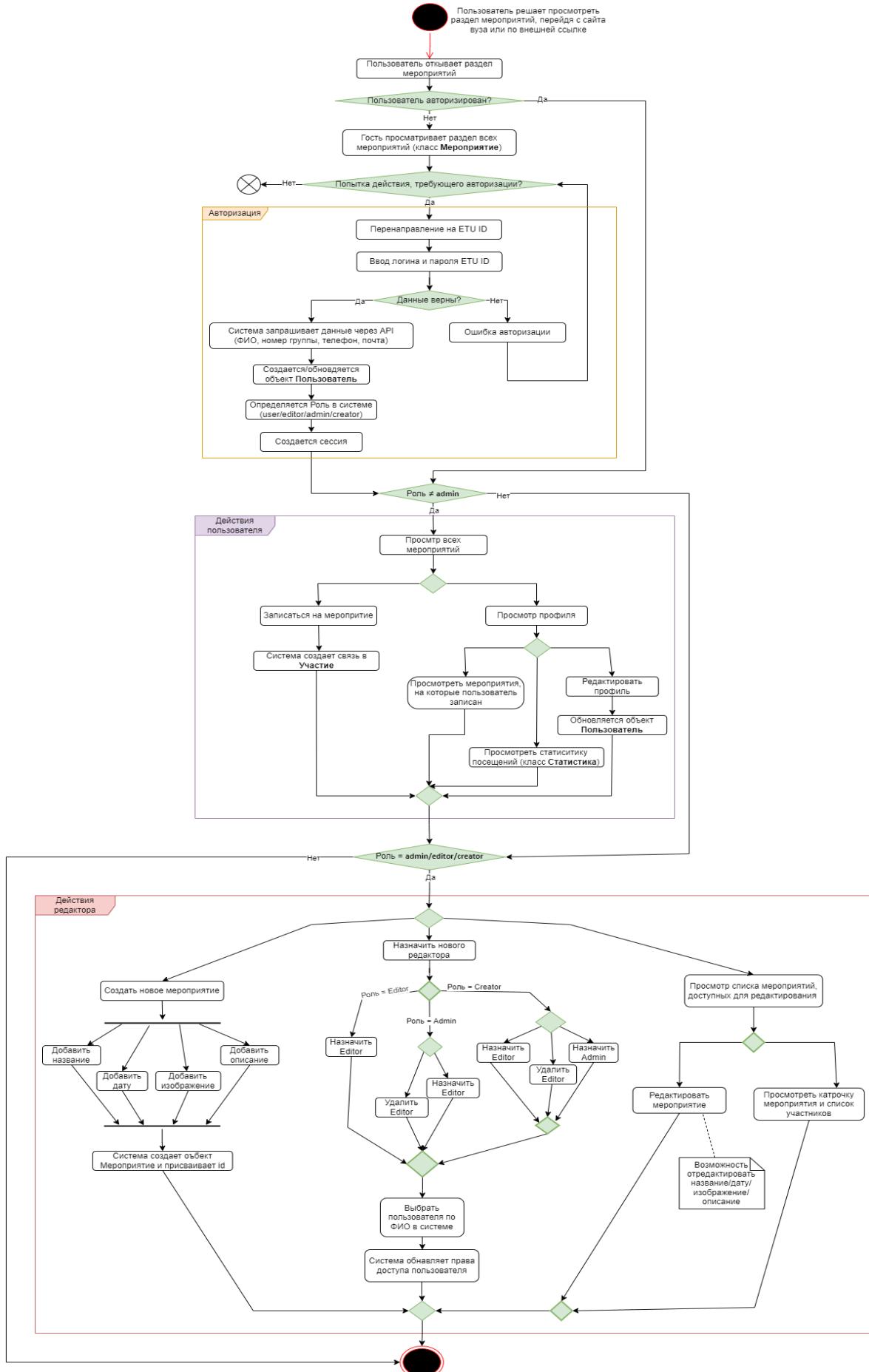


Диаграмма 3 - Диаграмма активности

#### **4.4 Диаграмма конечного автомата (State diagram)**

Диаграмма конечного автомата (State Machine Diagram) является графом, который представляет некоторый конечный автомат.

- Вершина – состояние (или псевдосостояние).
- Дуга – переход между состояниями.

На приведённой ниже диаграмме конечного автомата показан жизненный цикл объекта «Мероприятие». Оно создается, когда редактор начинает вводить данные и образуется черновик, затем мероприятие опубликовывается и получает статус Активно, в этот момент открывается регистрация участников. После чего мероприятие может быть либо отменено (в этом случае цикл закончится), либо состоится и затем перейдет в статус архивированного. В этом случае оно сохранится в системе и добавится в статистику участников.

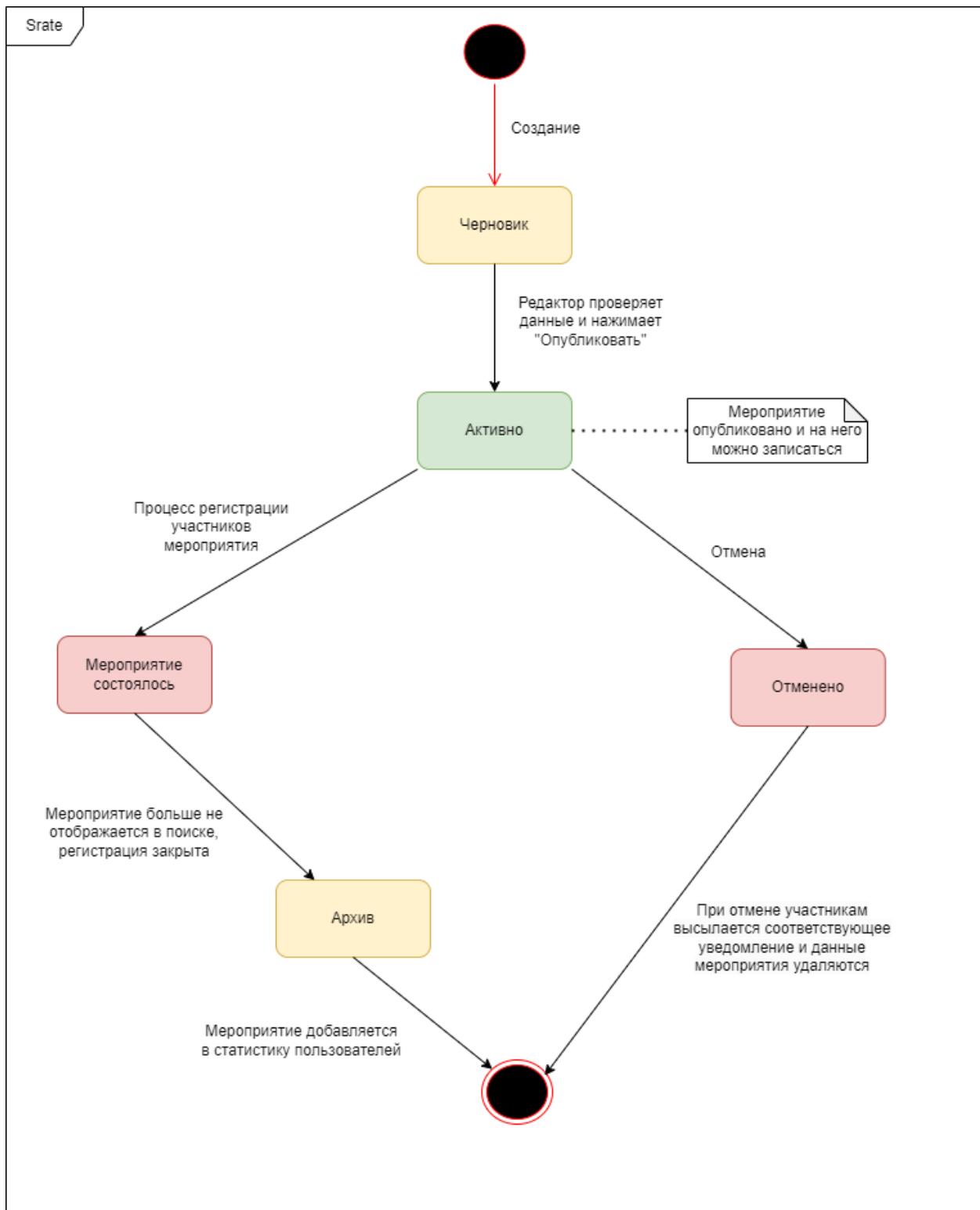


Диаграмма 4 - Диаграмма конечного автомата

## 5. Тесты

Функциональность модуля «Мероприятия» для ЛК ЛЭТИ должна быть протестирована в нескольких аспектах:

a. Юнит-тесты (в соответствии с диаграммой классов):

Проверка корректности создания и управления объектами классов:

- Пользователь: тестирование методов назначения и удаления ролей.
- Мероприятие: тестирование добавления и удаления участников, изменения статуса мероприятия.
- Участие: проверка создания и удаления записей об участии.
- Статистика: тестирование обновления данных о посещенных мероприятиях.

b. Функциональные тесты (в соответствии с use cases и заявленными функциями приложения):

*Для гостей (Guest):*

- Проверка возможности просмотра списка мероприятий без авторизации.
- Проверка корректности работы фильтров (по дате, направлению).

*Для пользователей (User):*

- Проверка регистрации на мероприятие и отмены записи.
- Проверка отображения списка запланированных и посещенных мероприятий в личном кабинете.
- Проверка редактирования профиля (изменение контактных данных, аватарки).

*Для организаторов (Editor):*

- Проверка создания и редактирования мероприятий.
- Проверка просмотра списка участников и скачивания данных.
- Проверка назначения новых редакторов.

*Для руководителей отделов (Admin):*

- Проверка удаления редакторов.
- Проверка управления мероприятиями подразделения.

*Для разработчиков (Creator):*

- Проверка назначения и удаления руководителей отделов.
- Проверка доступа ко всем мероприятиям и статистике.

с. Интеграционные тесты:

- Проверка взаимодействия с ETU ID (авторизация, получение данных пользователя).
- Проверка интеграции с Google Calendar API (синхронизация мероприятий).
- Проверка работы Telegram-бота (отправка уведомлений).

4. Нагрузочные тесты:

- Проверка времени отклика системы при одновременном доступе 1000 пользователей.
- Проверка стабильности работы при создании и редактировании мероприятий в условиях высокой нагрузки.
- Проверка обработки большого количества запросов к API (например, при массовой регистрации на мероприятие).

5. User acceptance тесты (UAT):

Проверка удобства интерфейса:

- Время на регистрацию на мероприятие не должно превышать 5 минут.
- Интерфейс должен быть интуитивно понятным для всех ролей (гость, пользователь, организатор, руководитель, разработчик).

Проверка соответствия сценариям использования:

- Система должна корректно отражать поведение, описанное в use cases (например, успешная регистрация, отмена записи, создание мероприятия).

Проверка информационных сообщений:

- Сообщения об ошибках, успешных действиях и уведомления должны быть четкими и понятными для пользователя.

6. Тесты безопасности:

- Проверка защиты от SQL-инъекций и других уязвимостей.
- Проверка шифрования передаваемых данных (HTTPS).
- Проверка корректности разграничения прав доступа (например, гость не может редактировать мероприятия, а пользователь не может назначать редакторов).

7. Тесты производительности:

- Проверка времени загрузки списка мероприятий (не более 2 секунд).
- Проверка времени отклика при редактировании профиля или создании мероприятия.

8. Тесты отказоустойчивости:

- Проверка работы системы при внезапном отключении интернета.
- Проверка восстановления данных после сбоя (например, при внезапном закрытии браузера).

9. Тесты мобильной адаптации:

- Проверка корректного отображения интерфейса на устройствах с разными разрешениями экрана (смартфоны, планшеты).
- Проверка удобства использования интерфейса на мобильных устройствах (например, кнопки должны быть достаточно крупными для нажатия).

10. Тесты API:

- Проверка корректности работы всех endpoints (GET, POST, PUT, DELETE).
- Проверка обработки некорректных запросов (например, неверные параметры, отсутствие авторизации).

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения лабораторной работы был спроектирован модуль "Мероприятия" для личного кабинета ЛЭТИ. Были сформированы технические требования, удовлетворяющие модели FURS, а также составлено архитектурное описание и тесты для проекта. При разработке архитектуры модуля для удобства было обозначено несколько видов пользователей с разными уровнями доступа: гость, обычный пользователь, организатор мероприятий, руководитель отдела и разработчик.

Административная часть системы содержит инструменты управления мероприятиями и пользовательскими ролями. В пользовательской части реализован удобный интерфейс для просмотра мероприятий, регистрации на них и управления личным профилем. Особое внимание было уделено интеграции с ETU ID для обеспечения безопасной аутентификации пользователей.

Разработанный модуль представляет собой законченное решение, готовое к внедрению в образовательный процесс ЛЭТИ. Он соответствует всем поставленным требованиям и обеспечивает удобный инструмент для организации мероприятий и учета участия студентов.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

4. Аナンьева В.Я. UML-диаграммы. Основные возможности. – СПб: Издательство СПбГЭТУ «ЛЭТИ», 2023 — 30 с.
5. Леоненков А. В. Самоучитель UML 2. – СПб: БХВ-Петербург, 2007 — 576 с.
6. Водяхо А.И. Выговский Л.С., Дубенецкий В.А.Цехановский В.В. Архитектурные решения информационных систем. — СПб: Издательство «Лань», 2022 — 356 с.
7. Bryan Basham, Kathy Sierra, Bert Bates. «Head First Servlets and JSP: Passing the Sun Certified Web Component Developer Exam» — O'Reilly Media, 2008. — 883 с.