Введение

В настоящее время активность студентов в учебном процессе не ограничивается только посещением лекций и выполнением заданий. Многие студенты стремятся проявлять себя в других областях, участвуя в различных мероприятиях, таких как соревнования, конференции, круглые столы и т.д. Проведение таких мероприятий, как правило, организуется силами организаторов из штата ИРНИТУ.

Мероприятия представляют различные направления студенческой деятельности: спортивные, культурные, образовательные, научные и общественные. Спортивные мероприятия могут быть организованы в форме соревнований по футболу, баскетболу, волейболу, теннису или другим видам спорта. Культурно-творческие мероприятия могут включать в себя концерты, выставки, театральные постановки и многие другие активности. Образовательные мероприятия, в свою очередь, могут быть организованы в форме лекций, семинаров, мастер-классов. Студенты могут участвовать в данных мероприятиях не только в качестве участников, но и в качестве зрителей или организаторов. Благодаря этому они могут получить ценный опыт и навыки, которые пригодятся им в будущем. Организаторы мероприятий могут приобрести навыки планирования и координации, а также узнать, как работать в команде.

В рамках университета действует рейтинговая стипендия, призванная поощрять и поддерживать студентов, активно участвующих в различных мероприятиях. Стипендия представляет собой возможность для студентов получить дополнительное финансовое поощрение за их активную жизненную позицию.

Участие в мероприятиях поощряется с помощью получения рейтинговых баллов, которые суммируются в их рейтинге. Количество баллов увеличивается, если студент занимает призовые места. Зрители являются просто наблюдателями. При этом в университете официально не ведется учет посещаемости мероприятий для формирования вовлеченности студентов. Организаторы отвечают за планирование, координацию и проведение мероприятий. За свою работу по организации, они также получают баллы в рейтинг.

Для участия в рейтинговой стипендии студенты должны подтвердить участие в мероприятии или его организацию. Для этого необходимо получить справку от сотрудника учебного заведения, причастного к проведению события, что может занимать продолжительное время.

Для упрощения процессов организации мероприятий, подсчета рейтинговой стипендии и отслеживания уровня вовлеченности студентов в ИРНИТУ разрабатывается информационная система «Рейтинг студента», затрагивающая перечисленные раннее аспекты. В рамках данной системы необходимо разработать модуль, отвечающий за регистрацию студентов на мероприятия.

Данная работа направлена на создание единой системы регистрации с возможностью выбора роли и подтверждения фактического присутствия.

Актуальность дипломной работы связана с растущей потребностью отслеживания участия студентов в мероприятиях. Разработка приложения позволит вести учет участия студентов в мероприятиях, производить градацию студентов в зависимости от уровня их вовлеченности.

Определение роли, уровня участия и взаимодействия студентов позволит достичь эффективного взаимодействия студентов и системы «Рейтинг студента».

Целью выпускной квалификационной работы является оптимизация процесса регистрации на мероприятия и отслеживания присутствия студентов, что позволит сделать данный процесс более эффективным и удобным для всех участников.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Изучить процесс учета участия студентов в мероприятиях. В рамках данной задачи необходимо изучить, каким образом участие студентов учитывается в рейтинговой стипендии, как происходит начисление баллов за участие, а также каким образом подтверждается участие студентов в мероприятиях;
2. Выполнить анализ предметной области. Эта задача включает в себя анализ существующих систем регистрации на мероприятия, а также знакомство с современными подходами в области разработки приложений;
3. Сформировать требования к системе. На основе анализа предметной области и описанного процесса учета участия студентов в мероприятиях в рейтинговой стипендии необходимо сформировать требования к разрабатываемой системе. В частности, следует определить функциональные и нефункциональные требования к приложению, графическому интерфейсу, безопасности данных;
4. Выполнить проектирование приложения. На данном этапе необходимо разработать архитектуру приложения, спроектировать пользовательский интерфейс, определить используемые технологии и инструменты;
5. Реализовать прототип приложения. На данном этапе необходимо реализовать прототип приложения на основе результатов проектирования;
6. Тестирование приложения. Необходимо провести тестирование системы для достижения корректной работы приложения. Тестирование позволит выявить и устранить ошибки и доработки, которые могут возникнуть в процессе использования приложения пользователями.

Успешная реализация проекта позволит существенно упростить процесс регистрации на мероприятия для студентов ИРНИТУ и улучшить учет уровня вовлеченности студентов в мероприятиях.

1 Анализ предметной области

1.1 Система начисления рейтинговой стипендии

На данный момент рейтинговая гонка в ИРНИТУ проводится два раза в год и направлена на поддержку студенческой активности в пяти направлениях: учебной, научной, спортивной, культурно-творческой и общественной деятельности. Студенты, проявившие высокую активность в каждом из направлений, имеют возможность получить повышенную рейтинговую стипендию [2].

Однако, текущая система сбора и обработки заявок на рейтинговую стипендию является очень громоздкой и требует много времени и усилий. Необходимо собирать различные бумаги и сертификаты о проведении мероприятий, участии в коллективах и других деятельностях, после чего ответственные за рейтинг проводят проверку и формируют списки студентов, получающих рейтинговую стипендию.

Проект «Рейтинг студента» позволит значительно упростить этот процесс и сделать его более удобным для всех участников. Создание единой системы для регистрации коллективов, проведения мероприятий, формирования заявок и начисления рейтинговой стипендии позволит автоматизировать многие процессы и уменьшить количество работы с бумагами. На рисунке 1.1 представлены модули системы «Рейтинг студента».

Итоговая информационная система будет охватывать многие аспекты студенческой жизни такие как: учет студенческих коллективов в рамках университета, облегчение подготовки и проведения мероприятий, формирование заявок студентов для участия в рейтинговой стипендии, проверку поданных заявок и начисление по результатам повышенной государственной рейтинговой стипендии.



Рисунок 1.1 – Система «Рейтинг студента»

Таким образом, проект «Рейтинг студента» будет способствовать развитию студенческой активности в ИРНИТУ и созданию условий для проявления талантов и способностей студентов. Это поможет формировать лучших специалистов в различных областях и повышать репутацию университета в глазах работодателей и общественности.

Одной из ключевых особенностей проекта «Рейтинг студента» является автоматизация процесса подачи заявок на рейтинговую стипендию. Вместо того чтобы тратить время на сбор и проверку бумажных заявлений, студенты могут подавать заявки онлайн через специальную платформу. Это сократит время, затрачиваемое на подачу заявки, и уменьшит возможность ошибок при заполнении.

Другой важной особенностью проекта является создание единой базы данных о деятельности студентов в различных направлениях. Это позволит ответственным за рейтинг легче и быстрее отслеживать достижения студентов в различных областях и принимать решения о назначении рейтинговой стипендии, поможет формировать статистику вовлеченности студентов в жизни ВУЗа.

Кроме того, проект «Рейтинг студента» позволит лучше отслеживать деятельность коллективов и учет внутренней деятельности. Это поможет более эффективно организовывать и проводить мероприятия, а также лучше поддерживать и развивать коллективы.

Помимо автоматизации процессов, проект «Рейтинг студента» позволит более точно и объективно определить самых активных студентов в университете. Вместо субъективных оценок и мнений, будут использоваться данные о реальной деятельности студентов, которые будут собираться и обрабатываться в единой системе.

Это также позволит студентам более точно планировать свою учебную и внеклассную деятельность, учитывая требования к получению рейтинговой стипендии. Кроме того, система рейтинговой стипендии может стать мощным стимулом для студентов, поскольку это позволит им не только получать финансовую поддержку, но и подтверждать свой активный образ жизни и лидерские качества.

В целом, проект «Рейтинг студента» позволит университету более эффективно управлять студенческой деятельностью и поощрять наиболее активных и успешных студентов. В результате университет сможет стать более привлекательным для потенциальных студентов и работодателей, укрепить свое лидерство в области высшего образования и научных исследований.

1.2 Учет мероприятий

Учет мероприятий является одним из важных элементов системы рейтинговой стипендии в университете. Данный процесс позволяет студентам получать баллы и повышать свои шансы на получение рейтинга.

В рамках Иркутского национального исследовательского технического университета мероприятия разделены по пяти направлениям рейтинговой системы:

1. Академические мероприятия: достижения отличных показателей в учебной деятельности, выступление на симпозиумах, семинарах, мастер-классах, лекциях и других мероприятиях, связанных с академической деятельностью студентов;
2. Научные мероприятия: участие в научных клубах, проведение научных исследований, экспериментов, разработка научных проектов. Также может включать представление научных работ, публикации в научных журналах, участие в научных проектах и исследованиях, защиту докладов и другие активности, связанные с научной деятельностью студентов внутри и вне университетской образовательной программы;
3. Спортивные мероприятия: участие в спортивных командах, соревнованиях, занятиях спортом, достижение спортивных рекордов и титулов;
4. Общественные мероприятия: участие в общественной деятельности, волонтерство, благотворительные акции, социальные проекты, организация мероприятий для социально уязвимых групп, участие в студенческих организациях, проведение мероприятий, направленных на укрепление общественного духа и формирование социальной активности студентов;
5. Культурные мероприятия: участие в театральных постановках, музыкальных концертах, художественных выставках, литературных чтениях, конкурсах культурного творчества и других мероприятиях, связанных с развитием творческих и культурных навыков студента.

В рамках рейтинговой стипендии ИРНИТУ мероприятия подразделены на внутренние и внешние. Внутренние мероприятия — это мероприятия, организованные на базе самого ВУЗа или его структурных подразделений, таких как факультеты, институты, кафедры и другие академические или научные подразделения университета.

Внешние мероприятия — это мероприятия, организованные вне университета, в академическом или научном сообществе, на научных площадках, научных конгрессах, симпозиумах, конференциях, соревнованиях и других мероприятиях, которые проводятся за пределами ВУЗа.

* 1. Описание предметной области

1.3.1 Словарь предметной области

Для создания словаря предметной области, приведенного в таблице 1.1, использовался метод Аббота. Метод Аббота предполагает разбиение предметной области на более мелкие блоки информации, называемые элементами [1]. Эти элементы могут быть любыми объектами или понятиями, связанными с предметной областью.

Таблица 1.1 – Словарь предметной области по Абботу

|  |  |  |
| --- | --- | --- |
| **Существительное** | **Глагол** | **Прочее** |
| Студент | Регистрироваться | Открытое |
| Ответственный | Регистрироваться | Открытое |
| Подтвердить | Присутствующий |
| QR-код | Сформировать |  |
| Развернуть |  |
| Мероприятие | Ознакомиться |  |
| Сохранить |  |

Продолжение таблицы 1.1 – Словарь предметной области по Абботу

|  |  |  |
| --- | --- | --- |
| Мероприятие | Отправить |  |
| Найти |  |
| Отфильтровать |  |
| Выбрать | Открытое |
| Календарь | Отметить | Зарегистрированное |

1.3.2 Объектно-ориентированный словарь предметной области

Объектно-ориентированный словарь предметной области — это словарь, содержащий термины, которые используются в контексте программной реализации системы, ориентированной на объекты. В этом словаре каждый термин представлен в виде объекта, имеющего свои свойства и методы.

В таблице 1.2 приведен объектно-ориентированный словарь предметной области.

Таблица 1.2 – Объектно-ориентированный словарь предметной области

|  |  |  |
| --- | --- | --- |
| **Классы** | **Свойства** | **Методы** |
| Мероприятие | Название | Просмотреть мероприя­тие() |
| Описание |  |
| Время |  |
| Дата |  |
| Локация |  |
| Количество участников |  |
| Тип мероприятия |  |
| Мероприятие | Теги мероприятия |  |
| Участник мероприятия |  |
| Регистрация |  |
| Мои мероприятия | Мероприятие |  |
| QR-код |  |
| Пользователь |  |
| pdf-файл |  |
| Календарь |  |
| Пользователь | Имя пользователя | Авторизоваться в си­стеме() |
| Изображение пользова­теля | Авторизоваться в си­стеме с помощью Кам­пуса() |
| Учебная группа |  |
| Статус пользователя |  |
| Участник мероприятия | ФИО участника | Просмотреть список участников() |

Продолжение таблицы 1.2 – Объектно-ориентированный словарь предметной области

|  |  |  |
| --- | --- | --- |
| **Классы** | **Свойства** | **Методы** |
|  | Учебная группа |  |
| Статус присутствия |  |
| Статус пользователя | Участник |  |
| Ответственный |  |
| Зритель |  |
| Статус присутствия | Присутствие | Отметить присутствие() |
| QR-код | Uid пользователя | Просмотреть QR-код() |
|  | Сформировать QR-код() |
| Pdf-файл |  | Сохранить pdf-файл() |
|  | Отправить pdf-файл() |
| Календарь |  | Добавить мероприятие в календарь() |
| Регистрация |  | Отменить регистрацию на мероприятие() |
|  | Регистрировать на ме­роприятие() |

1.4 Постановка задачи

В рамках проекта «Рейтинг студента» необходимо разработать приложение, пользователями которой являются студенты и ответственные за мероприятия. Для студентов приложение должно предоставлять возможность просмотра информации о мероприятии, регистрации на него и получения уникального QR-кода участника мероприятия. Данный QR-код будет использоваться для подтверждения фактического присутствия студента на мероприятии.

Для ответственных за мероприятия приложение должно предлагать функциональность считывания QR-кода и тем самым подтверждения фактического присутствия студента на мероприятии. Это будет осуществляться путем сканирования QR-кода с помощью камеры устройства, на котором установлено приложение.

Одновременно с реализацией приложения необходимо осуществить его интеграцию с разрабатываемой в рамках проекта «Рейтинг студента» системой. Для этого приложение должно связываться с информационной системой посредством API-запросов, чтобы передавать и получать информацию о мероприятиях, регистрации на студентов на них, подтверждении присутствия и других действиях, связанных с учетом посещаемости студентами мероприятий.

Основные функции разрабатываемого приложения должны включать:

1. Просмотр информации о мероприятиях. Пользователи системы должны видеть список доступных мероприятий и получать подробную информацию о каждом из них, такую как дата, время, место проведения и описание мероприятия;
2. Регистрация студентов на мероприятия и получение уникальных QR-кодов. Каждый зарегистрированный студент должен получить уникальный QR-код, который будет использоваться для подтверждения его фактического присутствия на мероприятии;
3. Подтверждение фактического присутствия студента на мероприятии. Для этого студент должен предъявить свой QR-код при входе на мероприятие, и система автоматически подтвердит его присутствие после сканирования кода ответственным за мероприятие;
4. Просмотр статистической информации о присутствующих на мероприятии студентах. Ответственные за мероприятия смогут увидеть количество зарегистрированных студентов и количество студентов, которые действительно присутствовали на мероприятии.

Конечное приложение направлено на управление мероприятиями и регистрацию студентов на них. Для реализации проекта необходимо разработать программную логику, которая будет обрабатывать запросы пользователей и ответственных за мероприятия и осуществить настройку интеграции с разрабатываемой информационной системой «Рейтинг студента».

Функциональность системы должна позволять пользователям просматривать информацию о доступных мероприятиях, включая дату, время, место проведения и описание мероприятия. Регистрация студентов на мероприятия должна быть простой и удобной, а каждый зарегистрированный студент должен получить уникальный QR-код, который будет использоваться для подтверждения его фактического присутствия на мероприятии.

Для подтверждения фактического присутствия студента на мероприятии, необходимо, чтобы студент предъявил свой QR-код на входе на мероприятие, и система автоматически подтвердила его присутствие после сканирования кода ответственным за мероприятие.

Ответственные за мероприятия должны иметь возможность просмотра статистической информации о присутствующих на мероприятии студентах, включая количество зарегистрированных студентов и количество студентов, которые действительно присутствовали на мероприятии.

Для успешной реализации проекта необходимо также разработать удобный интерфейс для пользователей системы, который будет удобен и легок в использовании.

1.5 Требования к ролям и правам доступа

При разработке приложения для регистрации на мероприятия необходимо учитывать различия в функционале и правах доступа для разных типов пользователей.

Первый тип пользователя – студент, который может просматривать доступные мероприятия, выбирать и регистрироваться на них, а также подтверждать свое присутствие на мероприятии.

Второй тип пользователя – ответственный за мероприятие, который может регистрироваться на мероприятия в качестве ответственных, просматривать информацию о мероприятиях. Также данному пользователю доступны возможности просмотра списка зарегистрированных студентов на мероприятие и подтверждение их присутствия на мероприятии.

Роли и функции пользователей представлены в таблице 1.3.

Таблица 1.3 – Роли и функции пользователей

|  |  |  |
| --- | --- | --- |
| **Роль** | **Подсистема** | **Функция** |
| Сту­дент | Мероприятия | просмотр списка предстоящих мероприятий |
| выбор мероприятия |
| просмотр подробного описания выбранного меро­приятия |
| поиск определенного мероприятия по названию |
| фильтрация мероприятий |
| регистрация на выбранное мероприятие в качестве участника |
| регистрация на выбранное мероприятие в качестве зрителя |
| Мои меро­приятия | просмотр списка мероприятий с пройденной реги­страцией |
| выбор мероприятия с пройденной регистрацией |
| демонстрация уникального идентификатора участ­ника мероприятия |
| сохранение уникального идентификатора |
| отправка уникального идентификатора |
| добавление мероприятия в календарь смартфона |
| отмена регистрации на мероприятие |
| Пользова­тель | возможность авторизации с помощью системы «Кампус» |
| возможность авторизации с помощью логина и па­роля |
| возможность выхода из учетной записи пользова­теля |
| Ответ­ствен­ный за меро­прия­тие | Мероприятия | просмотр списка предстоящих мероприятий |
| выбор мероприятия |
| регистрация на мероприятие в качестве ответ­ственного |
| просмотр подробного описания выбранного меро­приятия |
| фильтрация мероприятий |
| поиск определенного мероприятия по названию |

Продолжение таблицы 1.3 – Роли и функции пользователей

|  |  |  |
| --- | --- | --- |
| **Роль** | **Подсистема** | **Функция** |
| Ответ­ствен­ный за меро­прия­тие | Пользова­тель | возможность авторизации с помощью системы «Кампус» |
| возможность авторизации с помощью логина и па­роля |
| возможность выхода из учетной записи пользова­теля |
| Подтвержде­ние присут­ствия | выбор необходимого мероприятия |
| считывание уникальный код участника студента |
| подтверждение присутствие участника |
| Мои меро­приятия | просмотр списка мероприятий с пройденной реги­страцией | |
| выбор мероприятия с пройденной регистрацией | |
| просмотр статистики по выбранному мероприятию | |

1.6 Функциональное назначение системы

Функциональное назначение приложения представлено в виде нотации IDEF0. Эта графическая модель используется для описания функций системы, их взаимодействия и зависимостей.

Описание контекста модели:

1. Предмет моделирования – регистрация и подтверждение присутствия студентов на мероприятии;
2. Область моделирования – модель охватывает процессы, связанные проведением мероприятий;
3. Точка зрения – студенты и ответственные за мероприятия.

На рисунке 1.2 представлен общий процесс разрабатываемой системы. Данный блок декомпозируется на процессы авторизации в системе, отображения мероприятий, регистрации на них и последующего подтверждения физического присутствия на событии.

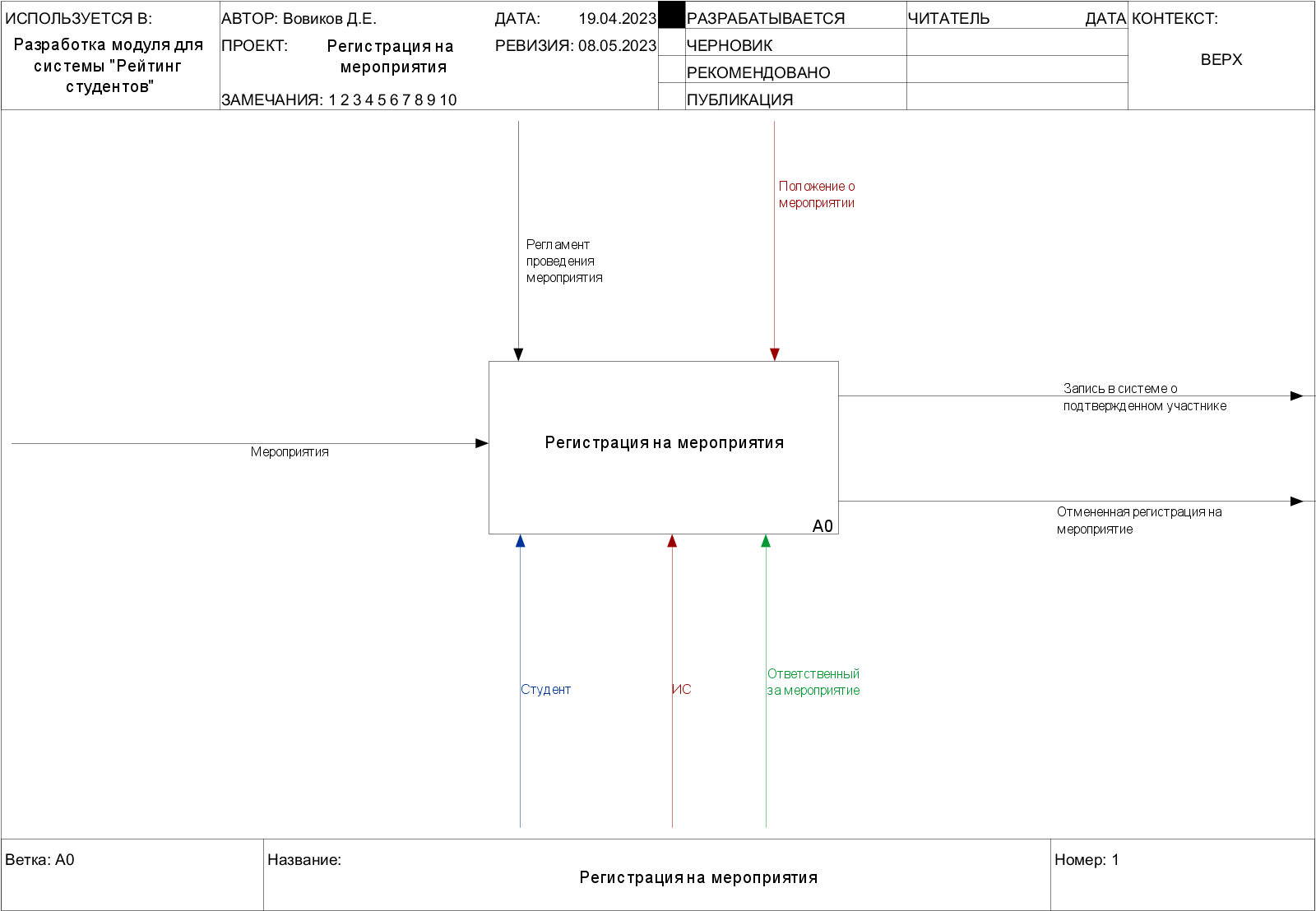


Рисунок 1.2 – Блок «Регистрация на мероприятия»

Декомпозиция блока А0 представлена на рисунке 1.3. Данный процесс декомпозируется на четыре подпроцесса:

1. Авторизоваться в системе. Данный процесс описывает способы авторизации пользователя в модуле информационной системы для получения возможностей дальнейшего взаимодействия с ним;
2. Отобразить мероприятия. Данный процесс описывает набор действий пользователя с мероприятиями;
3. Зарегистрироваться на мероприятие. Процесс включает в себя этапы, доступные пользователю при прохождении процедуры регистрации на мероприятие;
4. Подтвердить присутствие. Процесс описывает действия ответственного за мероприятие при выполнении процедуры подтверждения фактического присутствия зарегистрированного студента на мероприятии.

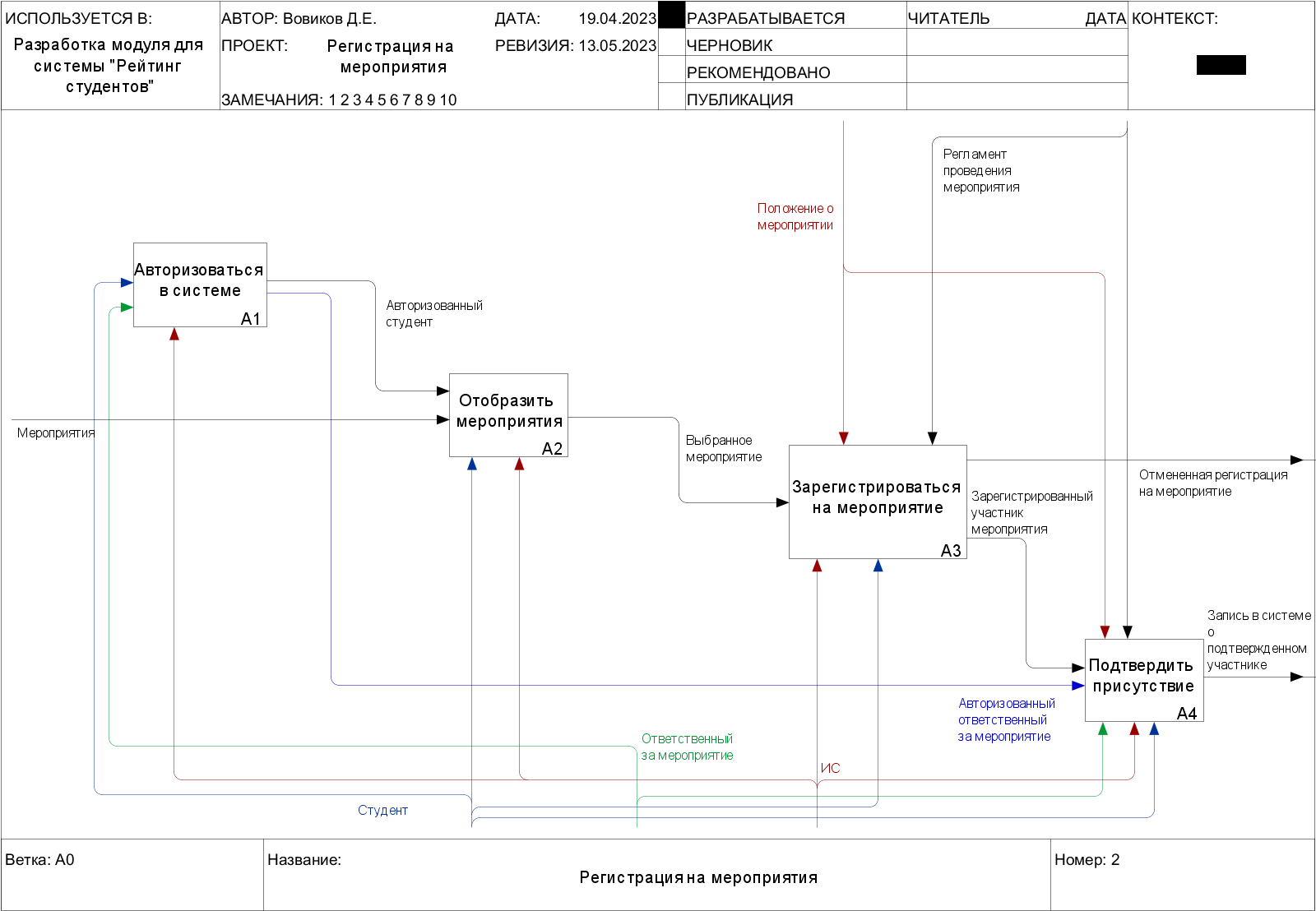


Рисунок 1.3 – Блок «Регистрация на мероприятия»

На рисунке 1.4 представлена декомпозиция блока А1 «Авторизоваться в системе».

Данный процесс декомпозируется на два этапа:

1. Авторизоваться с помощью логина и пароля. Студент и ответственный за мероприятие вводят данные от аккаунта в соответствующие поля в приложении. После система производит процесс валидации введенных данных и связывается с сервером ИС «Рейтинг студента». Информационная система проверяет введенные пользователем данные и на основании присвоенной в базе данных роли выдает соответствующий доступ к возможностям программного модуля: для студента – возможность регистрации на мероприятия, для ответственного – возможность подтверждения фактического присутствия участника;
2. Авторизоваться через Кампус. Студент и ответственный за мероприятие используют данные от аккаунта Кампуса для авторизации в приложении. В зависимости от полученного результата приложение выдает соответствующий доступ к возможностям.

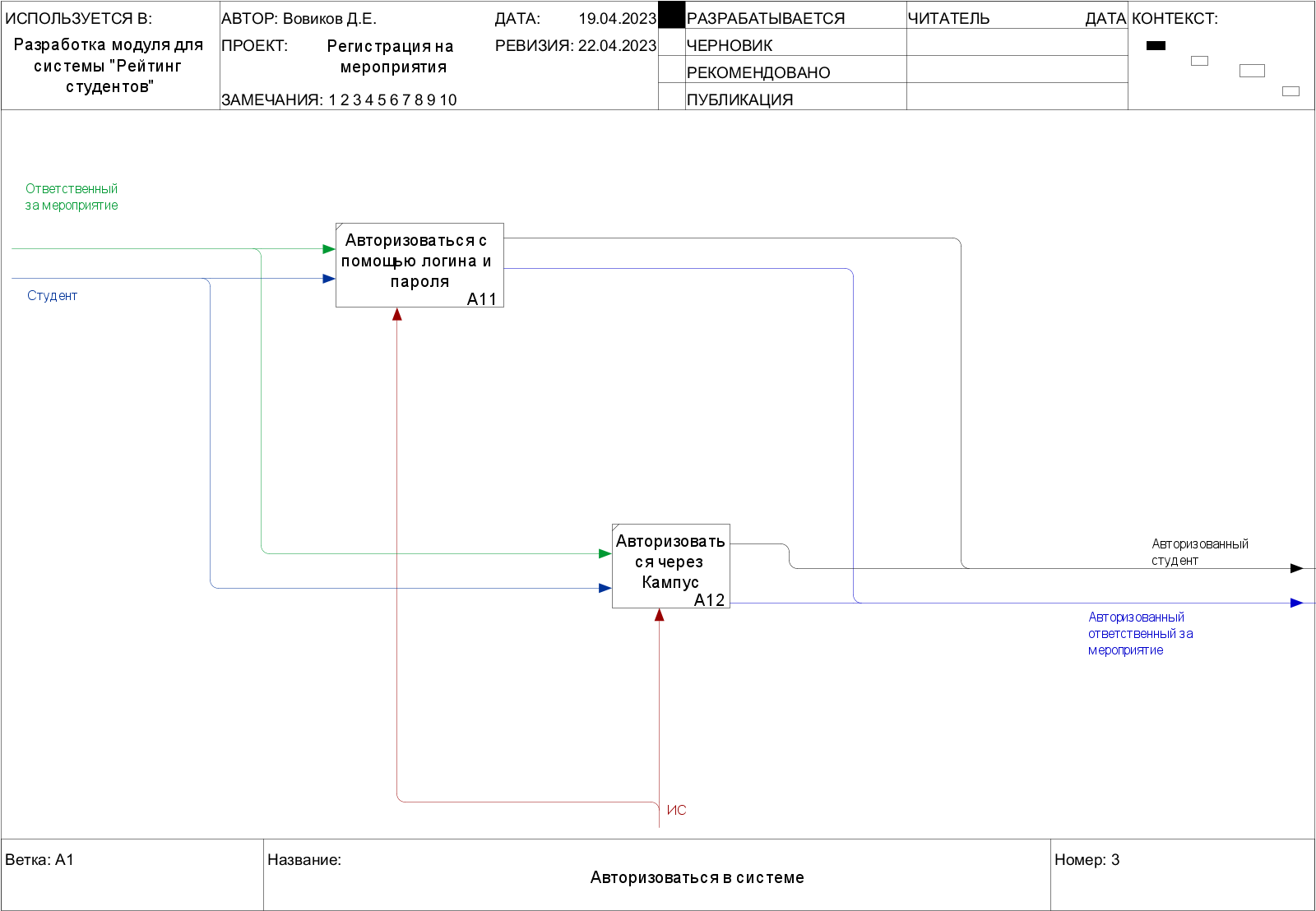


Рисунок 1.4 – Декомпозиция блока «Авторизоваться в системе»

На рисунке 1.5 представлена декомпозиция процесса А2 «Отобразить мероприятия».

Данный процесс декомпозируется на четыре этапа:

1. Открыть список мероприятий. Приложение получает перечень актуальных мероприятий и собирает их в список. Данный список отображается авторизованному студенту;
2. Отфильтровать мероприятия. Пользователь применяет фильтр мероприятий по необходимому критерию;
3. Найти определенное мероприятие. Пользователь применяет механизм поиска по названию для получения необходимого мероприятия;
4. Выбрать мероприятие. Пользователь выбирает необходимое мероприятие для просмотра подробной информации о нем.

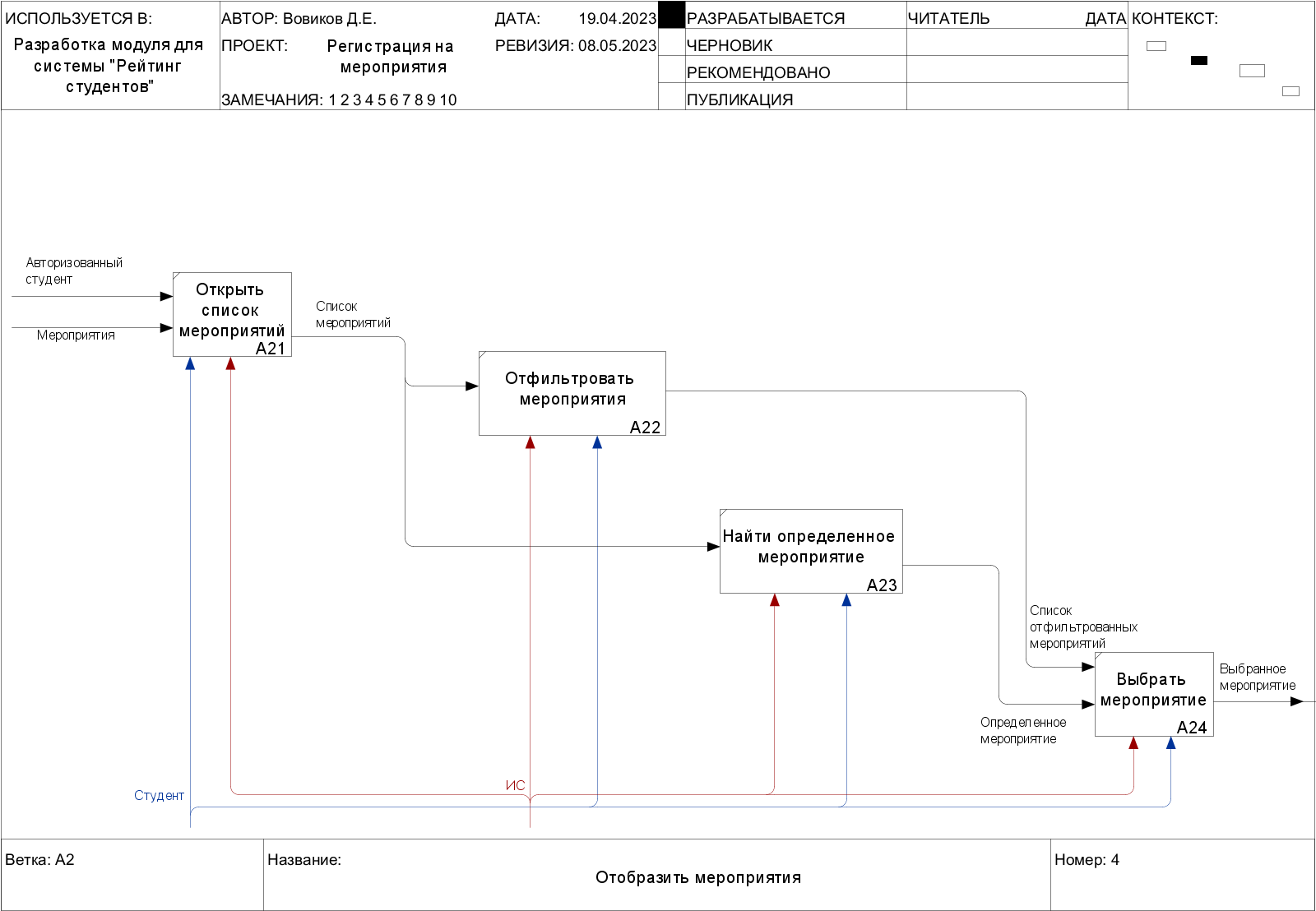


Рисунок 1.5 – Декомпозиция блока «Отобразить мероприятия»

На рисунке 1.6 представлена декомпозиция процесса А3 «Зарегистрироваться на мероприятие».

Данный процесс декомпозируется на четыре этапа:

1. Пройти регистрацию на мероприятие. На первом этапе студент проходит процедуру регистрации на выбранное им мероприятие. В результате в систему добавляется запись о регистрации. После прохождения данного этапа пользователю становятся открыты раннее недоступные процессы;
2. Отменить регистрацию на мероприятие. Данный процесс удаляет из системы запись пользователя о регистрации;
3. Добавить мероприятие в календарь смартфона. Студент может добавить мероприятие в календарь своего смартфона для создания напоминания;
4. Сформировать QR-код. После прохождения регистрации система формирует уникальный QR-код участника мероприятия с ключом пользователя;
5. Сохранить QR-код в pdf-файл. Пользователю доступна возможность сохранить QR-код и набор информации о событии на устройство в виде pdf-файла;
6. Отправить pdf-файл в мессенджер. Пользователю доступна возможность отправки pdf-файла в мессенджер или на электронную почту.
7. Посетить мероприятие в указанное время. Для подтверждения своего фактического присутствия студенту необходимо посетить мероприятие в указанное время.

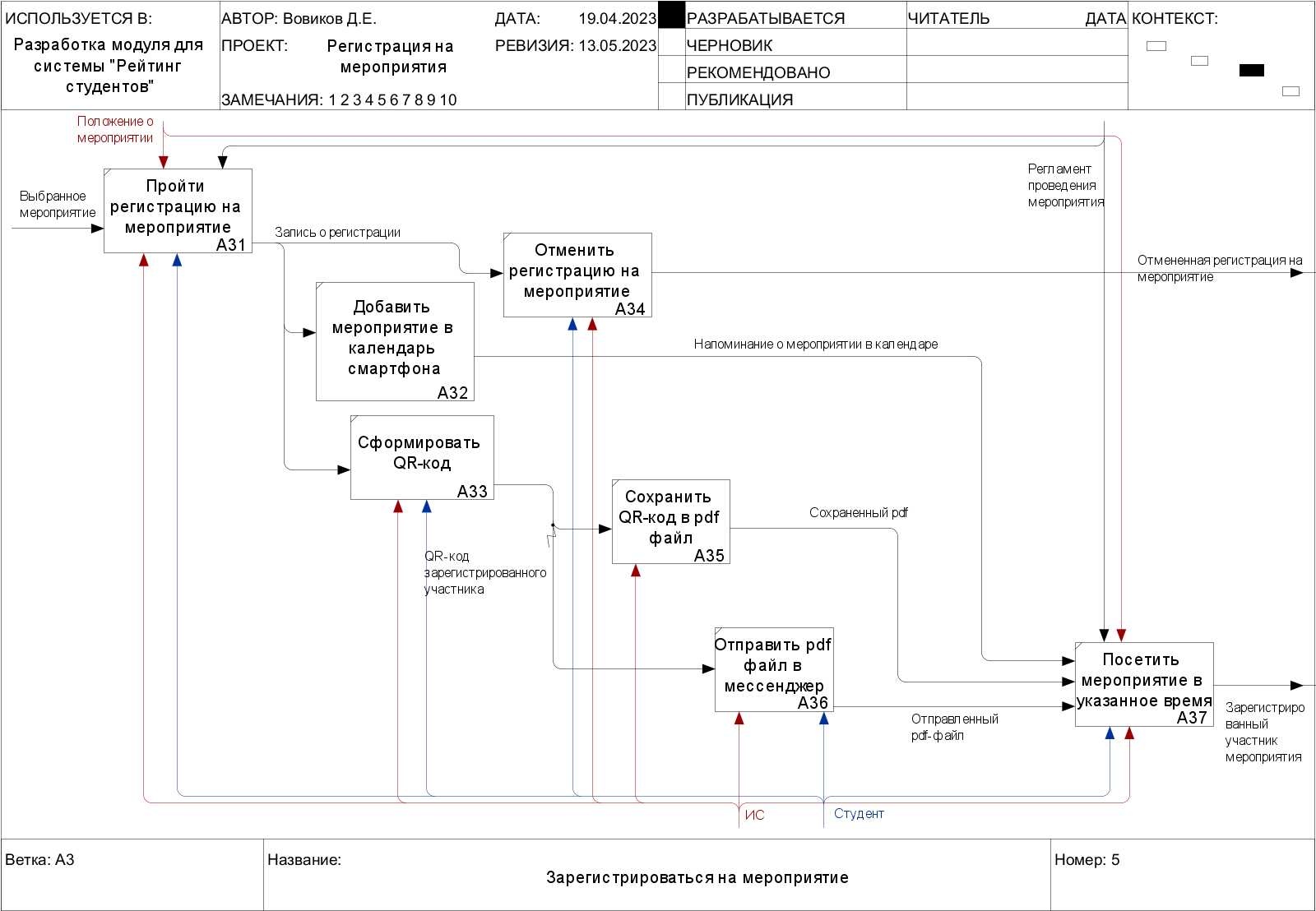


Рисунок 1.6 – Декомпозиция блока «Зарегистрироваться на мероприятие»

На рисунке 1.7 представлена детализация блока А4 «Подтвердить присутствие». Данный процесс декомпозируется на несколько этапов:

1. Открыть список закрепленных для проведения мероприятий. Авторизованный ответственный за мероприятие открывает список мероприятий, в которых он является закрепленным для проведения;
2. Выбрать мероприятие. На данном этапе ответственный выбирает из списка необходимое мероприятие. После выбора пользователь может перейти на этап три или пропустить его и перейти на этап четыре;
3. Просмотреть список зарегистрированных участников. Ответственный просматривает список участников, в котором отображается следующая информация: ФИО студента, группа, присутствие на мероприятии;
4. Запустить механизм подтверждения. Ответственный за мероприятие запускает механизм считывания QR-кодов участников мероприятия;
5. Подтвердить присутствие участника. Ответственный сканирует QR-код участника и тем самым подтверждает фактическое присутствие человека на событии.

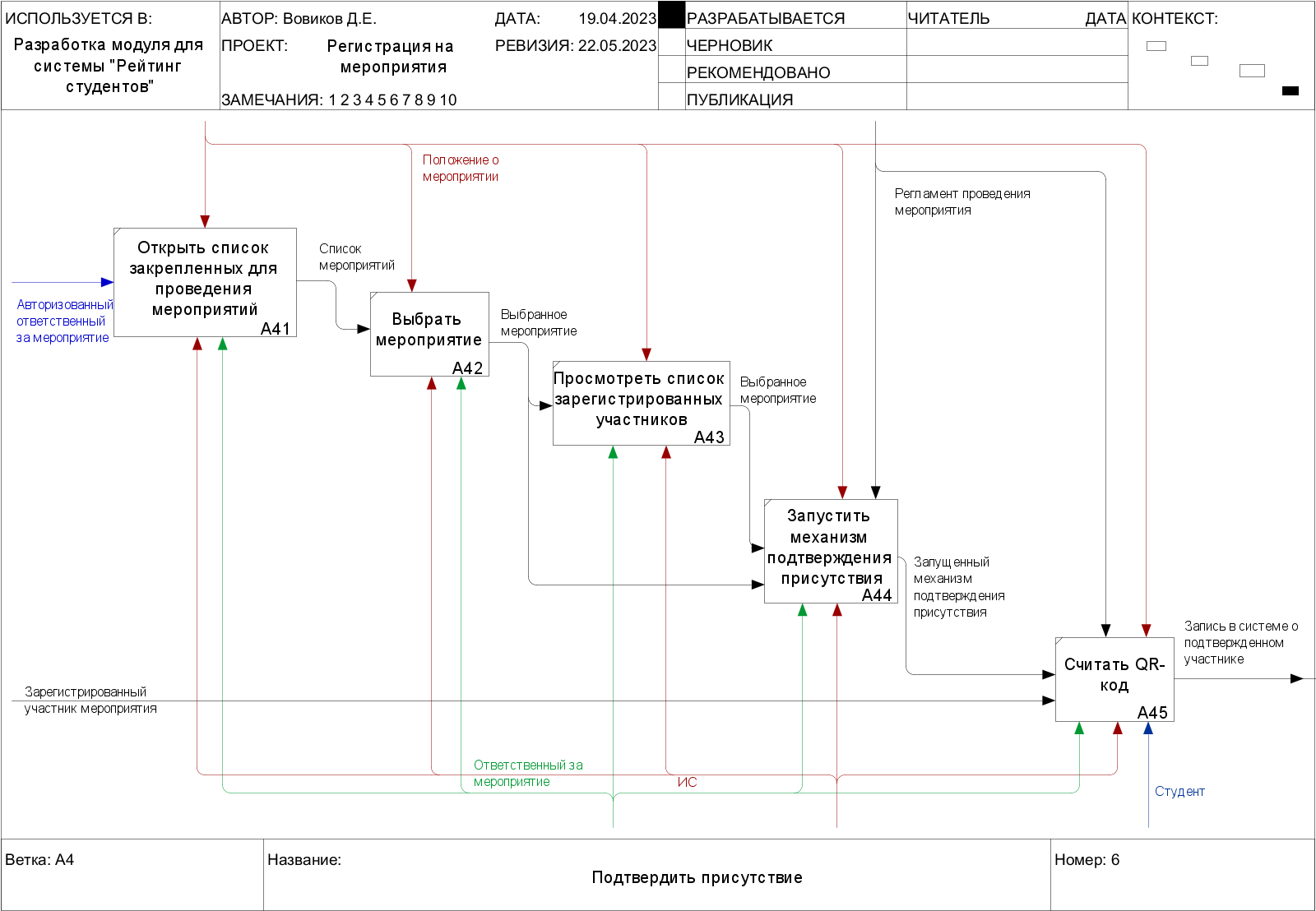


Рисунок 1.7 – Декомпозиция блока «Подтвердить присутствие»

1.7 Формирование функциональных требований пользователей к системе

Для описания функционала разрабатываемого программного модуля была спроектирована диаграмма вариантов использования системы. В системе выделены два активных субъекта: студент и ответственный за мероприятие. Студент использует приложение для просмотра и регистрации на участие в мероприятиях. Ответственный за мероприятие – для подтверждения присутствия участников.

Формирование функциональных требований является важным инструментом для разработки системы, поскольку она помогает определить требования к функциональности системы и обеспечивает связь между требованиями и реализацией системы.

На рисунке 1.8 представлена диаграмма формирования функциональных требований для приложения.

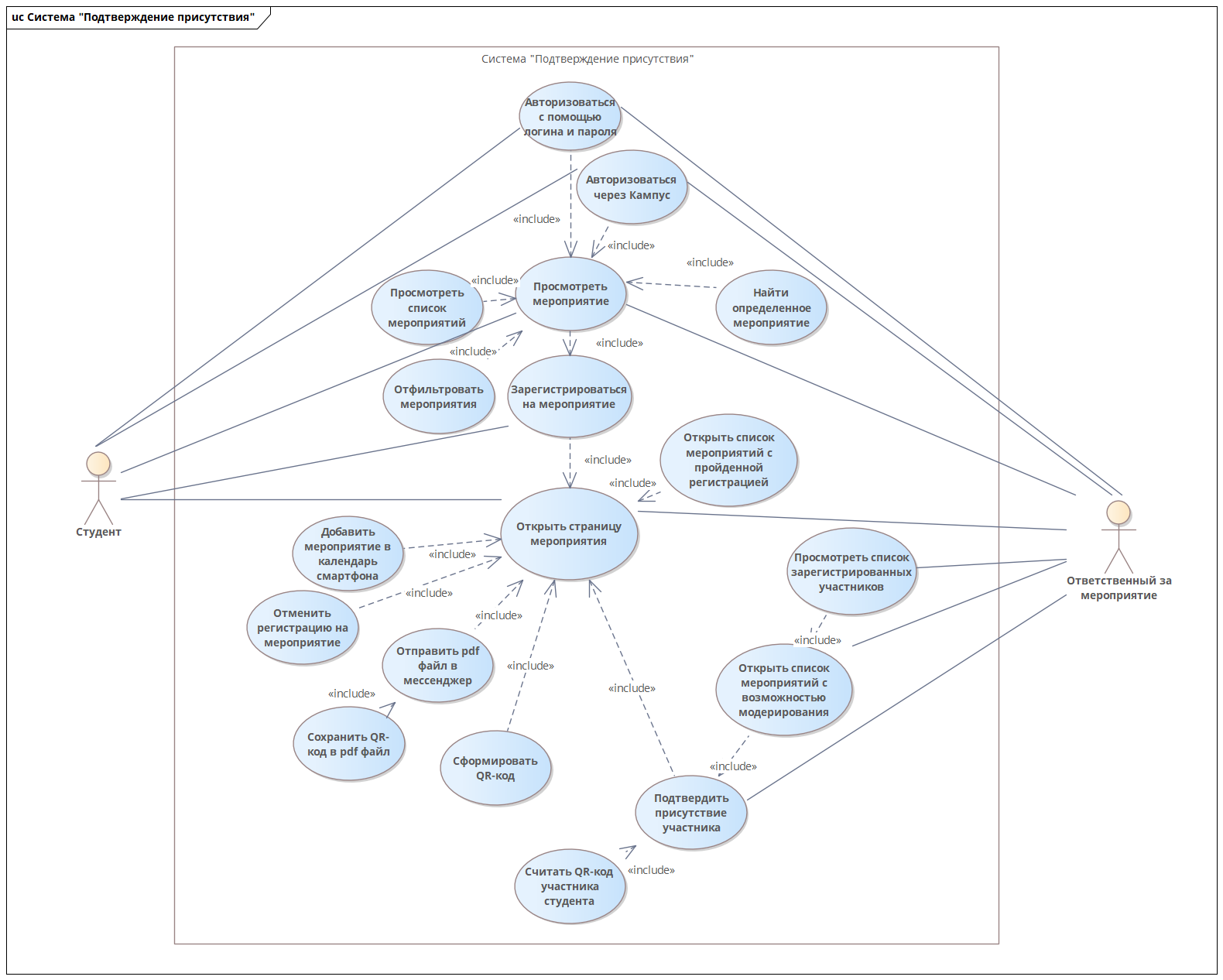


Рисунок 1.8 – Диаграмма формирования функциональных требований

1.8 Обзор аналогов

Для проведения анализа конкурентов были выбраны три интернет-площадки для создания мероприятий.

**RUNET-ID** — крупнейший российский интернет-ресурс, посвященный отечественной интернет-отрасли. В новой версии портал объединил в себе календарь основных мероприятий Рунета, отраслевое СМИ, видео, а также базу вакансий в интернет-компаниях [3]. RUNET-ID предлагает широкий набор возможностей для создания мероприятий крупного масштаба, таких как конференции, форумы. Платформа предоставляют возможность настройки регистрационных форм, определения типов билетов, создания программы мероприятия, а также интеграции с платежными системами. Однако, из-за своей сложности и масштабности, RUNET-ID может быть не оптимальным решением для создания небольших внутренних мероприятий, таких как студенческие мероприятия в учебных заведениях.

**Leader-ID** — это возможность присоединиться к миллионам предпринимателей, исследователей, студентов и волонтеров, чтобы получить доступ к Точкам кипения по всей стране и десяткам мероприятий каждый день [4].

Платформа позволяет организовать процесс регистрации участников мероприятий. Участники могут регистрироваться онлайн, заполнять регистрационные формы.

Leader-ID предоставляет инструменты для взаимодействия между участниками мероприятий. Участники могут обмениваться сообщениями, создавать профили, вступать в группы обсуждения, делиться материалами и сетевыми контактами.

Платформа позволяет организаторам проводить мероприятия как в режиме онлайн, так и в очном режиме на специализированных площадках «Точка Кипения». Мероприятия могут быть различных форматов, такие как вебинары, видеоконференции, чаты и другие форматы. Организаторы могут создавать программу мероприятий, назначать спикеров, устанавливать режимы доступа и проводить взаимодействие с участниками в режиме реального времени.

Leader-ID предоставляет инструменты для анализа и отчетности о проведенных мероприятиях. Организаторы могут получать данные о регистрациях, активности участников, оценках, отзывах и других показателях, что позволяет оценить эффективность мероприятий и принимать решения на основе данных.

Однако, Leader-ID также имеет свои недостатки, одним из которых является отсутствие фактического подтверждения присутствия участников на мероприятии в электронном формате. Кроме того, привязка мероприятий к Точке Кипения может быть неудобной и ограничивающей фактором проведения события.

**ФГАИС Молодежь России** – единственный профильный общенациональный орган государственной власти, деятельность которого целиком сосредоточена на развитии молодых граждан России.

Агентство рассматривает молодёжь как главный ресурс развития страны, и именно на этом убеждении строит свою работу в рамках реализации государственной молодёжной политики России. Росмолодёжь призвана создавать возможности для молодого поколения и координировать усилия для создания условий его самореализации.

ФГАИС Молодежь России также осуществляет работу по социальной поддержке молодежи, оказывает консультативную и методическую помощь молодежным организациям и их руководителям, и проводит проекты по поддержке молодежного предпринимательства, трудоустройству молодежи и развитию их профессиональных навыков.

Целью ФГАИС Молодежь России является создание условий для развития молодежного движения, активного участия молодежи в социально-экономической и культурной жизни страны, и поддержки молодежных инициатив в различных сферах деятельности.

Основная задача ФГАИС Молодежь России состоит в поддержке молодежных инициатив, развитии молодежной политики, и реализации государственных программ и проектов в сфере молодежи. Организация проводит широкий спектр деятельности, включая поддержку молодежных организаций, развитие молодежных центров, организацию молодежных форумов, фестивалей, семинаров, тренингов, и других мероприятий, направленных на развитие молодежного движения и поддержку молодежных инициатив. Однако, для создания мероприятий на базе данной платформы необходимо быть партнером ФГАИС «Молодежь России» и иметь аккаунт администратора. Кроме того, организация малых мероприятий на данной платформе может быть затруднительной из-за ориентации на массовые мероприятия [5].

1.9 Анализ базы данных информационной системы «Рейтинг студента»

База данных проекта «Рейтинг студента», представленная на рисунке 1.9, учитывает роли пользователей, коллективы, анкеты для вступления и мероприятия. Для приложения в проекте задействован следующий набор сущностей:

1. Мероприятия - данные о мероприятиях, такие как название, дата, время, место и описание. Также в базе данных хранятся теги мероприятий и виды мероприятий, чтобы упростить поиск нужных мероприятий;
2. Журнал - сущность, которая хранит данные о присутствии людей с той или иной ролью на мероприятии и их результатах на мероприятии. Журнал связывается с мероприятием и соответствующими пользователями;
3. Пользователи - сущность, которая хранит данные о студентах, такие как ФИО, электронная почта, пароль, группа, институт и ID роли. Пользователи связываются с журналом и ролями;
4. Роли - сущность, которая хранит названия ролей и соответствующие разрешения. Роли используются для определения уровня доступа пользователей к определенным функциям приложения.

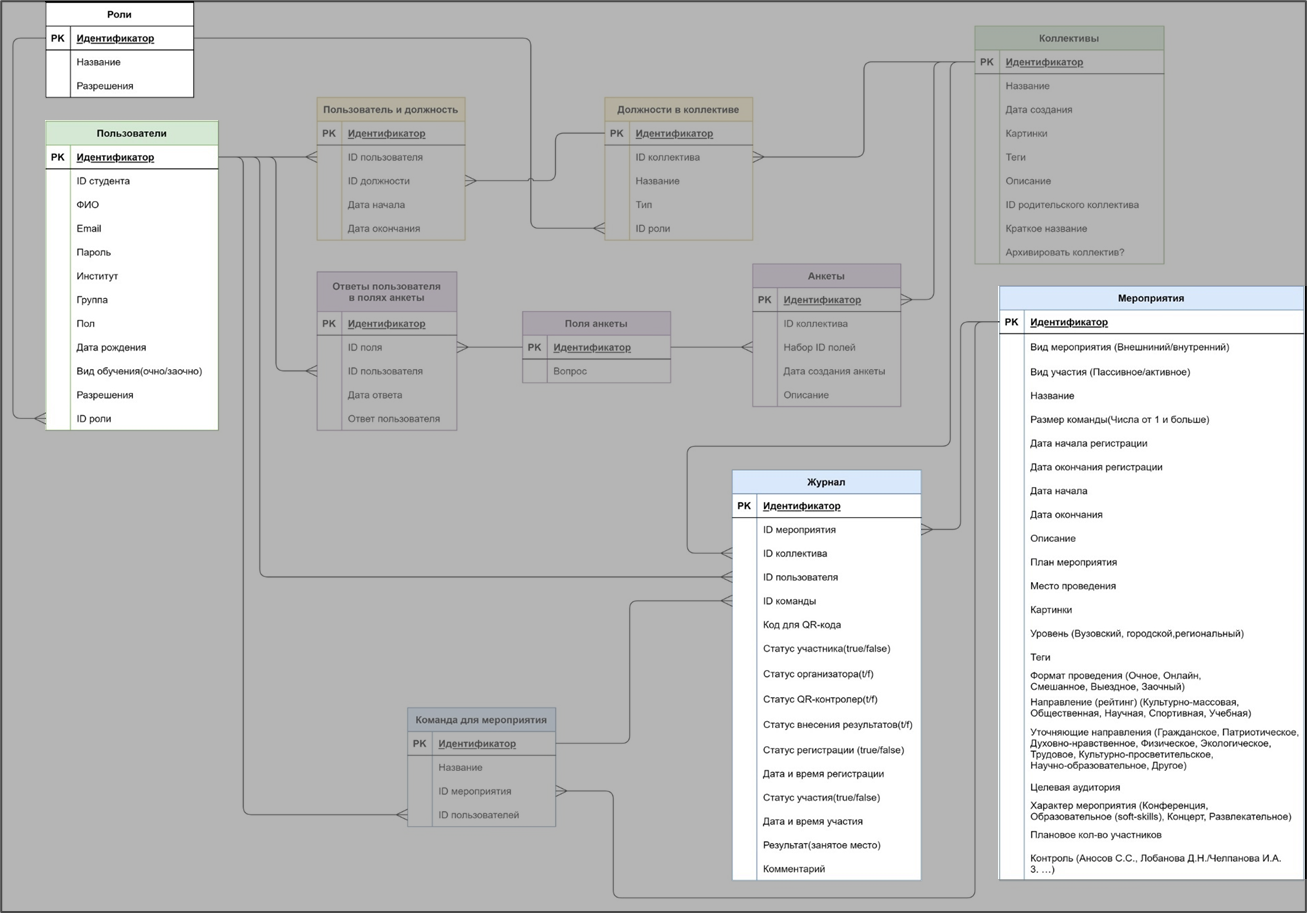


Рисунок 1.9 – База данных ИС «Рейтинг студента»

1.10. Требования графического интерфейса приложения

Для реализации графического интерфейса спроектирован набор экранов.

На рисунке1.10. представлена диаграмма интерфейсных экранов приложения.

Зеленым цветом помечены общие для ролей «Студент» и «Ответственный за мероприятие» экраны. К ним относятся:

1. Основной экран приложения;
2. Экран настроек;
3. Экраны профиля;
4. Экран страницы мероприятия;
5. Экран мероприятий пользователя.

Белым – экраны интерфейса для роли «Студент». К ним относятся:

1. Экран страницы мероприятия с пройденной регистрацией;
2. Экран отмены регистрации;
3. Экран QR-кода регистрации;
4. Экран регистрации на мероприятие.

Красным – для роли «Ответственный за мероприятие».

Изображение содержит название экранов и набор функций. К ним относятся:

1. Экран статистики посещаемости мероприятия;
2. Экран информации о мероприятии;
3. Экран подтверждения присутствия;
4. Экран сканера QR-кода;
5. Экраны результата сканера и подтверждения присутствия.

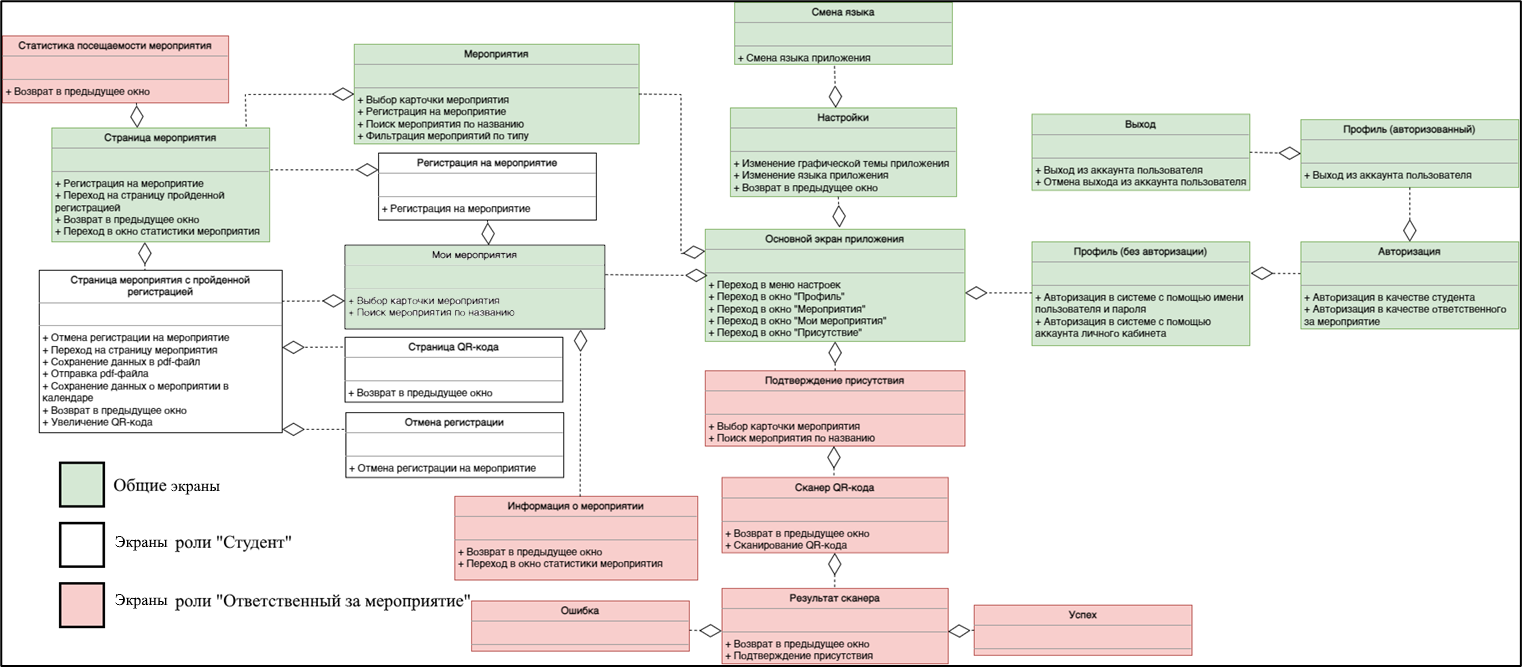


Рисунок 1.10 – Диаграмма интерфейсных экранов приложения

1.11 Вывод по главе

В ходе анализа предметной области были выявлены ключевые требования к разработке приложения. Одним из главных требований является реализация двух вариантов использования разрабатываемого приложения: для студента и для ответственного за проведение мероприятий.

В процессе анализа предметной области были изучены различные альтернативы, предлагающие решения для организации мероприятий. Однако, каждый из них обнаружился с ограничениями, которые могут затруднить их использование в полной мере. Исходя из этого, решено было осуществить интеграцию разрабатываемого приложения с собственной системой, чтобы преодолеть эти ограничения и обеспечить более гибкую и эффективную организацию мероприятий. Такой подход позволит максимально использовать преимущества собственной системы и разрабатываемого приложения, обеспечивая более удовлетворительный опыт использования и повышая конкурентоспособность решения.

Кроме того, в ходе анализа предметной области были составлены требования к ролям и правам доступа, которые будут установлены в разрабатываемом приложении. Данные требования описывают функционал приложения для двух ролей – студент и ответственный за мероприятия. Для студента необходимо реализовать функционал по поиску и просмотру мероприятий, регистрации на мероприятия, отмене регистрации, авторизации в системе. Для ответственного за мероприятия необходимо разработать функции по просмотру списка зарегистрированных студентов и подтверждению фактического присутствия на мероприятии. Такой подход к установлению прав доступа позволит эффективно реализовать потребности пользователей и обеспечить удобный и интуитивно понятный интерфейс для работы с приложением.

Помимо вышеупомянутых требований к ролям и правам доступа, в процессе анализа было сформировано функциональное назначение системы и функциональные требования пользователей к системе. Функциональное назначение системы заключается в просмотре мероприятий, регистрации на них и последующего подтверждения присутствия через приложение. Функциональные требования пользователей включают в себя возможность быстрого и удобного поиска мероприятий, просмотра информации о них, возможность регистрации и отмены регистрации на мероприятия, а также подтверждением присутствия и просмотра статистики по мероприятиям для ответственных за их проведение. Дополнительно составлены функциональные требования для графического интерфейса. Учет этих функциональных требований необходим для удовлетворения потребностей пользователей и создания эффективного приложения. В результате анализа предметной области и функциональных требований к системе сделан вывод, что необходимо разработать мобильное приложение.

2 Проектирование системы

2.1 Обоснование использования программных средств разработки

Выбор оптимальных инструментов и технологий может существенно снизить время разработки, повысить производительность и обеспечить высокое качество конечного продукта. Правильное решение на этапе выбора технологий – ключевой фактор успешной разработки мобильного приложения для Android.

**Kotlin** – это статически типизированный язык программирования, разработанный компанией JetBrains, работающий на платформе Java Virtual Machine (JVM) и являющийся официальным языком программирования для разработки Android-приложений, как и Java. Данный язык программирования является полностью совместимым с Java и может быть без проблем интегрирован в существующие проекты на Java. Это позволяет разработчикам постепенно переходить на Kotlin, сохраняя совместимость с существующим Java-кодом. Kotlin поддерживает функциональное программирование, что позволяет использовать лямбда-выражения, высокоуровневые функции, расширения функций и другие функциональные конструкции, тем самым упрощая разработку сложных функций и обработку данных [6].

**Material Design** – это дизайн-язык, разработанный компанией Google, который используется для создания современных и стильных пользовательских интерфейсов мобильных приложений, включая Android-приложения. Material Design предлагает набор готовых компонентов и рекомендаций по оформлению интерфейса, которые обеспечивают единый и согласованный внешний вид и поведение приложений на платформе Android. Material Design предлагает набор компонентов, таких как кнопки, поля ввода, списки, диалоговые окна и другие, выполненных в виде материальных объектов, которые имитируют физические объекты в трехмерном пространстве. Для разработки мобильных приложений на платформе Android с использованием Material Design, Google предоставляет библиотеку под названием «Material Components for Android», которая включает набор готовых компонентов и стилей, соответствующих принципам Material Design. Эта библиотека интегрируется в Android Studio и позволяет разработчикам легко создавать современные и стильные интерфейсы для своих приложений, соответствующие дизайн-языку Material Design [7].

**Volley** – это библиотека для обработки сетевых запросов в мобильных приложениях на платформе Android, разработанная компанией Google. Она предоставляет удобные и эффективные средства для работы с сетью, такие как отправка HTTP-запросов, обработка ответов, кэширование, управление очередью запросов и другие функции, что делает ее популярным инструментом для разработки мобильных приложений.

Volley также предлагает ряд дополнительных возможностей, таких как поддержка отмены запросов, обработка ошибок, автоматическая обработка cookie, поддержка множественных соединений и другие, что делает ее мощным инструментом для работы с сетью в мобильных приложениях на платформе Android [8].

**Draw.io** — это веб-приложение, предоставляющее возможность создавать диаграммы и схемы различных типов. Оно широко используется для создания блок-схем, потоковых диаграмм, организационных диаграмм, сетевых диаграмм, диаграмм классов, ER-диаграмм и многих других типов диаграмм [9].

Draw.io предлагает простой в использовании интерфейс с интуитивно понятными инструментами рисования и множеством готовых элементов, таких как блоки, стрелки, символы и формы, которые можно использовать для создания диаграммы. Пользователи могут создавать диаграммы «перетаскиванием и редактированием», добавлять текст, изменять цвета, размеры и стили элементов, а также настраивать соединения между элементами.

**Ramus** — это программное обеспечение для создания диаграмм требований и моделирования бизнес-процессов [10].

Ramus предлагает графический интерфейс, который позволяет создавать различные виды диаграмм, такие как диаграммы потоков данных (DFD), диаграммы IDEF0, диаграммы требований и другие. Он также предоставляет возможность создания таблиц, описывающих требования, атрибуты и другую дополнительную информацию о моделируемых системах.

Одна из основных особенностей Ramus – это поддержка структурного подхода к моделированию, такого как функциональное моделирование и моделирование потоков данных. Он также предлагает возможность анализировать и описывать требования системы, управлять версиями моделей, создавать отчеты и экспортировать модели в различные форматы файлов.

**GitHub** — это веб-платформа, предназначенная для хостинга и управления репозиториями (хранилищами) кода на базе системы контроля версий Git. GitHub предлагает удобные инструменты для разработчиков, которые позволяют им работать с кодом, вносить изменения, отслеживать историю изменений, сотрудничать с другими разработчиками, проводить рецензирование кода, управлять проектами и многое другое [11].

Одна из основных особенностей GitHub — это возможность совместной работы над проектами. Разработчики могут вносить изменения в репозиторий, создавать ветки и отправлять запросы на слияние (Pull Requests), что позволяет обсуждать изменения, проводить рецензирование кода и принимать решение о включении изменений в основную ветку проекта. Это способствует коллаборации и обмену знаниями между разработчиками.

В целом, GitHub является мощным инструментом для хостинга и управления репозиториями кода, с широким набором функций для совместной разработки, автоматизации процессов разработки и интеграции с другими инструментами разработки. Он широко используется разработчиками по всему миру и является важным инструментом в современной разработке программного обеспечения.

**Android Studio** – интегрированная среда разработки, специально разработанная для создания мобильных приложений на платформе Android. Она предлагает широкий набор функций, таких как автодополнение кода, рефакторинг, отладка, профилирование, визуальный макет редактор и другие инструменты. Android Studio поддерживает языки программирования Java, Kotlin и C++, а также различные фреймворки и библиотеки. Она также обладает богатыми возможностями тестирования и настраиваемым интерфейсом, что делает ее мощным инструментом для разработки высококачественных мобильных приложений на Android [12].

**Figma** – это онлайн-инструмент для дизайна интерфейсов и прототипирования. Он предоставляет возможности для создания графических макетов, векторной графики, анимации и коллаборации между дизайнерами и разработчиками.

Figma отличается от других инструментов дизайна интерфейса тем, что он полностью базируется на веб-технологиях и работает в браузере без необходимости установки дополнительного программного обеспечения [13].

2.2 Архитектура подсистемы

2.2.1 Общая архитектура системы

Общая архитектура системы представляет собой набор из двух компонентов: клиентским мобильным приложением и сервером. Обмен данными осуществляется в двустороннем формате. Серверное приложение обрабатывает запросы, взаимодействует с базой данным и отправляет результат в виде JSON-файла. Клиент обращается к серверу и получает данные, которые использует для заполнения соответствующих полей графического интерфейса.

Архитектура системы представлена на рисунке 2.1.

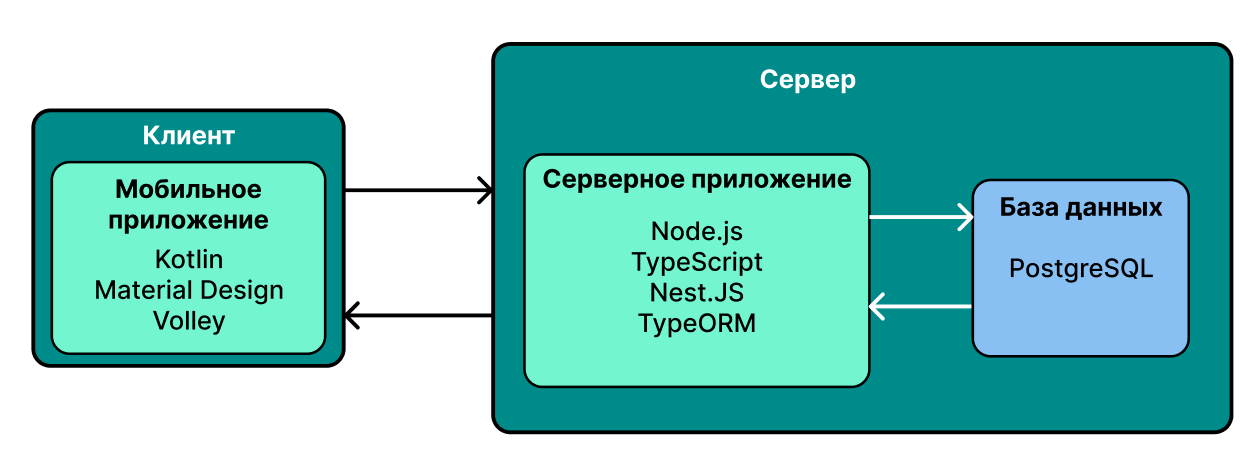


Рисунок 2.1 – Архитектура системы

В качестве данных выступают: список проводимых мероприятий с открытой регистрацией, информация о необходимом мероприятии, перечень участников мероприятия, информация об авторизованном пользователе.

2.2.2 Архитектура мобильного приложения

В мобильном приложении используется многослойная архитектура, которая состоит из четырех основных частей: UI Layer, Adapter Layer, Data Layer и Helper Layer.

UI Layer - первый слой, который отвечает за обработку пользовательского взаимодействия с графическим интерфейсом. В данном слое содержится набор Activity, включая EventActivity, SettingsActivity, EventInfoActivity, ScannerActivity, ResultScannerActivity, QrActivity, MyEventActivity, EventStatsActivity, NetworkErrorActivity и MainActivity.

1. EventActivity – Activity для демонстрации пользователю информации о выбранном им мероприятии;
2. SettingsActivity – Activity настроек мобильного приложения;
3. EventInfoActivity – вывод полной информации о мероприятии для ответственного;
4. ScannerActivity – Activity сканера QR-кодов мероприятия;
5. ResultScannerActivity – вывод результата сканирования QR-кода;
6. QrActivity – демонстрация увеличенного варианта QR-кода участника мероприятия;
7. MyEventActivity – вывод информации о мероприятии с пройденном регистрацией;
8. MainActivity – основное окно приложения;
9. EventStatsActivity – Activity для отображения статистики мероприятия;
10. NetworkErrorActivity – Activity отображения информации об ошибке приложения в случае отсутствия интернет-соединения.

В классе MainActivity находится меню управления фрагментами, а также объект для отображения фрагментов. Фрагменты — это маленькие кусочки пользовательского интерфейса, которые могут быть заменены друг на друга в рамках одной активности.

Список фрагментов в приложении включает в себя:

1. EventsFragment – фрагмент, содержащий карточки мероприятий, отображаемые в виде списка;
2. PresenceFragment – фрагмент, отображающий список мероприятий, доступных для подтверждения присутствия участников;
3. MyEventsFragment – фрагмент, отображающий список мероприятий, на которые пользователь зарегистрировался для участия;
4. EventsResponsibleFragment – фрагмент, содержащий карточки мероприятий для ответственного за мероприятия;
5. ProfileCleanFragment – фрагмент профиля без авторизации. Он позволяет пользователям авторизоваться в системе;
6. ProfileFragment – фрагмент профиля с авторизацией, отображает информацию об авторизованном пользователе;
7. ModerationFragment – фрагмент, отображающий список мероприятий, доступных для модерации.

Слой адаптеров (Adapter Layer) представлен тремя классами-адаптерами, необходимые для правильного взаимодействия с RecyclerView:

1. EventAdapter – контроллер карточек мероприятий;
2. MyEventAdapter – контроллер списка мероприятий, на которые зарегистрировался пользователь;
3. EventResponsibleAdapter – контроллер списка карточек мероприятий для ответственных за мероприятия.

Слой данных (Data Layer) состоит из четырех классов, каждый из которых отвечает за хранение определенных данных:

1. Participant – содержит данные для таблицы статистики мероприятий;
2. MyEvent – содержит данные о мероприятии, с пройденной регистрацией;
3. Event – содержит данные о мероприятиях;
4. User – содержит данные об авторизованном пользователе.

Слой Helper Layer включает в себя классы, необходимые для правильной работы мобильного приложения:

1. LanguageConfig – позволяет менять язык мобильного приложения;
2. MyContextWrapper – наследует и переопределяет стандартный класс ContextWrapper, чтобы делегировать его вызовы другому контексту;
3. ServerHelper – отвечает за связь приложения с API информационной системы «Рейтинг студента»;
4. SharedPrefs – позволяет работать с чтением и записью необходимой информации в SharedPreferences.

Многослойная архитектура мобильного приложения позволяет более гибко управлять его компонентами и облегчает поддержку приложения в будущем. Каждый слой выполняет свою функцию и обеспечивает правильное взаимодействие компонентов приложения.

Архитектура мобильного приложения представлена на рисунке 2.2.

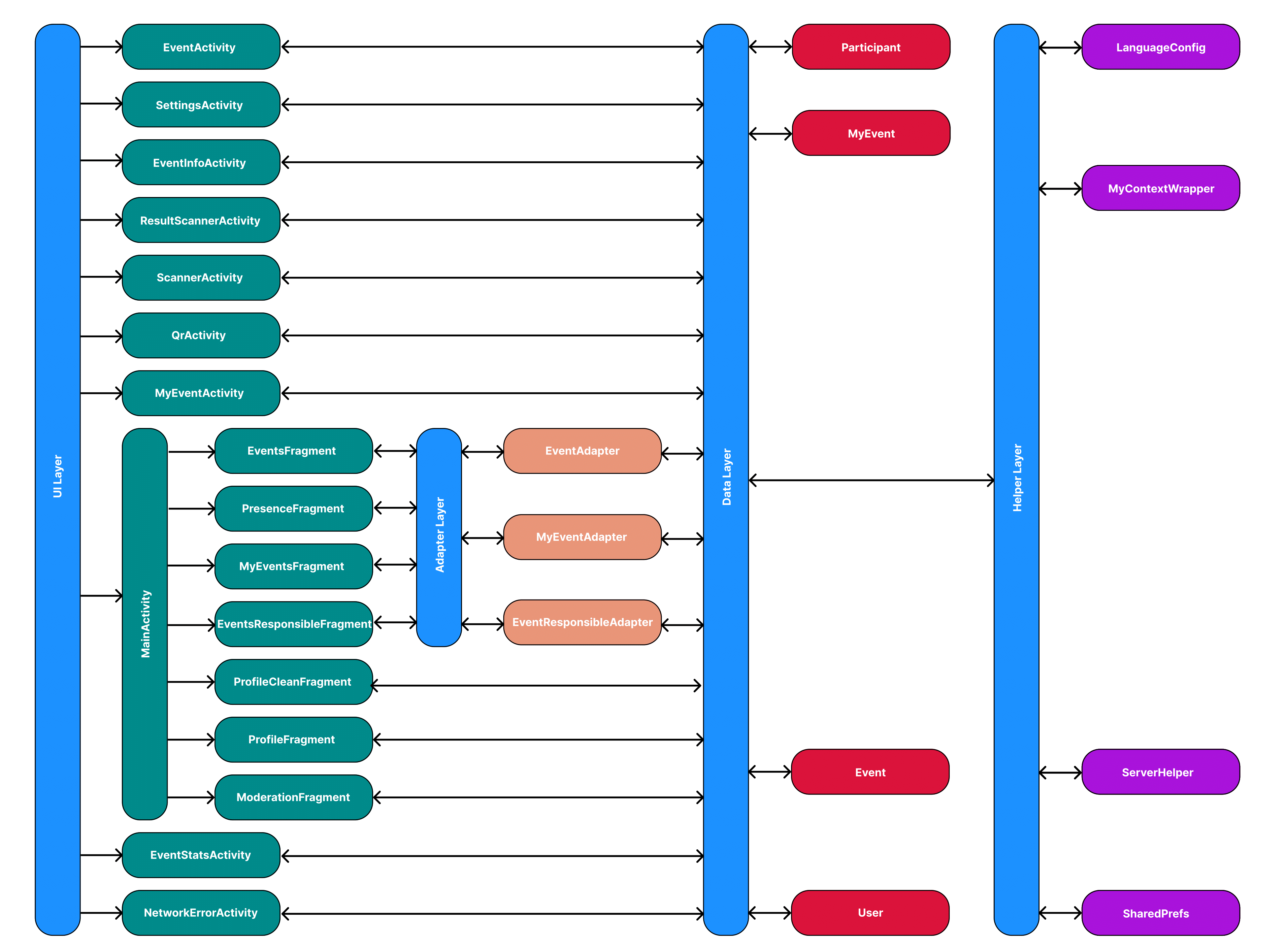


Рисунок 2.2 – Архитектура мобильного приложения

2.3 Проектирование классов данных

Дата классы описывает систему управления мероприятиями и участниками в мобильном приложении.

Сущность Event представляет собой конкретное мероприятие, описывая его id, название, описание, изображения, теги, дату и место проведения, количество зарегистрированных участников, даты начала и окончания регистрации, а также организаторов.

Сущность MyEvent хранит данные о мероприятии, на которое зарегистрировался пользователь, включая id регистрации и id мероприятия, название мероприятия, теги, дату и место проведения, uid пользователя для подтверждения присутствия.

Сущность User описывает авторизованного пользователя, включая его id, ФИО, роль в системе, учебную группу и изображение.

Сущность Participant представляет собой таблицу статистики мероприятия, описывая присутствие зарегистрированных на мероприятии пользователей, их ФИО и учебную группу, фактическое присутствие, роль на мероприятии и подтверждение присутствия ответственным за мероприятие.

На рисунке 2.3 представлена схема классов данных, в таблице 2.1 представлено описание полей.

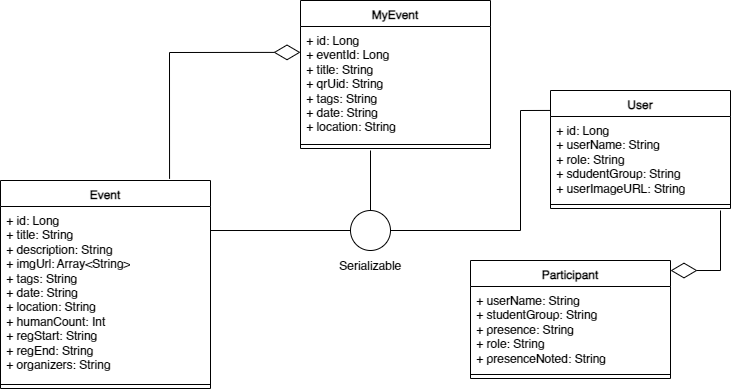


Рисунок 2.3 – Классы данных приложения

Таблица 2.1 – Описание полей Data-классов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Уровень доступа** | **Тип данных** | **Свойство** | **Назначение** |
| Event | | | | |
| id | public | Long | val | id мероприятия в си­стеме |

Продолжение таблицы 2.1 – Описание полей Data-классов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Уровень доступа** | **Тип данных** | **Свойство** | **Назначение** |
| title | public | String | val | Название мероприя­тия |
| description | public | String | val | Описание мероприя­тия |
| imgUrl | public | Array<String> | val | Набор ссылок на изоб­ражения |
| tags | public | String | val | Набор тегов меропри­ятия |
| date | public | String | val | Дата и время проведе­ния мероприятия |
| location | public | String | val | Место проведения ме­роприятия |
| humanCount | public | Int | val | Количество зареги­стрированных участ­ников |
| regStart | public | String | val | Дата и время начала регистрации |
| regEnd | public | String | val | Дата и время конца регистрации |
| organizers | public | String | val | Список организаторов мероприятия |
| MyEvent | | | | |
| id | public | Long | val | id зарегистрирован­ного мероприятия |
| eventId | public | Long | val | id мероприятия в си­стеме |
| title | public | String | val | Название мероприя­тия |
| qrUid | public | String | val | Uid регистрации |
| tags | public | String | val | Набор тегов меропри­ятия |
| date | public | String | val | Дата и время проведе­ния мероприятия |
| location | public | String | val | Место проведения ме­роприятия |
| User | | | | |
| id | public | Long | val | id пользователя в си­стеме |
| userName | public | String | val | ФИО пользователя |

Продолжение таблицы 2.1 – Описание полей Data-классов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Уровень доступа** | **Тип данных** | **Свойство** | **Назначение** |
| userImageURL | public | String | val | Ссылка на изображе­ние пользователя |
| studentGroup | public | String | val | Учебная группа |
| role | public | String | val | Роль в системе |
| Participant | | | | |
| eventId | public | Long | val | id мероприятия в си­стеме |
| userName | public | String | val | ФИО пользователя |
| studentGroup | public | String | val | Учебная группа |
| presence | public | String | val | Фактическое присут­ствие |
| role | public | String | val | Роль в системе |
| presenceNoted | public | String | val | Подтвердивший ответ­ственный |

2.4 Описание сценариев использования

Описание сценариев использования является важной частью разработки мобильного приложения, позволяющее понять, как пользователи будут взаимодействовать с ним в различных ситуациях. Описание сценариев использования способствует созданию более интуитивного и функционального приложения и повышает эффективность его разработки.

Ниже представлены диаграммы спецификаций функций мобильного приложения.

2.4.1 Спецификация вариантов использования «Авторизоваться с помощью логина и пароля» и «Авторизация через Кампус»

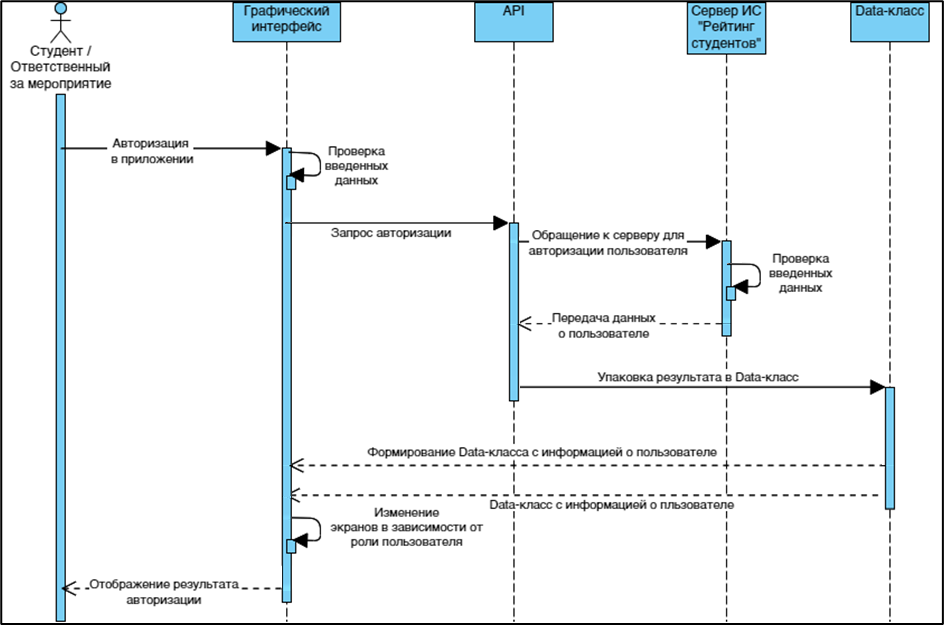


Рисунок 2.4 – «Авторизоваться с помощью логина и пароля» и «Авторизация через Кампус»

2.4.2 Спецификация варианта использования «Регистрация на мероприятие»

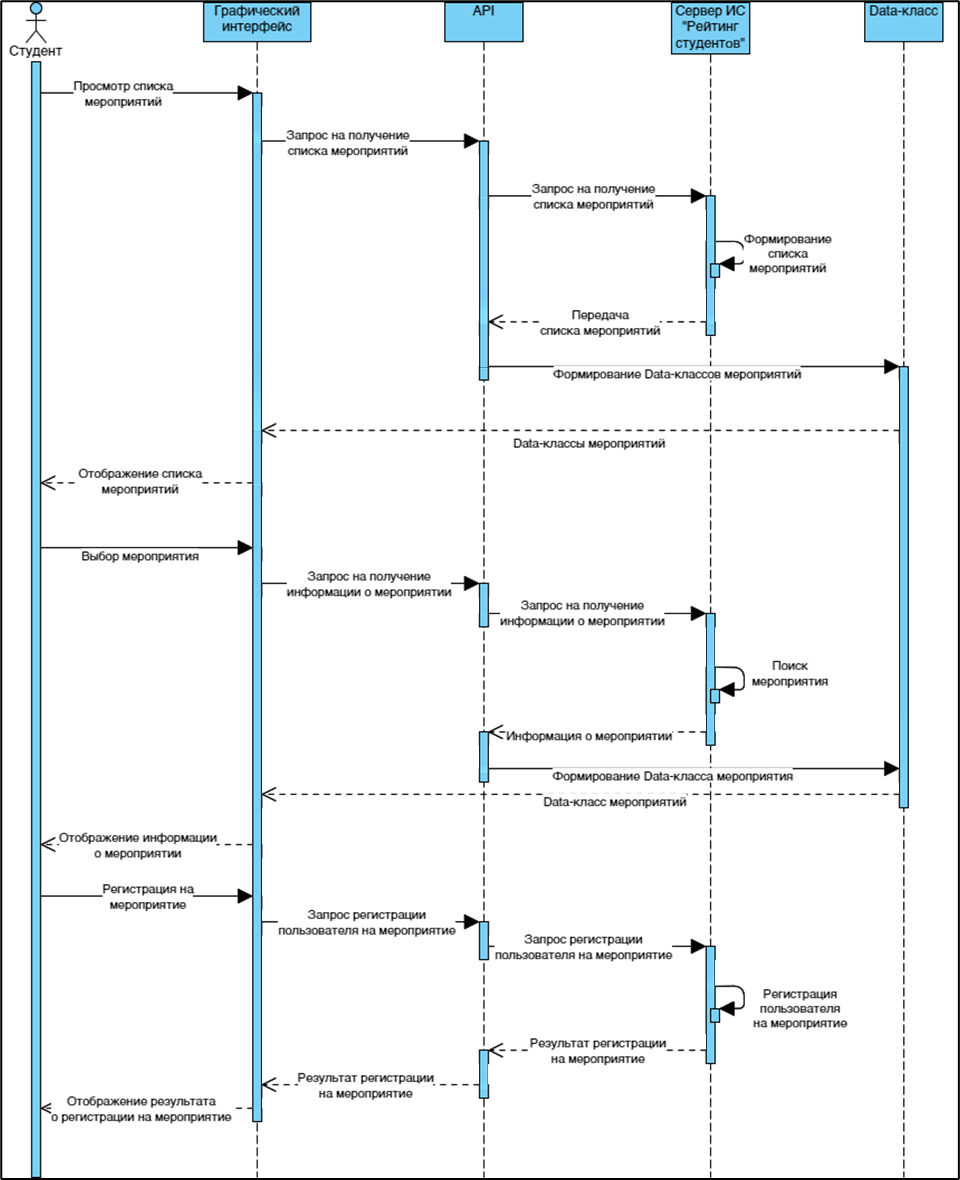


Рисунок 2.5 – «Регистрация на мероприятие»

2.4.3 Спецификация варианта использования «Подтвердить присутствие участника»

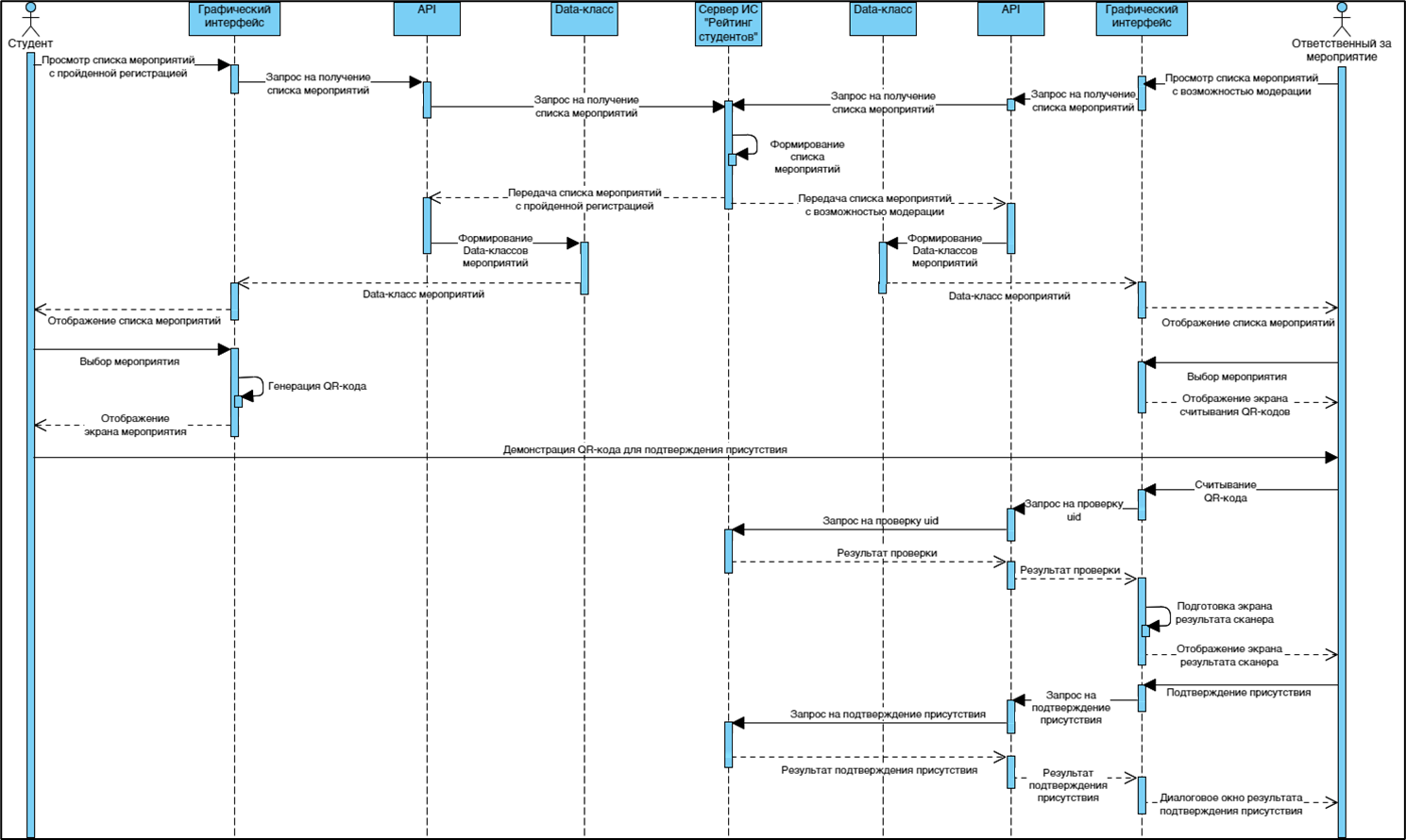


Рисунок 2.6 – «Подтвердить присутствие участника»

2.5 Проектирование графического интерфейса

Для реализации графического интерфейса спроектирован набор экранов.

На рисунке 2.7. представлена диаграмма интерфейсных классов приложения. Зеленым цветом помечены общие для ролей «Студент» и «Ответственный за мероприятие» класса. Белым – классы интерфейса для роли «Студент», красным – для роли «Ответственный за мероприятие». Изображение содержит название экранов и набор функций. Подробное описание и представление элементов представлено на рисунках с 2.8 по 2.22.

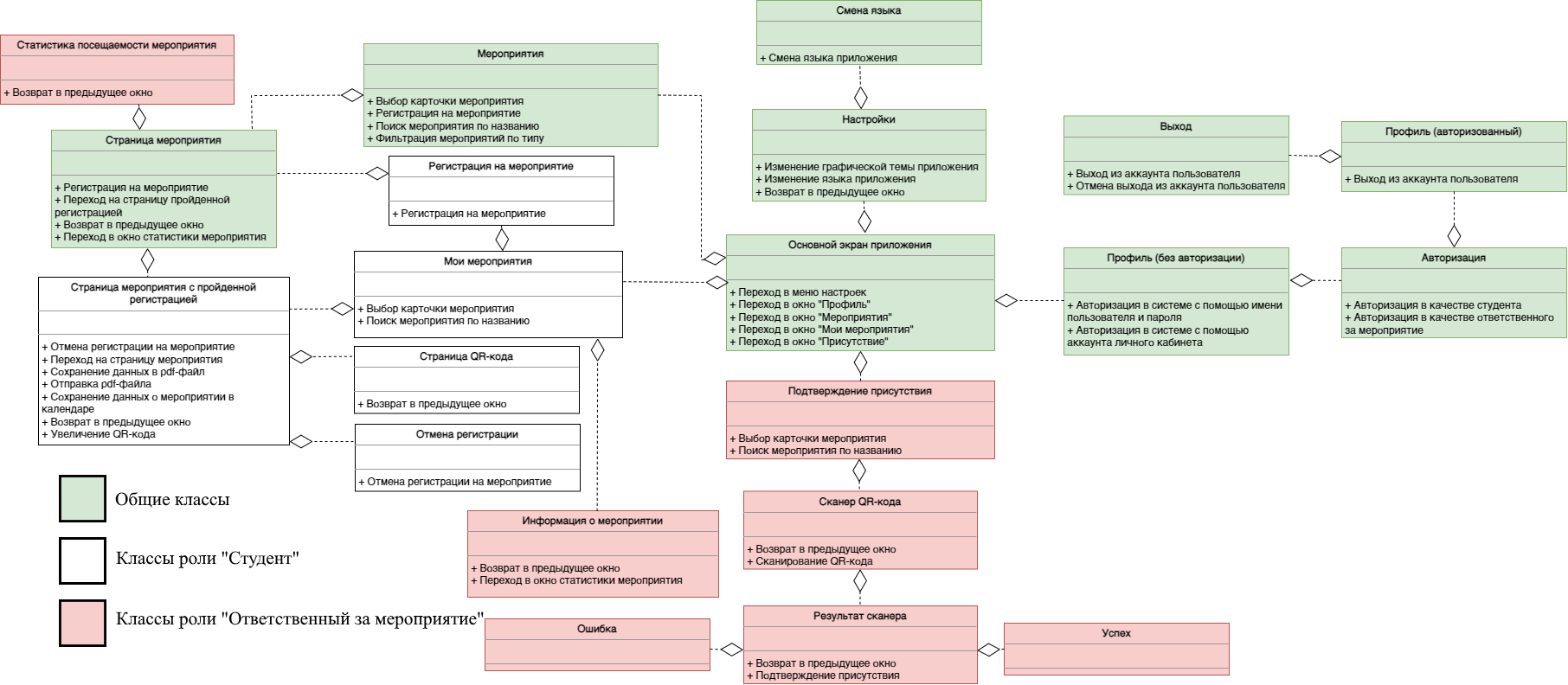


Рисунок 2.7 – Диаграмма интерфейсных классов приложения

2.5.1 Проектирование графического интерфейса для роли «Студент»

**2.5.1.1 Описание основного экрана приложения**

При запуске приложения пользователь должен попадать на основной экран приложения с автоматически загруженным фрагментом «Мероприятия». Экран состоит из следующего набора элементов:

1. **Элемент 1** – верхнее меню приложения. Данный элемент состоит из текстового поля заголовка названия фрагмента и кнопки вызова контекстного меню настроек (**элемент 4**);
2. **Элемент 2** –экран фрагмента. Фрагмент находится внутри основного контейнера экранной формы и расширяет его другими фреймами. При необходимости данный элемент может быть пролистан жестом перемещения пальца снизу вверх. Данная возможность позволяет размещать внутри контейнера фрагмента неограниченное количество карточек мероприятий (**элемент 7**);
3. **Элемент 3** – нижнее меню навигации. Данный компонент отвечает за динамическую загрузку необходимого фрагмента. Выбор необходимого осуществляется за счет трех кнопок, содержащих соответствующую иконку и подпись вызываемого окна;
4. **Элемент 5** – панель фильтрации мероприятий. Данная панель включает три кнопки-варианта применяемых фильтров. Одновременно применен может быть только один фильтр. Для отображения всех возможных мероприятий необходимо активировать кнопку «Любые»;
5. **Элемент 6** – строка поиска мероприятия. Данный компонент отвечает за поиск определенного мероприятия по критерию «Название мероприятия»;
6. **Элемент 7** – карточка мероприятия. Данный элемент состоит из изображения мероприятия (устанавливается нулевой объект из набора графических изображений мероприятия), текстового поля для названия мероприятия, тегов мероприятия, даты и времени проведения мероприятия в формате «dd.mm.yyyy hh:mm», текстового поля для отображения фрагмента описания мероприятия и кнопок «Открыть описание» и «Участвовать» (**элементы 8 и 9 соответственно**);
7. **Элемент 8** – кнопка «Открыть описание». При нажатии на данную кнопку пользователь будет перенаправлен на экран описания мероприятия (см. рисунок 2.8);
8. **Элемент 9** – кнопка «Участвовать». При нажатии на данную кнопку мобильное приложение отобразит диалоговое окно выбора варианта регистрации на мероприятие в качестве участника или зрителя. Данная кнопка находится в неактивном состоянии в двух случаях: при не пройдённой авторизации в систему и при пройденной регистрации на мероприятие.

Макет данного экрана представлен на рисунке 2.8.

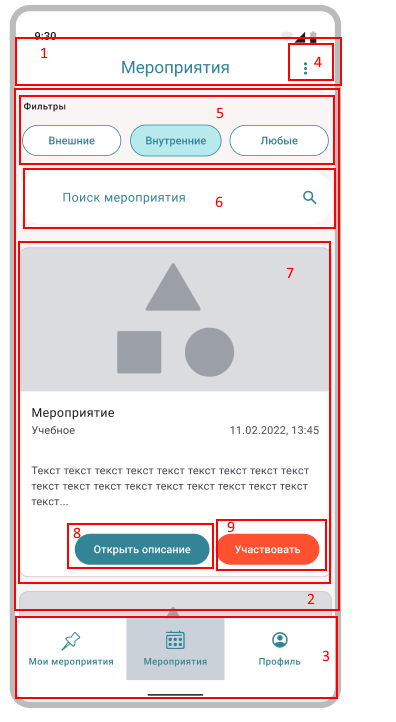


Рисунок 2.8 – Макет основного экрана приложения

**2.5.1.2 Описание экрана «Мои мероприятия»**

Данный экран вызывается нажатием на кнопку «Мои мероприятия» в нижнем меню навигации **элемента 3**, при этом заменяется фрагмент экрана, изменяется надпись текстового поля верхнего меню приложения **элемента 1**. Экран фрагмента состоит из следующего набора элементов:

1. **Элемент 1** – строка поиска мероприятия. Данный компонент отвечает за поиск определенного мероприятия по критерию «Название мероприятия»;
2. **Элемент 2** – список мероприятий с пройденной регистрацией. Список имеет возможность пролистывания, аналогично карточкам мероприятий во фрагменте экрана «Мероприятия». Каждый объект списка состоит из названия мероприятия, набора тегов, даты и времени начала мероприятия и изображения мероприятия (устанавливается нулевой объект из набора графических изображений мероприятия). При нажатии на объект открывается экран мероприятия с пройденной регистрацией (см. рисунок 2.11).

Макет данного экрана представлен на рисунке 2.9.

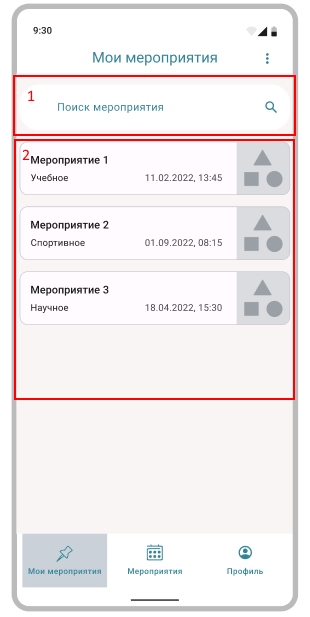


Рисунок 2.9 – Макет экрана «Мои мероприятия»

**2.5.1.3 Описание экрана «Мероприятие»**

Данный экран вызывается при нажатии кнопки «Открыть описание» **элемента 8** карточки мероприятия (см. рисунок 2.8). Экран отображает информацию о доступном для участия мероприятии. Состоит из следующего набора элементов:

1. **Элемент 1** – кнопка возврата на предыдущий экран. Данный элемент интерфейса расположен в верхнем меню приложения. При нажатии на данную кнопку приложение перемещает пользователя в основной экран приложения. Также в верхнем меню приложения присутствует текстовое поле с надписью «Мероприятие»;
2. **Элемент 2** – контейнер информации о мероприятии. Данный элемент имеет возможность пролистывания, что позволяет избежать проблемы некорректного отображения полей при большом объеме информации. Первым компонентом данного контейнера является карусель изображений. Изображения помещены в специальный компонент, поддерживающий возможность как ручной, так и автоматической смены набора изображений мероприятия. Под каруселью расположено текстовое поле для отображения названия мероприятия. Ниже расположены текстовые поля для тегов, присвоенных мероприятию, поле для отображения даты и времени начала мероприятия. Следующим компонентом является описание мероприятия. После размещен компонент для вывода информации о месте проведения мероприятия, поле для вывода информации о количестве участников, прошедших регистрацию на мероприятие, текстовое поле для отображения даты и времени окончания возможности регистрации на мероприятие;
3. **Элемент 3** – кнопка «Участвовать». При нажатии на данную кнопку вызывается диалоговое окно регистрации на мероприятие. В данном окне можно выбрать два варианта регистрации на мероприятие: в качестве участника и в качестве зрителя. Выбор осуществляется с помощью радиокнопок (**элемент 5**). Подтверждение выбора варианта регистрации осуществляется с помощью нажатия на кнопку «ОК» диалогового окна (**элемент 6**). После успешного прохождения процедуры регистрации на мероприятие приложение открывает основной экран приложения (см рисунок 2.5). При отсутствии возможности регистрации на мероприятие (вызов данной страницы без прохождения процедуры авторизации в приложении) кнопка является неактивной;
4. **Элемент 4** – кнопка «Посмотреть QR-код». Данная кнопка заменяет собой **элемент 3** после успешной регистрации на мероприятие. При нажатии на данную кнопку приложение открывает экран мероприятия с пройденной регистрацией (см. рисунок 2.11).

Макет данного экрана представлен на рисунке 2.10.

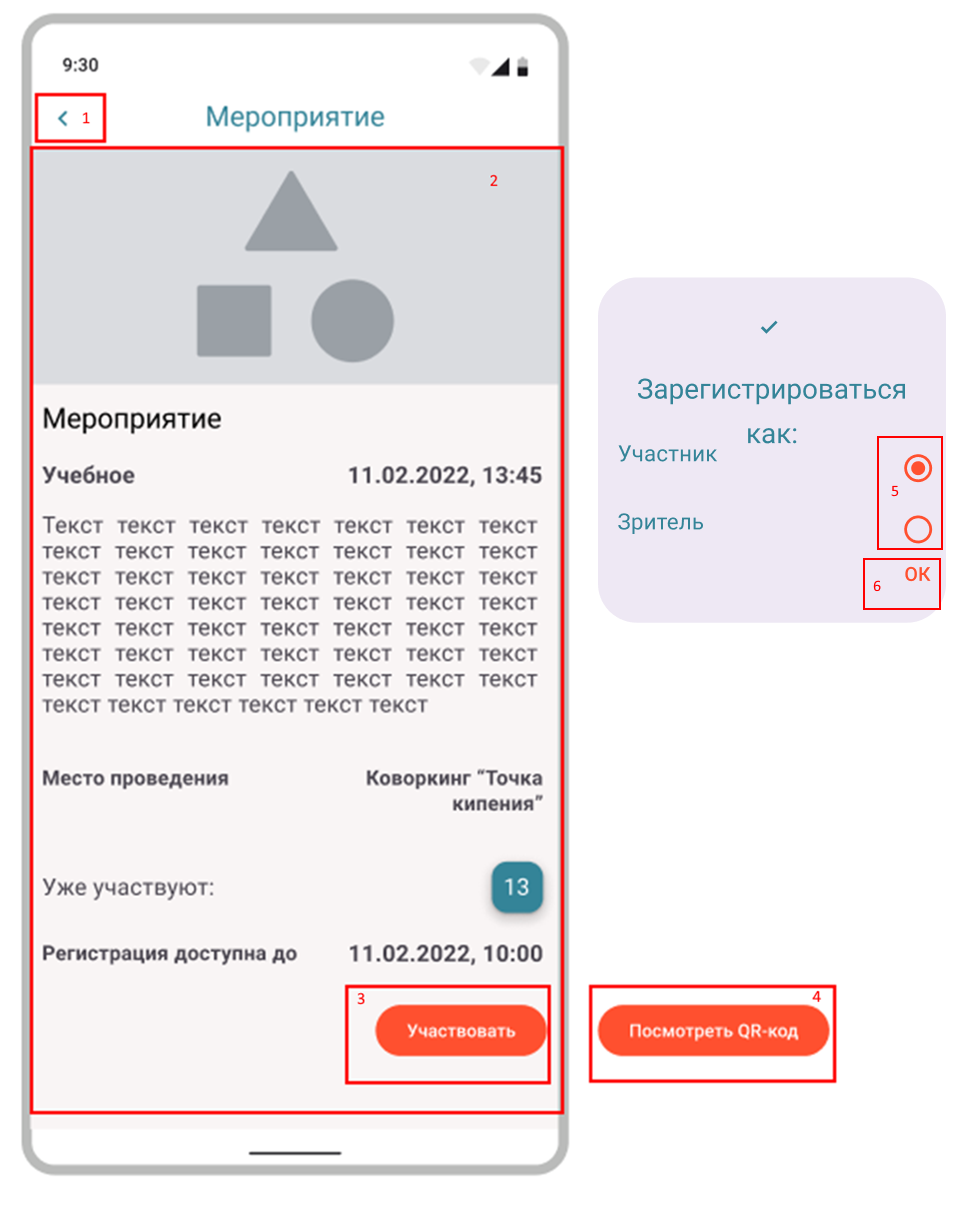


Рисунок 2.10 – Макет экрана «Мероприятие»

**2.5.1.4 Описание экрана мероприятия с пройденной регистрацией**

Данный экран вызывается при нажатии кнопки «Посмотреть QR-код» **элемента 4** экрана «Мероприятие» (см. рисунок 2.10) или выбора компонента из списка **элемента 2** экрана «Мои мероприятия» (см. рисунок 2.9). Экран отображает информацию о мероприятии с пройденной регистрацией. Состоит из следующего набора элементов:

1. **Элемент 1** – кнопка возврата на предыдущий экран. Данный элемент интерфейса расположен в верхнем меню приложения. При нажатии на данную кнопку приложение перемещает пользователя в основной экран приложения. Также в верхнем меню приложения присутствует текстовое поле с надписью «Мероприятие»;
2. **Элемент 2** – изображение QR-код для подтверждения фактического присутствия студента на мероприятии. Изображение является интерактивным. При нажатии на него открывает экран с QR-кодом в увеличенном масштабе (**элемент 7**);
3. **Элемент 3** – кнопка сохранения информации о мероприятии в файл. При нажатии создается pdf-файл и сохраняется в памяти устройства;
4. **Элемент 4** – кнопка отправки файла. При нажатии вызывается системное меню отправки файла с помощью мессенджера и/или электронной почты;
5. **Элемент 5** – кнопка календаря. При нажатии приложение собирает необходимую информацию и передает ее в системное приложение календаря, установленного на устройстве;
6. **Элемент 9** – кнопка «Страница мероприятия». При нажатии на данную кнопку открывается экран «Мероприятие» (см. рисунок 2.10);
7. **Элемент 6** – кнопка «Отменить регистрацию». При нажатии на данную кнопку вызывается диалоговое окно отмены регистрации. В данном окне отображается информация, предупреждающая пользователя о том, что регистрация на мероприятие будет отменена;
8. **Элемент 8** – кнопки «Да» и «Нет» диалогового окна. При нажатии на кнопку «Да» регистрация пользователя будет отменена на активное мероприятие, приложение откроет основной экран приложения (см. рисунок 2.8). При нажатии на кнопку «Нет» диалоговое окно отмены регистрации закроется, экран мероприятия с пройденной регистрацией останется активным.

Макет данного экрана представлен на рисунке 2.11.

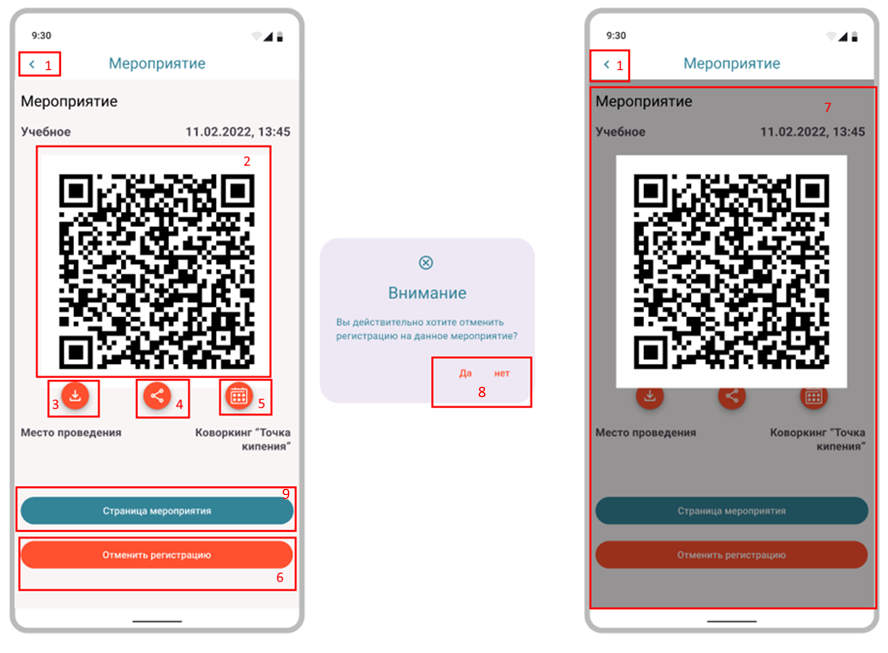


Рисунок 2.11 – Макет экрана мероприятия с пройденной регистрацией

2.5.2 Проектирование графического интерфейса для роли «Ответственный за мероприятие»

**2.5.2.1 Описание основного экрана приложения**

При запуске приложения пользователь должен попадать на основной экран приложения с автоматически загруженным фрагментом «Мероприятия». Набор элементов основного экрана практически идентичен варианту экрана для роли «Студент». Отличаются следующие элементы:

1. **Элемент 1** – карточка мероприятия. Данный элемент состоит из изображения мероприятия (устанавливается нулевой объект из набора графических изображений мероприятия), текстового поля для названия мероприятия, тегов мероприятия, даты и времени проведения мероприятия в формате «dd.mm.yyyy hh:mm», текстового поля для отображения фрагмента описания мероприятия и кнопки «Открыть описание». Отличие заключается в отсутствии кнопки «Участвовать» для данной роли;
2. **Элемент 2** – нижнее меню навигации. Данный компонент отвечает за динамическую загрузку необходимого фрагмента. Выбор необходимого осуществляется за счет четырех кнопок, содержащих соответствующую иконку и подпись вызываемого окна.

Макет данного экрана представлен на рисунке 2.12.

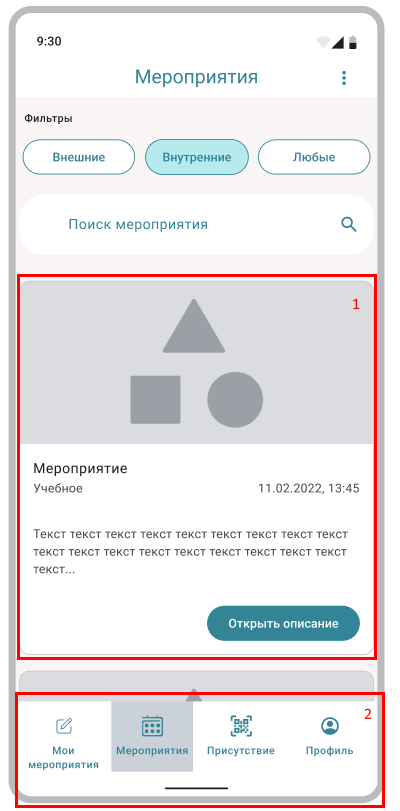


Рисунок 2.12 – Макет основного экрана приложения

**2.5.2.2 Описание экрана «Мероприятие»**

Данный экран вызывается при нажатии кнопки «Открыть описание» кнопки «Открыть описание» карточки мероприятия (см. рисунок 2.12). Экран отображает информацию о доступном для участия мероприятии. Наполнение данного экрана аналогично экрану «Мероприятие» для роли «Студент». Единственное отличие заключается в отсутствии кнопки «Участвовать».

Макет данного экрана представлен на рисунке 2.13.

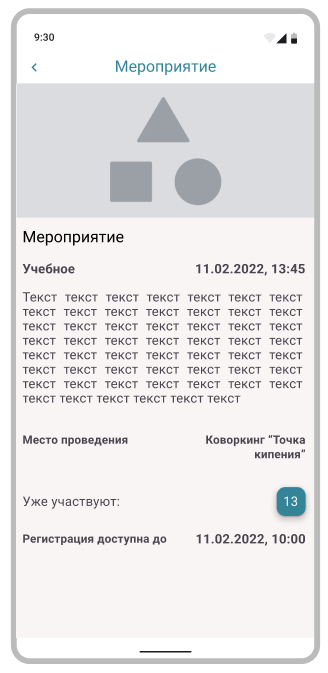


Рисунок 2.13 – Макет экрана «Мероприятие»

**2.5.2.3 Описание экрана «Подтверждение присутствия»**

Данный экран вызывается нажатием на кнопку «Присутствие» в нижнем меню навигации **элемента 2**, при этом заменяется фрагмент экрана, изменяется надпись текстового поля верхнего меню приложения. Экран фрагмента состоит из следующего набора элементов:

1. **Элемент 1** – строка поиска мероприятия. Данный компонент отвечает за поиск определенного мероприятия по критерию «Название мероприятия»;
2. **Элемент 2** – список мероприятий с возможностью модерирования. Список имеет возможность пролистывания, аналогично карточкам мероприятий во фрагменте экрана «Мероприятия». Каждый объект списка состоит из названия мероприятия, набора тегов, даты и времени начала мероприятия и изображения мероприятия (устанавливается нулевой объект из набора графических изображений мероприятия). При нажатии на объект открывается экран сканера QR-кодов (см. рисунок 2.15).

Макет данного экрана представлен на рисунке 2.14.

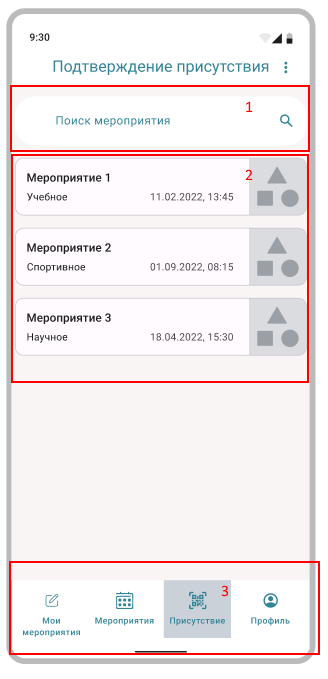


Рисунок 2.14 – Макет экрана «Подтверждение присутствия»

**2.5.2.4 Описание экрана сканера QR-кодов**

Данный экран вызывается при выборе объекта списка в экране «Подтверждение присутствия» **элемента 2** (см. рисунок 2.14). Экран отображает информацию о мероприятии с пройденной регистрацией. Состоит из следующего набора элементов:

1. **Элемент 1** – кнопка возврата на предыдущий экран. Данный элемент интерфейса расположен в верхнем меню приложения. При нажатии на данную кнопку приложение перемещает пользователя в основной экран приложения. Также в верхнем меню приложения присутствует текстовое поле с надписью названия выбранного мероприятия;
2. **Элемент 2** – текстовое поле для отображения краткой информации об использовании сканера QR-кодов;
3. **Элемент 3** – сканер QR-кодов. Представляет из себя рамку для позиционирования QR-кода и отображение изображения, поступающего из камеры мобильного устройства. При успешном сканировании кода приложение открывает экран результата сканирования. В данном окне присутствует **элемент 1**, открывающий экран сканера, **элемент 4**, отображающий изображение пользователя, установленное в профиле, **элемент 5**, представляющий из себя текстовое поле для отображения ФИО пользователя и **элемент 6** – кнопку «Подтвердить присутствие». При нажатии на данную кнопку и выполнении подтверждения присутствия вызывается **элемент 7** - диалоговое окно ошибки, в случае неудачного подтверждения или диалоговое окно успеха, в случае успешного подтверждения присутствия.

Макет данного экрана представлен на рисунке 2.15.

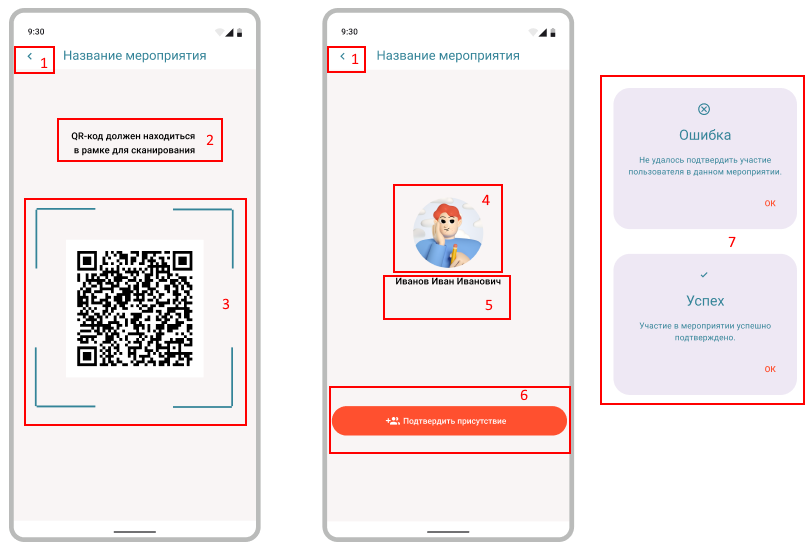


Рисунок 2.15 – Макет экрана сканера QR-кода

**2.5.2.5 Описание экрана «Мои мероприятия»**

Данный экран вызывается нажатием на кнопку «Мои мероприятия» в нижнем меню навигации элемента 2, при этом заменяется фрагмент экрана, изменяется надпись текстового поля верхнего меню приложения. Экран фрагмента состоит из следующего набора элементов:

1. **Элемент 1** – строка поиска мероприятия. Данный компонент отвечает за поиск определенного мероприятия по критерию «Название мероприятия»;
2. **Элемент 2** – список мероприятий с возможностью просмотра подробной информации о мероприятии. Список имеет возможность пролистывания, аналогично карточкам мероприятий во фрагменте экрана «Мероприятия». Каждый объект списка состоит из названия мероприятия, набора тегов, даты и времени начала мероприятия и изображения мероприятия (устанавливается нулевой объект из набора графических изображений мероприятия). При нажатии на объект открывается экран подробной информации о мероприятии (см. рисунок 2.17).

Макет данного экрана представлен на рисунке 2.16.

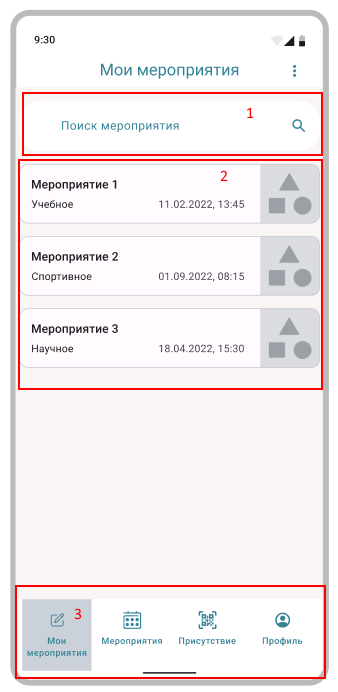


Рисунок 2.16 – Макет экрана «Мои мероприятия»

**2.5.2.6 Описание экрана «Информация о мероприятии»**

Данный экран вызывается при выборе объекта списка в экране «Мои мероприятия». Экран отображает подробную информацию о мероприятии. Состоит из следующего набора элементов:

1. **Элемент 1** – кнопка возврата на предыдущий экран. Данный элемент интерфейса расположен в верхнем меню приложения. При нажатии на данную кнопку приложение перемещает пользователя в основной экран приложения. Также в верхнем меню приложения присутствует текстовое поле с надписью «Информация о мероприятии»;
2. **Элемент 2** – контейнер с информацией о мероприятии. Включает в себя карусель изображений мероприятия, текстовое поле названия мероприятия, текстовое поле тегов мероприятия, дату и время проведения мероприятия, описание мероприятия, место проведения, текстовые поля начала и конца регистрации на мероприятие, а также перечень организаторов мероприятия. Данный контейнер имеет возможность пролистывания, решающий проблему размещения большого объема информации;
3. **Элемент 3** – кнопка «Статистика мероприятия». При нажатии на данный элемент открывается экран статистики мероприятия (см. рисунок 2.18).

Макет данного экрана представлен на рисунке 2.17.

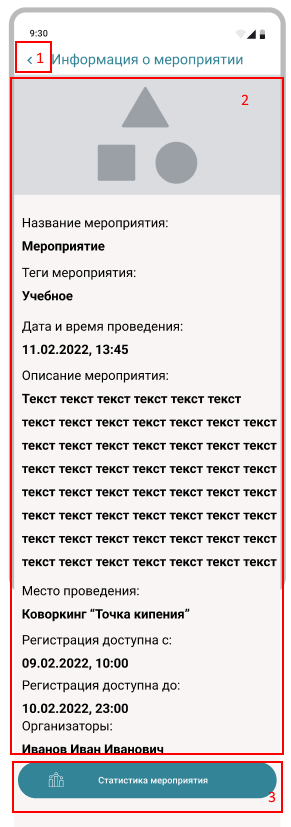


Рисунок 2.17 – Макет экрана «Информация о мероприятии»

**2.5.2.7 Описание экрана статистики мероприятия**

Данный экран вызывается при нажатии на кнопку «Статистика мероприятия» **элемента 3** экрана «Информация о мероприятии» (см. рисунок 2.17). Экран отображает статистическую информацию о мероприятии. Состоит из следующего набора элементов:

1. **Элемент 1** – кнопка возврата на предыдущий экран. Данный элемент интерфейса расположен в верхнем меню приложения. При нажатии на данную кнопку приложение перемещает пользователя в экран информации о мероприятии. Также в верхнем меню приложения присутствует текстовое поле с надписью названия мероприятия;
2. **Элемент 2** – таблица статистики. Данный элемент отображает статистику присутствия студентов на мероприятии. Таблица состоит из шести колонок: «№» – отображение порядкового номера студента в таблице, «ФИО Студента» – отображение ФИО студента, «Группа» – отображение студенческой группы, «Роль» – отображение роли участника мероприятия (зритель, участник), «Присутствие отметил» – отображение ФИО ответственного за мероприятие, отметившего присутствие студента, «Присутствует» – статус фактического присутствия на мероприятии («Да» или «Нет»)

Макет данного экрана представлен на рисунке 2.18.



Рисунок 2.18 – Макет экрана статистики мероприятия

2.5.3 Проектирование графического интерфейса непривязанных к конкретной роли

**2.5.3.1 Описание экрана «Профиль»**

Данный экран вызывается при выборе объекта списка в экране «Профиль» (см. рисунок 2.12 – для ответственного, см. рисунок 2.8 – для студента). Данный экран состоит из следующего набора элементов:

1. **Элемент 1** – текстовое поле для отображения информации о необходимости авторизации для использования полного функционала мобильного приложения;
2. **Элемент 2** – поля ввода данных от аккаунта для авторизации в приложении;
3. **Элементы 3 и 4** – кнопки авторизации. При нажатии на данные кнопки вызывается диалоговое окно авторизации. В нем доступна возможность выбора варианта авторизации (в качестве студента или ответственного за мероприятие) с помощью радиокнопок (**элемент 5**) и подтверждения выбранного варианта с помощью кнопки «ОК» (**элемент 6**). Обоснование наличия двух кнопок авторизации заключается в том, что кнопка «Войти» отвечает за авторизацию с помощью данных от аккаунта информационной системы «Рейтинг студентов», кнопка «Войти через личный кабинет» отвечает за авторизацию с помощью аккаунта «Кампуса».

Макет данного экрана представлен на рисунке 2.19.

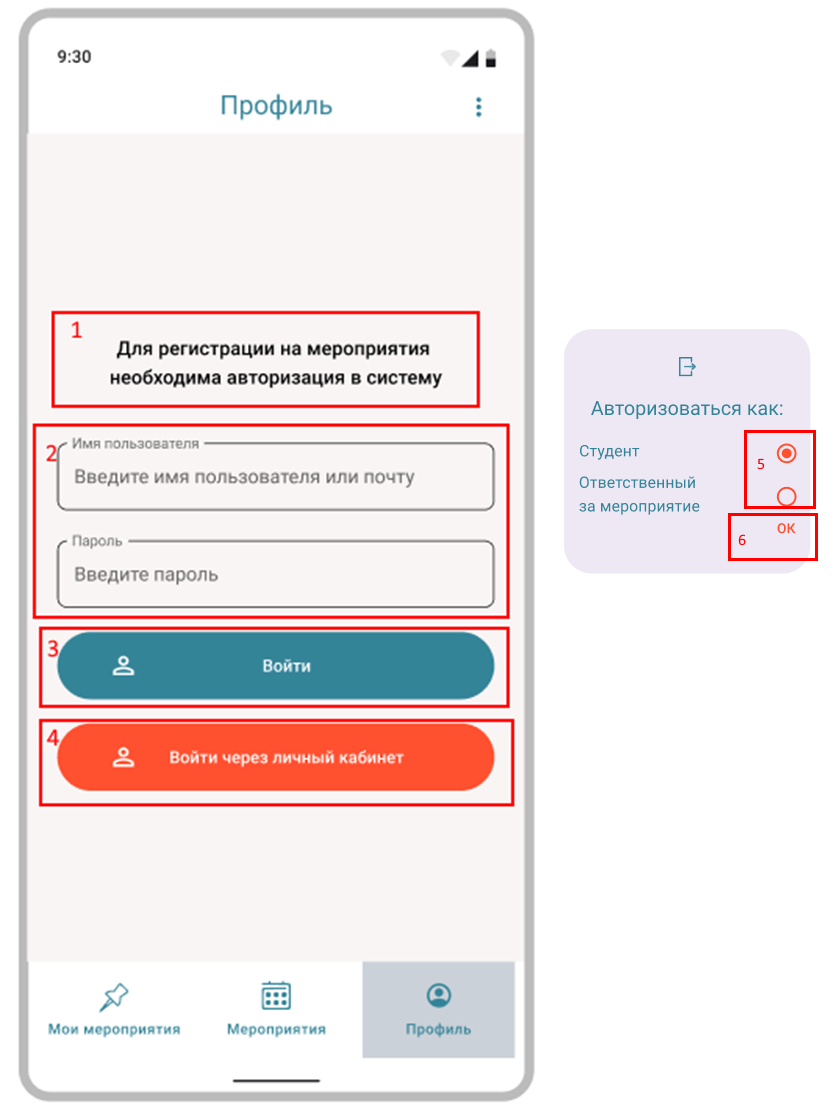


Рисунок 2.19 – Макет экрана «Профиль»

**2.5.3.2 Описание экрана «Профиль» (авторизованный)**

Данный экран заменяет собой экран «Профиль» без авторизации (см. рисунок 2.19). Экран состоит из следующего набора элементов:

1. **Элемент 1** – изображение профиля пользователя;
2. **Элемент 2** – текстовое поле отображение информации об авторизованном пользователе;
3. **Элемент 3** – кнопка «Выйти из аккаунта». При нажатии на данную кнопку вызывается диалоговое окно выхода;
4. **Элемент 4** – кнопки «Да» и «Нет» диалогового окна. В случае нажатия на кнопку «Да» система осуществит выход из приложения, откроет основной экран приложения. В случае нажатия на кнопку «Нет» диалоговое окно закроется, пользователь останется авторизованным.

Макет данного экрана представлен на рисунке 2.20.

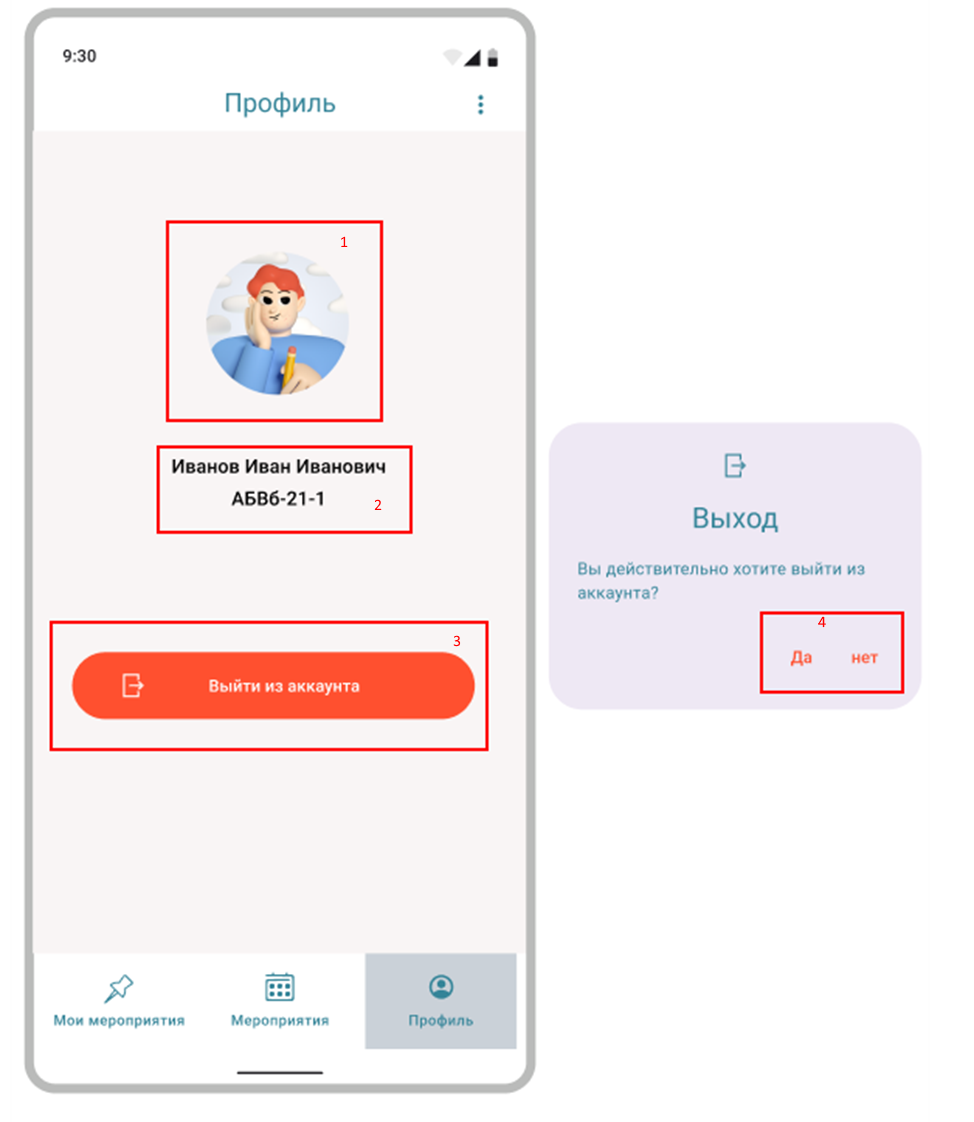


Рисунок 2.20 – Макет экрана «Профиль» (авторизованный)

**2.5.3.3 Описание экрана «Настройки»**

Данный экран вызывается при вызове контекстного меню настроек верхнего меню приложения (**элемент 1**). Экран состоит из следующего набора элементов:

1. **Элемент 2** – кнопка возврата на предыдущий экран. Данный элемент интерфейса расположен в верхнем меню приложения. При нажатии на данную кнопку приложение перемещает пользователя в основной экран приложения. Также в верхнем меню приложения присутствует текстовое поле с надписью «Настройки»;
2. **Элемент 3** – Раздел настройки темы приложения. Включает в себя текстовое поле подписи настройки и переключатель темы приложения. Темная тема приложения представлена на рисунке 2.22;
3. **Элемент 4** – кнопка «Изменить язык приложения». При нажатии на данную кнопку вызывается диалоговое окно выбора языка приложения с помощью радиокнопок (**элемент 5**).

Макет данного экрана представлен на рисунке 2.21.

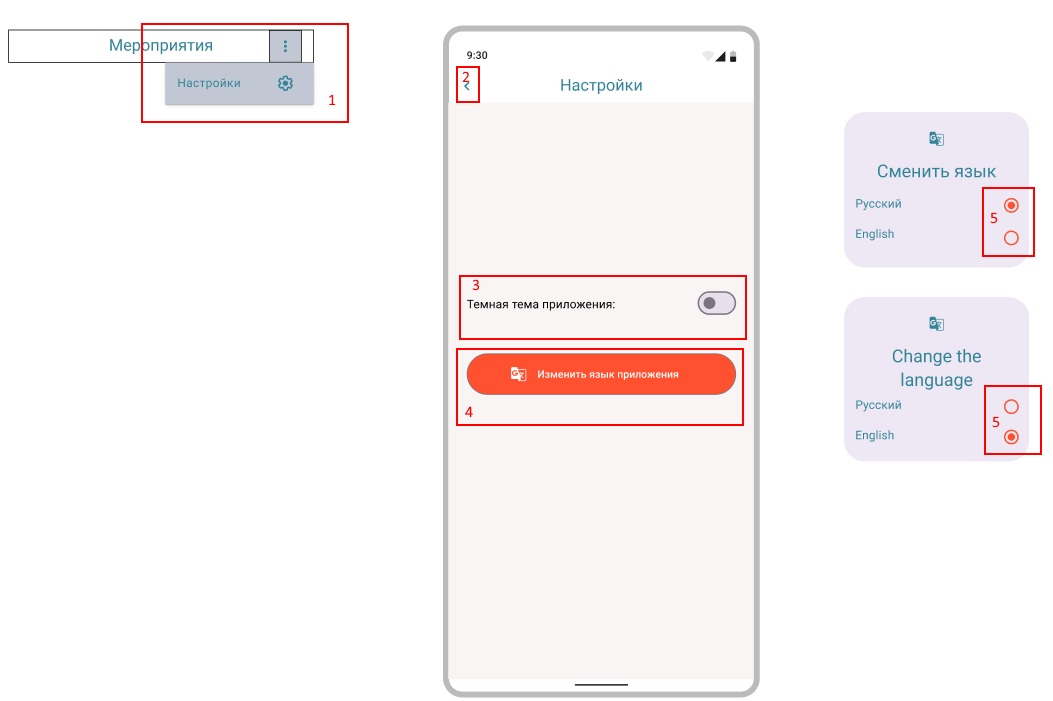


Рисунок 2.21 – Макет экрана «Настройки»

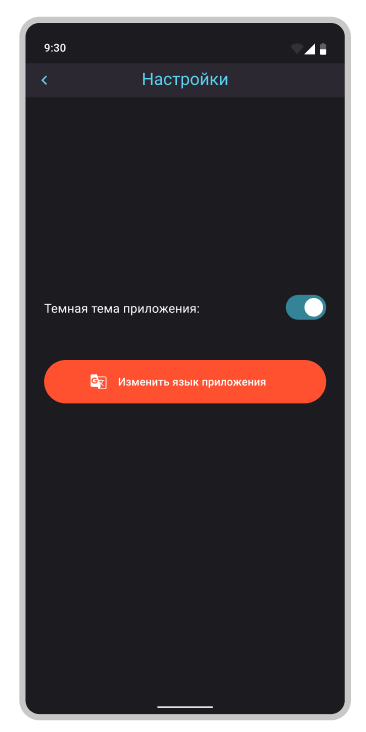


Рисунок 2.22 – Макет экрана «Настройки» (темная тема)

3 Реализация системы

3.1 Реализация функции «Сформировать QR-код»

Мобильное приложение получает регистрационный uid пользователя для мероприятия, генерирует на его основе QR-код и помещает его в контейнер для изображения (см. рисунок 3.1). Функция реализуется в классе MyEventActivity.

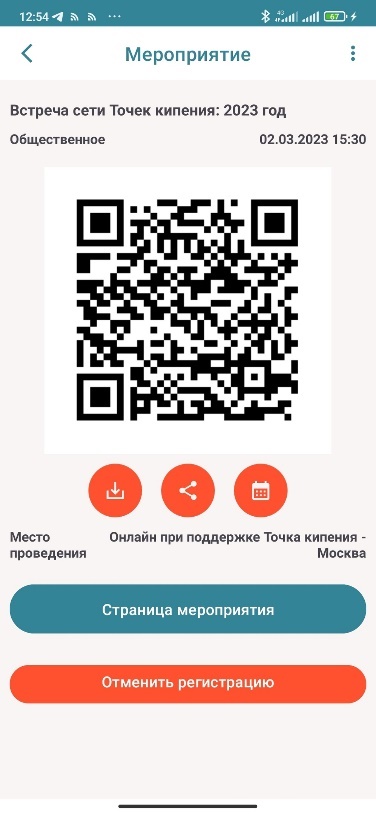


Рисунок 3.1 – Генерация QR-кода

**Фрагмент кода для реализации функции**

private fun generateQRCode(uid: String){

val qrCode = QRGEncoder(uid, null, QRGContents.Type.TEXT, 300)

qrCode.colorBlack = Color.WHITE

qrCode.colorWhite = Color.BLACK

try {

val bitMap = qrCode.bitmap

qrImg.setImageBitmap(bitMap)

}

catch (e: WriterException){

println(e)

}

}

3.2 Реализация функции «Сохранить QR-код в pdf-файл»

Мобильное приложение сохраняет необходимую информацию о мероприятии в виде pdf-файла (см. рисунок 3.2). Функция выполняется в классе MyEventActivity, запуск осуществляется с помощью кнопки сохранения.



Рисунок 3.2 – Pdf-файл информации о мероприятии

**Фрагмент кода для реализации функции**

private fun savePDF() {

try {

val FONT = "/res/font/timesnewromanpsmt.ttf"

val bf = BaseFont.createFont(FONT.toString(), BaseFont.IDENTITY\_H, BaseFont.EMBEDDED)

val fontBold = Font(bf, 18f, Font.BOLD)

val fontNormal = Font(bf, 14f, Font.NORMAL)

val doc = com.itextpdf.text.Document()

val fileName = SimpleDateFormat("yyyyMMdd\_HHmmss", Locale.getDefault()).format(System.currentTimeMillis())

val filePath = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY\_DOWNLOADS).toString() + "/" + fileName + ".pdf"

pathToShare = filePath

PdfWriter.getInstance(doc, FileOutputStream(filePath))

doc.open()

val bitmap =

(qrImg.getDrawable().getCurrent() as BitmapDrawable).bitmap

val bmp = Bitmap.createScaledBitmap(bitmap, 300, 300, false);

val stream = ByteArrayOutputStream()

bmp.compress(Bitmap.CompressFormat.JPEG, 100, stream)

val image = Image.getInstance(stream.toByteArray())

image.alignment = ALIGN\_CENTER

val title = "${event.title}\n".toString().trim()

val titleParagraph = Paragraph(title, fontBold)

titleParagraph.alignment = ALIGN\_CENTER

val footer = "Место проведения ${event.location}\n" +

"Дата начала ${event.date}".toString().trim()

val footerParagraph = Paragraph(footer, fontNormal)

footerParagraph.alignment = ALIGN\_CENTER

doc.addAuthor("Рейтинг стипендии")

doc.add(titleParagraph)

doc.add((image))

doc.add(footerParagraph)

doc.close()

Toast.makeText(this, "$fileName.pdf $filePath", Toast.LENGTH\_SHORT).show()

}

catch (e: Exception){

println(e)

}

}

3.3 Реализация функции «Отправить pdf-файл в мессенджер»

Мобильное приложение подготавливает pdf-файл с информацией о мероприятии, сохраняет его в памяти устройства и вызывает Intent отправки файлов (см. рисунок 3.3). Функция реализуется в классе MyEventActivity.

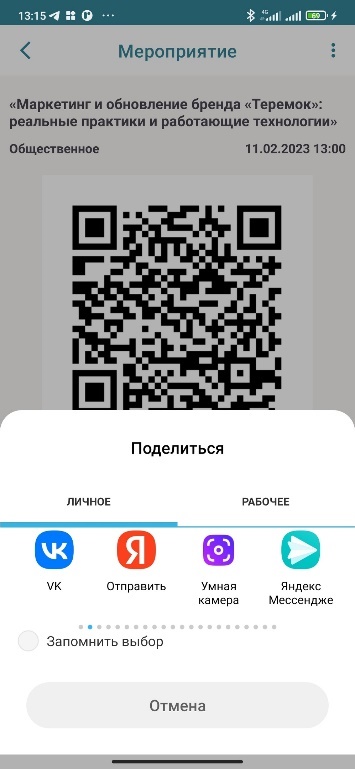


Рисунок 3.3 – Отправка pdf-файла

**Фрагмент кода для реализации функции**

private fun startFileShareIntent() {

val shareIntent = Intent(Intent.ACTION\_SEND).apply {

type = "\*/\*"

flags = Intent.FLAG\_GRANT\_READ\_URI\_PERMISSION

putExtra(

Intent.EXTRA\_SUBJECT,

""

)

putExtra(

Intent.EXTRA\_TEXT,

""

)

val fileURI = FileProvider.getUriForFile(

this@MyEventActivity!!, this@MyEventActivity!!.packageName + ".provider",

File(pathToShare)

)

putExtra(Intent.EXTRA\_STREAM, fileURI)

}

startActivity(shareIntent)

}

3.4 Реализация функции «Добавить мероприятие в календарь смартфона»

Приложение подготавливает данные о мероприятии и вызывает Intent выбора приложения-календаря (см. рисунок 3.4.). Выбранное приложение получает набор данных и формирует новое событие в день проведения мероприятия (см. рисунок 3.5). Функция реализуется в классе MyEventActivity.

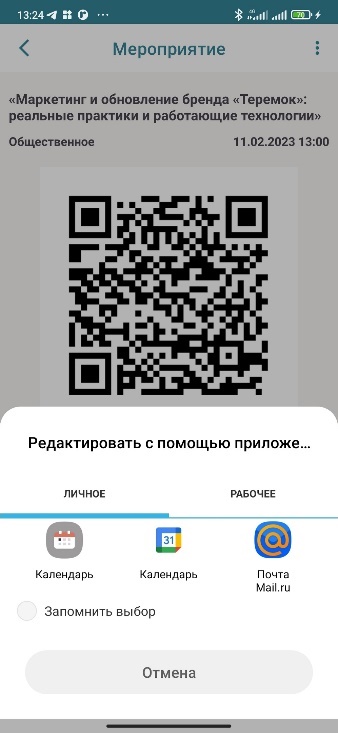


Рисунок 3.4 – Вызов приложения-календаря устройства

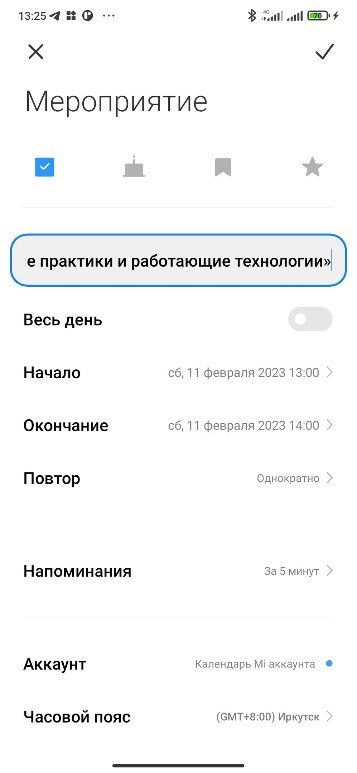


Рисунок 3.5 – Создание нового события в календаре

**Фрагмент кода для реализации функции**

calendarButton.setOnClickListener {

val mIntent = Intent(Intent.ACTION\_EDIT)

mIntent.type = "vnd.android.cursor.item/event"

mIntent.putExtra("beginTime", mStartTime.time)

mIntent.putExtra("time", true)

mIntent.putExtra("rule", "FREQ=YEARLY")

mIntent.putExtra("title", "${event.title}")

startActivity(mIntent)

}

3.5 Реализация функции «Отменить регистрацию на мероприятие»

При нажатии на соответствующую кнопку на странице мероприятия вызывается диалог-предупреждение об отмене регистрации на мероприятии (см. рисунок 3.6).

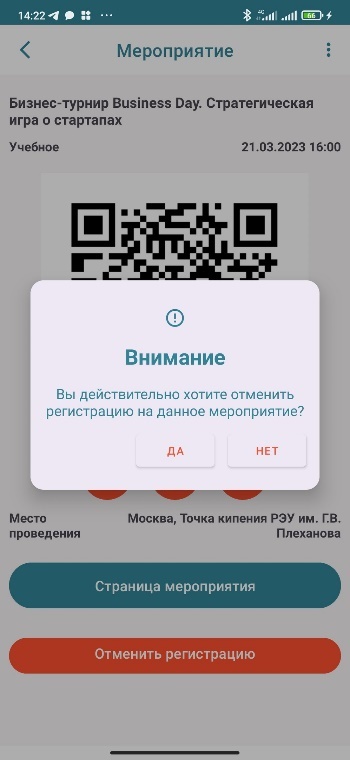


Рисунок 3.6 – Отмена регистрации на мероприятие

**Фрагмент кода для реализации функции**

private fun showDialogUnreg() {

val builder = MaterialAlertDialogBuilder(this, R.style.MaterialAlertDialog\_Rounded)

val dialogView = LayoutInflater.from(this).inflate(R.layout.dialog\_warning, null)

builder.setView(dialogView)

val yes\_button = dialogView.findViewById<MaterialButton>(R.id.yes\_button)

val no\_button = dialogView.findViewById<MaterialButton>(R.id.no\_button)

val dialog = builder.create()

yes\_button.setOnClickListener {

serverHelper.cancellationRegistration()

dialog.dismiss();

}

no\_button.setOnClickListener {

dialog.dismiss();

}

dialog.show()

}

3.6 Реализация функции «Просмотреть мероприятие»

При открытии страницы мероприятия приложение обращается к серверу информационной системы «Рейтинг студента», получает информацию и размещает ее в элементы экрана. Функция получения данных реализована в классе ServerHelper, заполнение данных в EventActivity (см. рисунок 3.7).



Рисунок 3.7 – Страница мероприятия

**Фрагмент кода для реализации функции**

fun getEventInfo(id: String){

val stringRequest = StringRequest(Request.Method.GET, EVENT\_URL + "$id", { //Передача запроса и получение ответа

response -> //Случай удачного результата отклика api

val obj = JSONObject(response) //Получение json файла

try {

var images = obj.optJSONArray("images").getString(0)

val id = obj.optString("id").toString().toLong()

val type = obj.optString("type")

val title = obj.optString("title")

val dateStart = dateParsing(obj.optString("dateStart"))

val description = obj.optString("description")

val location = obj.optString("location")

val humanCount = obj.optInt("count\_people")

val res = images.toString()

event = Event(id, title, description, arrayOf(res), type, dateStart, location, humanCount, null, null, null)

}

catch (ex: java.lang.NullPointerException){

println(ex)

}

}, {

error -> //Случай неудачного результата отклика api

println("resp error ${error}")

})

queue.add(stringRequest) //Добавление запроса в очередь

}

3.7 Реализация функции смены языка приложения

Приложение поддерживает локализацию на два языка: русский и английский. Смена языка осуществляется путем выбора посредством радиокнопок диалогового окна приложения. Функция реализована в классе SharedPrefs, вызов диалогового окна в классе SettingsActivity (см. рисунок 3.8).

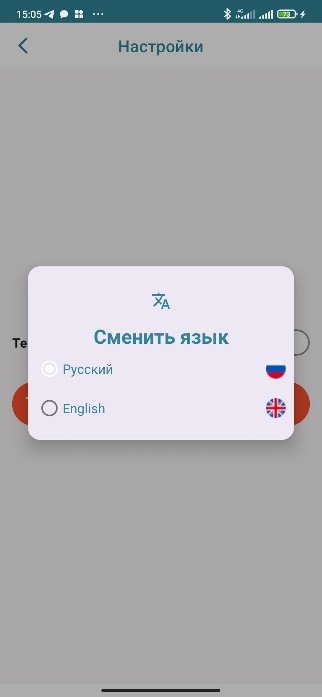


Рисунок 3.8 – Диалоговое окно смены языка

**Фрагмент кода для реализации функции**

val preferences = context.getSharedPreferences(PREFERENCE\_NAME, Context.MODE\_PRIVATE)

fun getLanguageCount(): String{

return preferences.getString(PREFERENCE\_LANGUAGE, "ru")!!

}

fun setLanguageCount(language: String) {

val editor = preferences.edit()

editor.putString(PREFERENCE\_LANGUAGE, language)

editor.apply()

}

3.8 Реализация функции смены графической темы приложения

Смена графической темы приложения осуществляется за счет изменения состояния переключателя (см. рисунок 3.9). Функция реализована в классе SharedPrefs, вызов диалогового окна в классе SettingsActivity.

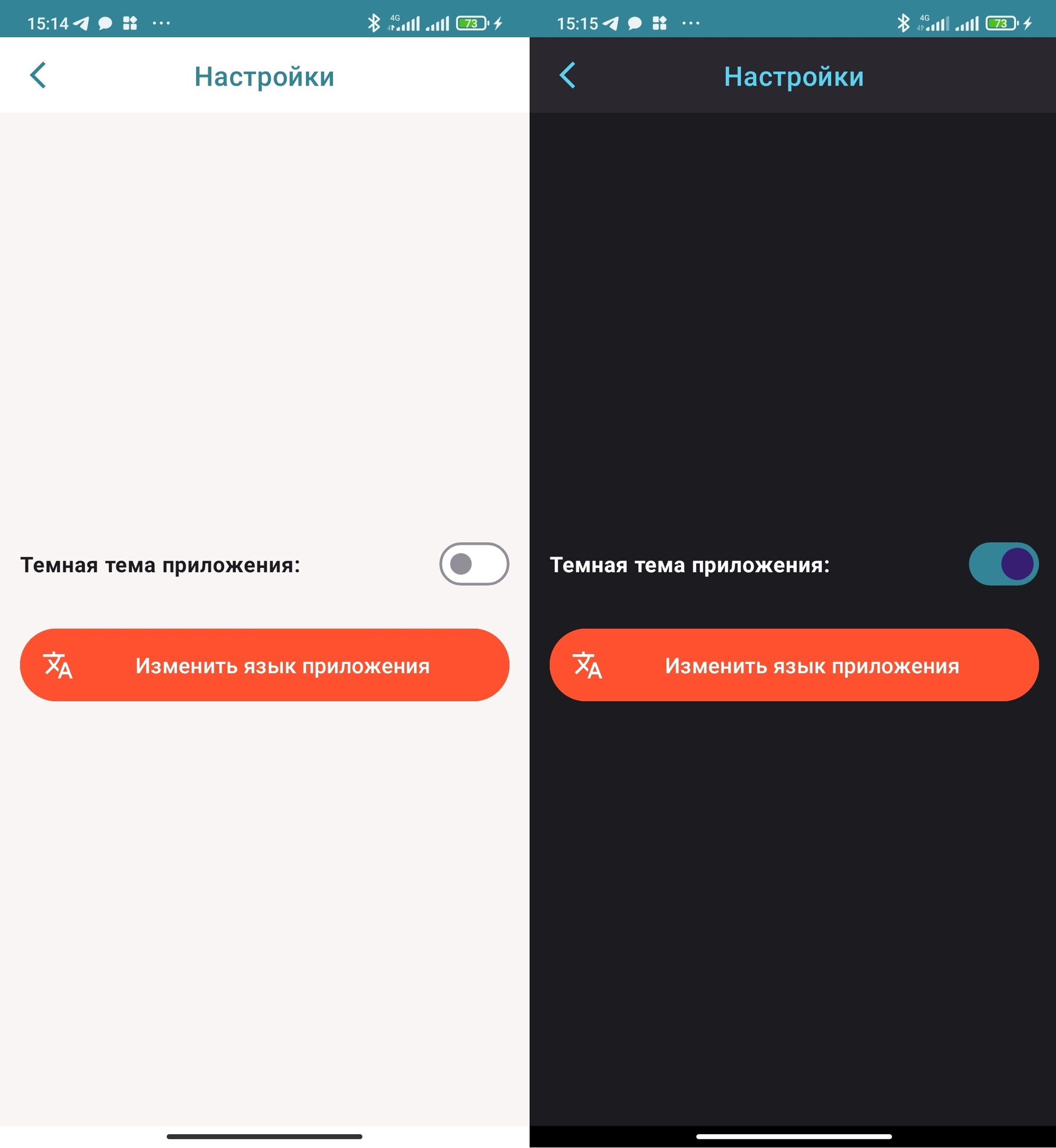


Рисунок 3.9– Смена графической темы

**Фрагмент кода для реализации функции**

switch.setOnCheckedChangeListener { compoundButton, b ->

if(b){

try {

AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_YES);

appPrefs.setThemeCount(AppCompatDelegate.MODE\_NIGHT\_YES)

}

catch (ex: Exception){

println(ex)

restartActivity(this)

}

}else{

try{

AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE\_NIGHT\_NO);

appPrefs.setThemeCount(AppCompatDelegate.MODE\_NIGHT\_NO)

}

catch (ex: Exception){

println(ex)

restartActivity(this)

}

}

}

3.9 Реализация функции «Авторизоваться с помощью логина и пароля»

Для запуска данной функции пользователю необходимо заполнить поля имени пользователя и пароля. При отсутствии заполнения или частичной заполненности данных полей приложение не начинает выполнение функции. При заполнении полей вызывается диалоговое окно выбора роли для авторизации (см. рисунок 3.10). Функция реализована в классе SharedPrefs, вызов диалогового окна в классе ProfileCleanFragment.

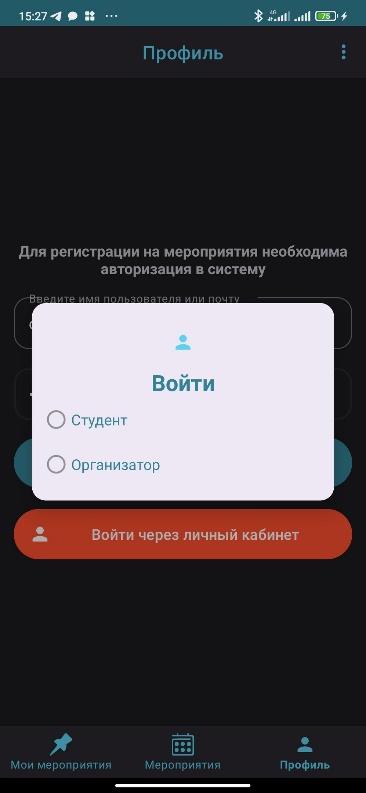


Рисунок 3.10– Авторизация в приложении

**Фрагмент кода для реализации функции**

singInButton.setOnClickListener {

if (loginText.text.toString().isEmpty()){

loginText.error = getString(R.string.error)

}

if (passwordText.text.toString().isEmpty()){

passwordText.error = getString(R.string.error)

}

if (loginText.text.toString().isNotEmpty() && passwordText.text.toString().isNotEmpty()) {

login()

}

}

3.10 Реализация функции «Зарегистрироваться на мероприятие»

Функция регистрации на мероприятие запускается нажатием соответствующей кнопки на странице мероприятия или на карточке в списке. Без авторизации в пользовательский аккаунт система предупредит о невозможности регистрации. Вызов функции осуществляется в классе EventActivity (см. рисунок 3.11)



Рисунок 3.11– Регистрация на мероприятие

**Фрагмент кода для реализации функции**

regButton.setOnClickListener {

var resp = myPreference.getLoginCount()

when(resp){

0 -> {

Toast.makeText(this, getString(R.string.clean\_profile\_text), Toast.LENGTH\_SHORT).show()

}

1 -> {

serverHelper.regForEvent(user.id, event.id)

}

}

}

3.11 Реализация функции проверки интернет-подключения

Для корректной работы приложению необходимо постоянное интернет-соединение. Каждый экран вызывает функцию проверки подключения к сети. При отсутствии подключения приложение вызывает экран NetworkErrorActivity, который предоставляет возможность проверить доступность соединения (см. рисунок 3.12). Реализация функции осуществляется в классе ServerHelper.

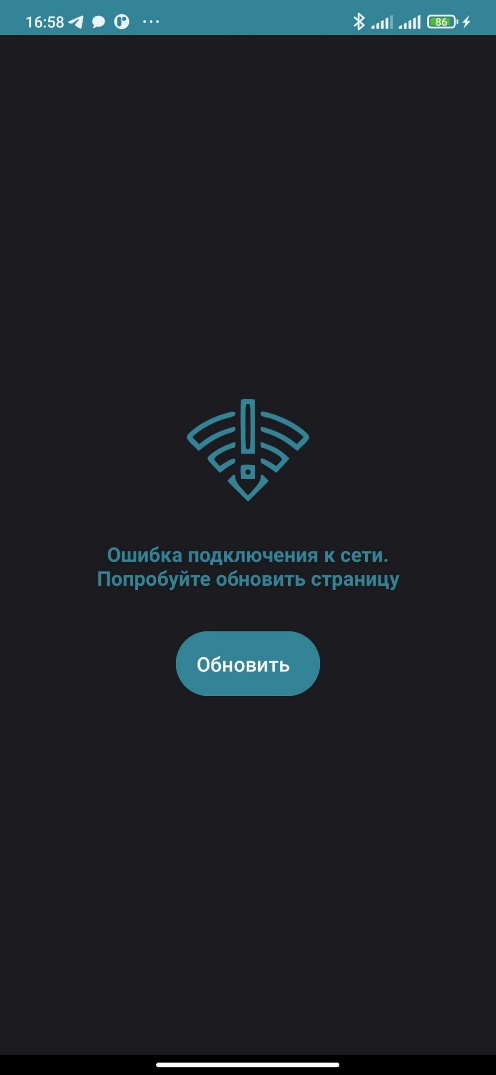


Рисунок 3.12 – Ошибка подключения к сети

**Фрагмент кода для реализации функции**

fun isOnline(context: Context): Boolean {

val connectivityManager =

context.getSystemService(Context.CONNECTIVITY\_SERVICE) as ConnectivityManager

if (connectivityManager != null) {

val capabilities =

connectivityManager.getNetworkCapabilities(connectivityManager.activeNetwork)

if (capabilities != null) {

if (capabilities.hasTransport(NetworkCapabilities.TRANSPORT\_CELLULAR)) {

Log.i("Internet", "NetworkCapabilities.TRANSPORT\_CELLULAR")

return true

} else if (capabilities.hasTransport(NetworkCapabilities.TRANSPORT\_WIFI)) {

Log.i("Internet", "NetworkCapabilities.TRANSPORT\_WIFI")

return true

} else if (capabilities.hasTransport(NetworkCapabilities.TRANSPORT\_ETHERNET)) {

Log.i("Internet", "NetworkCapabilities.TRANSPORT\_ETHERNET")

return true

}

}

}

return false

}

3.12 Реализация функции «Подтвердить присутствие участника»

Данная функция вызывается при считывании QR-кода студента сканером. Для реализации используется класс ScannerActivity (см. рисунок 3.13) и ResultScannerActivity (см. рисунок 3.14).



Рисунок 3.13 – Подтверждение присутствия

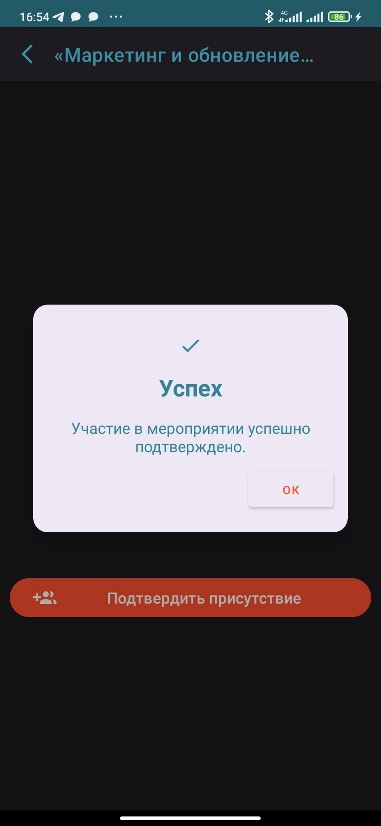


Рисунок 3.14 – Подтверждение присутствия

**Фрагмент кода для реализации функции**

confirmButton.setOnClickListener {

var resultConfirm: Boolean = serverHelper.confirmPresence(user.id, event.id)

if(resultConfirm == true){

showSuccessfulDialog()

}

else{

showErrorDialog()

}

}

3.13 Реализация функции «Просмотреть список зарегистрированных участников»

При открытии страницы статистики мероприятия приложение обращается к серверу информационной системы «Рейтинг студента», получает данные о посещаемости. Функция получения данных реализована в классе ServerHelper, заполнение данных в EventStatsActivity (см. рисунок 3.15).

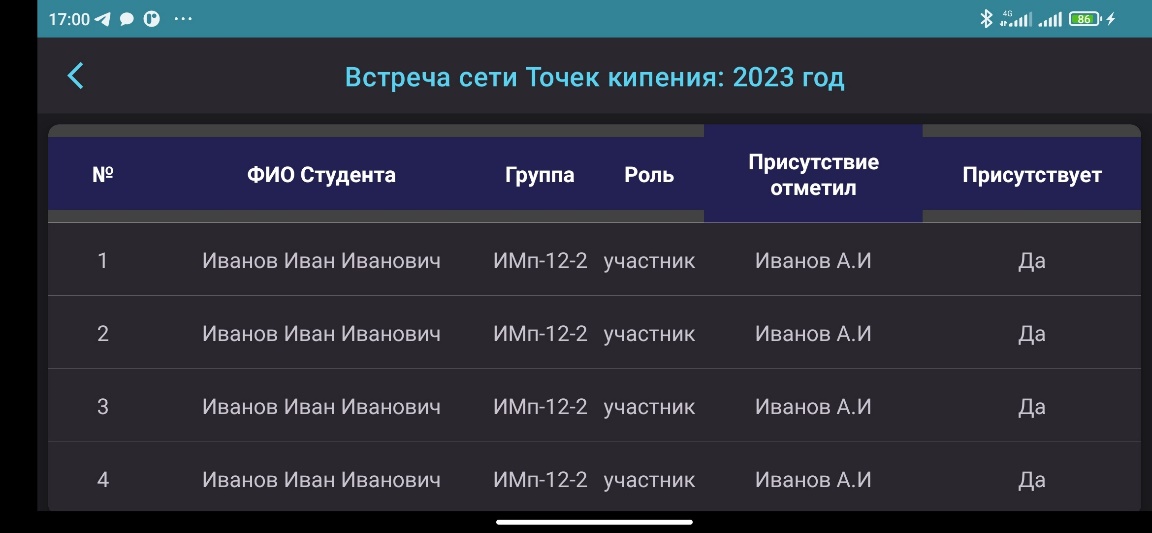


Рисунок 3.15 – Статистика мероприятия

**Фрагмент кода для реализации функции**

private fun setTable() {

var header = DataTableHeader.Builder()

.item(getString(R.string.table\_num\_text), 10)

.item(getString(R.string.table\_name\_text), 30)

.item(getString(R.string.table\_group\_text), 10)

.item(getString(R.string.table\_role\_text), 10)

.item(getString(R.string.table\_pres\_presence\_noted\_text), 20)

.item(getString(R.string.table\_pres\_text), 20)

.build()

val rows: ArrayList<DataTableRow> = ArrayList()

var i = 1

participantList.forEach {

val row = DataTableRow.Builder()

.value(i.toString())

.value(it.userName)

.value(it.studentGroup)

.value(it.role)

.value(it.presenceNoted)

.value(it.presence)

.build()

rows.add(row)

i ++

}

4 Тестирование системы

5 Экономическое обоснование

**Расчёт единовременных затрат разработчика**

Единовременные затраты включают расходы, связанные с выполнением пяти этапов проектирования программного обеспечения в нашем случае.

Таблица 5.1 – Стадии разработки приложения

|  |  |  |
| --- | --- | --- |
| **Этап** | **Трудоемкость, дн** | **Трудоемкость, %** |
| Предварительный этап | 2 | 5.6 |
| Анализ предметной обла­сти | 7 | 19.4 |
| Проектирование мобиль­ного приложения | 10 | 27.8 |
| Реализация мобильного приложения | 15 | 41.7 |
| Тестирование мобильного приложения | 2 | 5.6 |
| Итого | 36 | 100 |

К единовременным затратам разработчика отнесём затраты на разработку программного обеспечения для выпускной квалификационной работы (ВКР).

К затратам по экономическим элементам на создание информационной системы относятся:

* материальные затраты
* затраты на электроэнергию.

**Материальные затраты**

Под материальными затратами понимают отчисления на материалы, использующиеся в процессе разработки и внедрении программного продукта (в т.ч. стоимость бумаги, картриджей для принтера, дискет, дисков и т.д.) по действующим ценам.

В процессе работы использовались следующие материалы и принадлежности, представленные в смете.

* + - * 1. Материальные затраты

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование** | **Цена** | **Количество** | **Стоимость** |
| Ручки, карандаши и тд. | 5 руб | 2 | 10 |
| Печать (1 лист) | 2 руб | 80 | 160 |
| Интернет (1 гб) | 50 руб | 5 | 250 |
| **Итого** | | | **400** |

**Затраты на электроэнергию**

В процессе разработки проекта информационной системы для работы использовался ноутбук Acer Swift SF114-32. Потребление электроэнергии ноутбуком – 0,12 кВт/ч/

Стоимость 1кВт электроэнергии составляет 1,11 руб.

Стоимость машинного часа работы равна:

Время работы ЭВМ, :

Себестоимость электроэнергии рассчитывается по формуле:

Используя формулу (4), получаем:

По данным вышеприведённых расчётов составляется смета затрат на программное обеспечение.

* + - * 1. Смета затрат

|  |  |
| --- | --- |
| **Элемент затрат** | **Сметная стоимость, руб** |
| Материальные затраты | 400 руб |
| Затраты на электроэнергию | 27,972 руб |
| **Итого** | **427,972 руб** |

Затраты на ВКР равны:

**Расчет экономической выгоды по времени**

Для расчета временной выгоды необходимо узнать, какое количество времени тратится сотрудником центра заселения для принятия решения о заселении или незаселении студента.

При заселении студентов сотрудник центра заселения опирается, в первую очередь, на успеваемость студентов, иными словами, на отсутствие задолженностей. То есть студент 2-4 курса, нуждающийся в общежитии и не имея задолженностей будет заселен. У сотрудника на таких студентов уходит чуть менее 5 минуты: узнать данные студента, убедиться в отсутствие задолженностей и приступить к оформлению. Для студентов с задолженностями тратится немного больше времени. Сотруднику необходимо проверить данные студента по нескольким критериям, помимо его успеваемости: активное участие в какой-либо организации ИРНИТУ, а также отсутствие нарушений правил проживания в общежитии за предыдущий год. На таких студентов у сотрудника тратится около 7 минут.

Общежития ИРНИТУ вмещают в себя, в общей сложности, 4000 студентов (по данным приказа «О порядке заселения студентов в общежития ИРНИТУ» за 2020 г. [2]). Вычитаем из этого количества места, выделенные для международных студентов, аспирантов и студентов геологоразведочного техникума, получаем 3376 мест. Предположим, что около 50% мест занимают студенты 1 курса, тогда для 2-4 курсов остается 1688 мест. Также около 30% студентов не имеют задолженностей.

Рассчитаем время, затрачиваемое одним сотрудником для рассмотрения заявлением от студентов без задолженностей ():

Рассчитаем время, затрачиваемой одним сотрудником для рассмотрения заявлений от студентов, имеющих задолженности ():

Также учтем, что около 4-5% студентов от числа заселенных получают отказ в заселении (). На их рассмотрение тратится также около 7 минут:

Общее время:

По данным предыдущего года, в центре заселения одновременно работают около 20 человек, соответственно, время (В), затрачиваемое на рассмотрение заявок, будет равно:

С помощью разрабатываемой системы, в свою очередь, на формирование списка сотрудник потратит приблизительно 30 мин., с учетом загрузки необходимых документов в систему.

Следовательно, временная выгода () от внедрения системы составляет:

6 Безопасность жизнедеятельности

Важность данной науки происходит из сущности двух понятий, которые входят в ее название: безопасность и жизнедеятельность.

В совокупности, цель науки безопасность жизнедеятельности можно определить как достижение безопасности человека в процессе его существования в организованной социальной системе, удовлетворения им своих потребностей и выполнения предназначенных ему функций, не противоречащих интересам общества [1].

На работе, как и везде, люди сталкиваются с факторами, влияющими на жизненные показатели человека и работоспособность. Программисты, как и любые другие офисные работники, подвержены негативному влиянию данных аспектов. Поэтому для защиты от негативного воздействия существуют разные способы.

В качестве цели данного раздела выступает изучение и анализ рабочего пространства. В качестве объекта для рассмотрения выступает коворкинг «Точка кипения» - ИрНИТУ.

Задачами данного раздела являются:

1. Приведение краткой характеристики рассматриваемого объекта;
2. Проведение анализа уровня вредных и опасных факторов рабочего пространства;
3. Расчет освещения в помещении учебном помещении коворкинга;
4. Проведение анализа электробезопасности помещения;
5. Проведение анализа пожаробезопасности помещения.

6.1 Краткая характеристика рабочего помещения

Здание Иркутского национального исследовательского технического университета находится на возвышенности. Исходя из этого постройка не подвержена затоплению. Вывоз мусора с территории осуществляется специальной техникой. Территория оснащена сетью канализационных систем, осуществляющих процесс слива отходов в стоки. Вблизи здания обустроена парковочная зона для удобства водителей транспортных средств. По периметру университета, сквера студенческого городка установлен металлический забор. Таким образом выполняются требования антитеррористической защищенности объектов Министерства науки и высшего образования, согласно постановлению Правительства РФ №1421 от 7 ноября 2019 года [2]. При построении здания учитывались требования СНиП II-7-81, включающие нормы и правила строительства зданий в сейсмических районах [3]. Соблюдение данных норм обусловлено нахождением Иркутской области в сейсмическом районе акватории озера Байкал. Схема рабочего помещения представлена на рисунке 6.1.

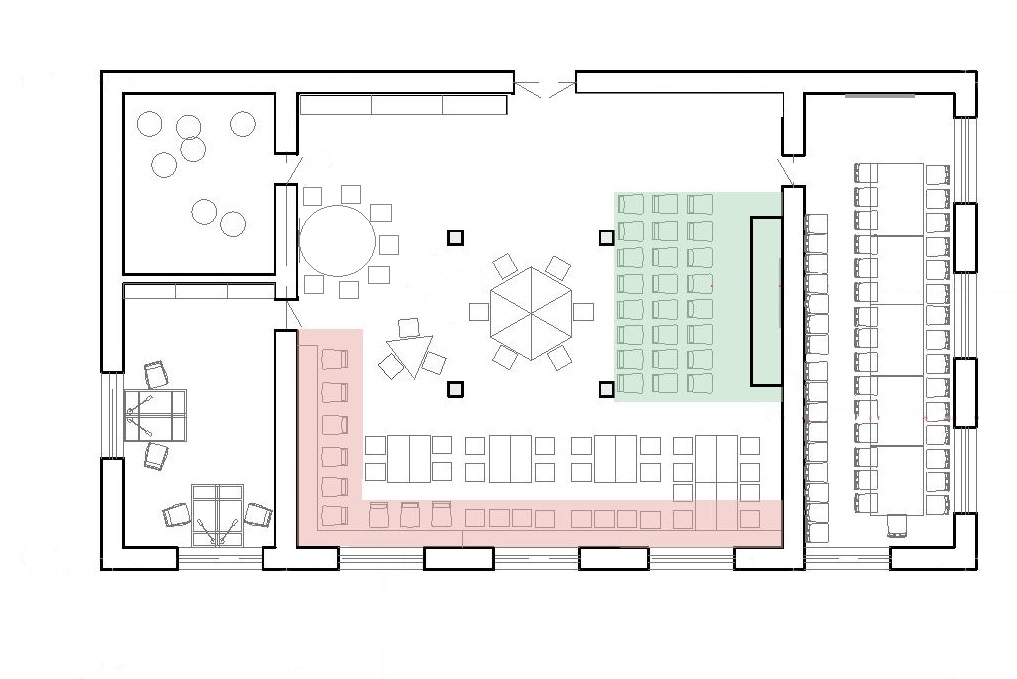


Рисунок 6.1 – Схема рабочего помещения

Рабочее пространство коворкинга представлено тремя помещениями:

1. Зал Игошина – центральное помещение, оснащенное аудио и мультимедийными системами для выступления, сценой, удобными столами и стульями для работы студентов в свободное от учебной деятельности времени, набором раскладных стульев для быстрого развертывания зрительских мест, кулером для воды и шкафом для верхней одежды. В помещении отсутствует естественное освещение, так как выходящие на улицу окна закрыты плотной тканью. Освещение искусственное;
2. Зал Головных – помещение для проведения собраний, круглых столов и лекций, оснащен большим столом набором мягких кресел для посетителей [4]. В зале установлена мультимедийная система для выступлений, оборудовано место и установлен персональный компьютер;
3. Зал НИРС – рабочее помещение для отдела развития научно-исследовательской и предпринимательской деятельности студентов. Оснащено набором офисных столов, стульев и персональных компьютеров для работы сотрудников отдела [5].

6.2 Характеристика опасных и вредных производственных факторов

Во время выполнения работы человек может оказаться под действием вредных факторов, негативно влияющих на его здоровье. Данные факторы делятся на несколько видов: физические, химические, биологические и психофизические. Классификация вредоносных и опасных факторов представлено на рисунке 6.2.

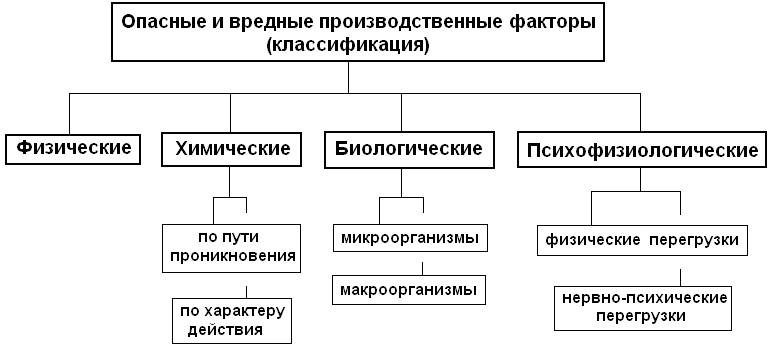


Рисунок 6.2 – Классификация вредоносных и опасных факторов

В соответствии с руководящими документами Р 2.2.2006-05 [6] и ГОСТ 12.0.003-2015 [7] перечень и характеристика воздействующих на работника вредных и опасных факторов приведен в таблице 6.1.

Таблица 6.1 – Характеристика вредных и опасных факторов

| **Опасные и вредные факторы** | **Источники опасных и вредных факторов** | **Нормируемые параметры** | **Основные средства защиты** |
| --- | --- | --- | --- |
| Вредные факторы | | | |
| Повышенная или пониженная освещенность | Неисправные осветительные приборы, загрязненность окон | ен =1,275 % Ен =300 лк СанПиН 1.2.3685-21 [8]  СанПиН 52.13330.2016 [9] | Заменить осветите-льные приборы и очистить окна от загрязнений |
| Повышенный уро-вень интенсивнос-ти звука | Кулер, оргтехни-ка, электрома-гнитный шум. Внешний шум L>80-120 дБ | L = 50 дБ СанПиН 1.2.3685-21 [8] | Монтаж звукоизо-ляции, замена обо-рудования на менее шумное |
| Повышенная  или пониженная интенсивность электромагнитного излучения | Компьютер, оргтехника | В диапазоне 5Гц – 2кГц: E=25, В/м; В=250нТл, Н= 2,5 В/м  СанПиН 1.2.3685-21 [8] | Регламентирован-ный перерыв, изо-ляционные экра-ны |

Продолжение таблицы 6.1 – Характеристика вредных и опасных факторов

|  |  |  |  |
| --- | --- | --- | --- |
| **Опасные и вредные факторы** | **Источники опасных и вредных факторов** | **Нормируемые параметры** | **Основные средства защиты** |
| Вредные факторы | | | |
| Отклонение  параметра  микроклимата  от нормиру-емых значений | Отсутствие вентиляции, систем кондиционирова-ния, наличие ще-лей в окнах, неис-правность отоп-ления | Категории работ 1b Холодный период: 𝑡=21−23℃ 𝜑=60−40%  𝜗=0,1 м/с  𝑡 поверхностей = 20-24℃.  Тёплый период: 𝑡=22−24℃ 𝜑=60−40%  𝜗=0,1 м/с 𝑡 поверхностей = 21-25℃.  СанПиН 1.2.3685-21 [8] | Установка допол-нительных обогре-вателей и кондици-онеров, вентилято-ров |
| Повышенный показатель напряженности трудового процесса | Плотность сигналов (световых, звуко-вых) и сообщений в среднем за 1 час работы | допустимо 76 – 175  Р 2.2.2006 – 05 [6] | Использовать средства индиви-дуальной защиты: специальные очки для работы за компьютером |
| Стереотипные рабочие движения (количество за смену). При локальной нагрузке (с учас-тием мышц кистей и пальцев рук) | Оптимально до 20000 однотипных движений за смену  Р 2.2.2006 – 05 [6] | Смена позы, рег-ламентированный перерыв |

Продолжение таблицы 6.1 – Характеристика вредных и опасных факторов

|  |  |  |  |
| --- | --- | --- | --- |
| **Опасные и вредные факторы** | **Источники опасных и вредных факторов** | **Нормируемые параметры** | **Основные средства защиты** |
| Вредные факторы | | | |
| Повышенный показатель напряженно-сти трудового процесса | Рабочая поза | Свободная, удобная поза, возможность смены рабочего положения тела (сидя, стоя). Нахождение в позе стоя до 40 % времени смены  Р 2.2.2006 – 05 [6] | Смена позы и регламентированный перерыв |
| Сенсорные нагру-зки: плотность сигналов (световых, звуковых) и сообщений в среднем за 1 час работы | допустимо 76 – 175 оптимально до 75  Р 2.2.2006 – 05 [6] | Использовать средс-тва индивидуальной защиты: специаль-ные очки для работы за компьютером |
| Эмоциональные нагрузки: Степень ответственности за результат собственной деятельности. Значимость ошибки | Несет ответственность за выполнение отдельных элементов заданий. Влечет за собой допол-нительные усилия в работе со стороны работника  Р 2.2.2006 – 05 [6] | Регламентирован-ный перерыв |
| Режим работы: Фактическая продолжитель-ность рабочего дня | Оптимально 6-7 часов  Р 2.2.2006 – 05 [6] |  |

Продолжение таблицы 6.1 – Характеристика вредных и опасных факторов

|  |  |  |  |
| --- | --- | --- | --- |
| **Опасные и вредные факторы** | **Источники опасных и вредных факторов** | **Нормируемые параметры** | **Основные средства защиты** |
| Вредные факторы | | | |
| Повышенный показатель напряженнос-ти трудового процесса | Сложность работы, ответственность | Перерывы регламентированы, достаточной продолжительнос-ти: 7% и более рабочего времени Р 2.2.2006 – 05 [6] | Наличие регламентирован-ных перерывов и их продолжитель-ность |
| Опасные факторы | | | |
| Опасность поражения электричес-ким током | Оголенные провода, неисправности электропровод-ки и оборудова-ния | U=220B I=0,6-1,5мА (ток самоосвобождение)  ГОСТ 12.1.019-2017 [10] | Наличие изоляций токопроводящих поверхностей. Наличие защит-ного заземления или зануления |
| Опасность возникновения пожаров и взрывов | Горючие материалы, короткое замыкание, Несоблюдение техники безопасности | Категория по пожарной опасности: В1-В4 Классов горючих веществ и материалов: Класс А Класс Е  123-ФЗ [11] | Использование предупредительной сигнализации, автоматическое пожаротушение, условия эвакуации людей |

Проанализировав таблицу 6.1, можно увеличить уровень безопасности труда на рабочем месте. Для этого достаточно лишь придерживаться рекомендациям и не переходить границы нормируемых значений.

6.3 Эргономические требования к организации рабочего помещения в Коворкинг "Точка кипения"

Эргономика рабочего места программиста – важный аспект в соблюдении требований для обеспечения безопасности жизнедеятельности. При создании правильного рабочего места необходимо соблюсти перечень условий: оптимальное размещение оборудования, достаточность пространства для работы. Дополнительно необходимо следить, чтобы было достаточно расстояния для перемещения и движения.

Главными элементами рабочего места программиста являются стол и кресло, поэтому основным рабочим положением является положение сидя. Положение сидя вызывает минимальное утомление программиста. Рациональная планировка рабочего места предусматривает четкий порядок и постоянство размещения предметов, необходимых инструментов и документации. То, что требуется для выполнения работы чаще всего, расположено в зоне легкой досягаемости рабочего пространства. На рисунке 6.3 продемонстрирована схема нормативных показателей высоты рабочего места.

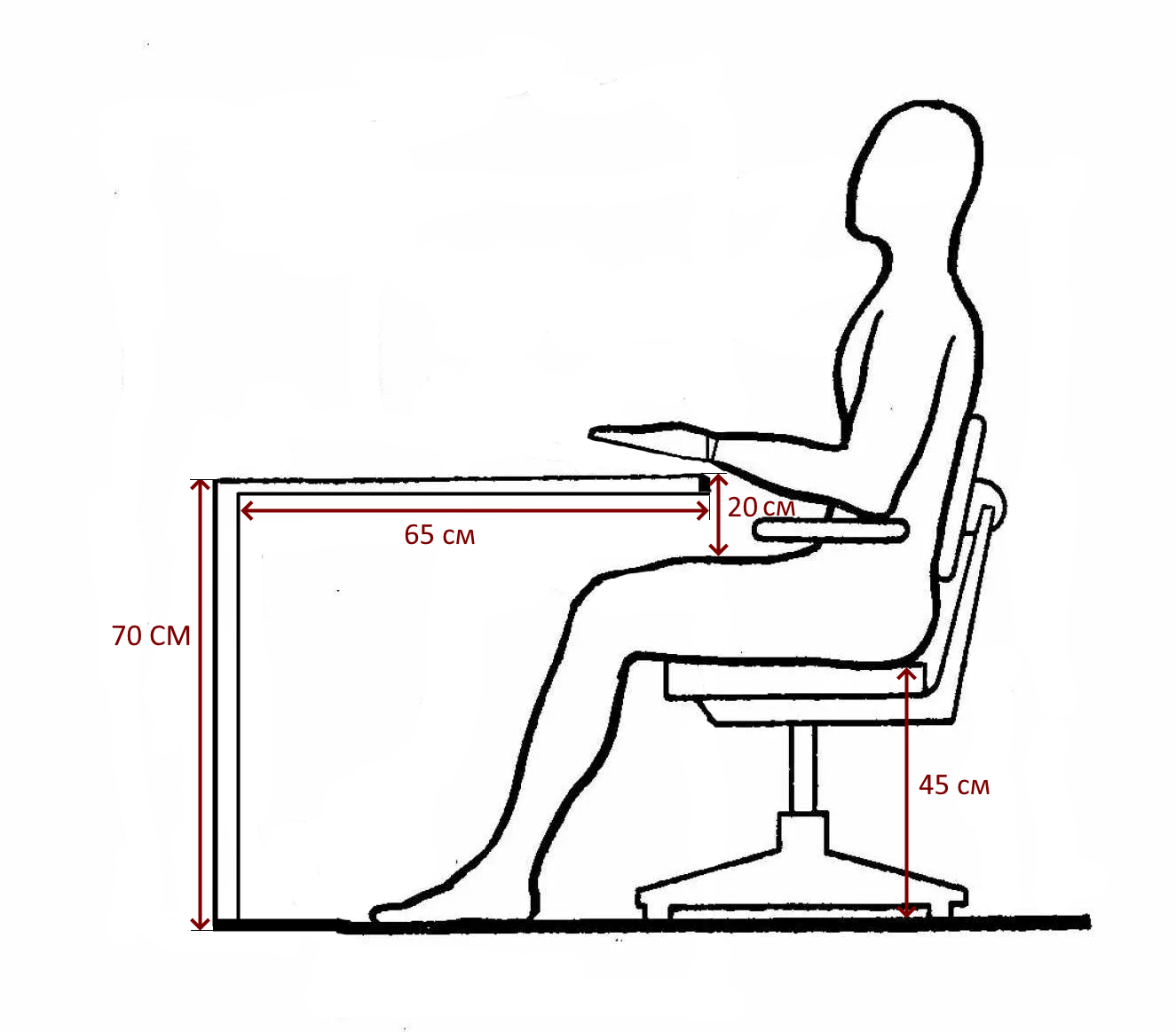


Рисунок 6.3 – Схема нормированных показателей рабочего места

Кроме того, для создания комфортных условий для работы программиста, необходимо учитывать нормативные стандарты относительно параметров высоты и пространства для ног рабочего места. Например, высота столешницы должна быть подобрана так, чтобы руки программиста находились в горизонтальном положении при работе на клавиатуре, а угол между локтями и руками не превышал 90 градусов. Кресло должно обеспечивать достаточное пространство для ног, чтобы не создавать дискомфорта при сидении в течение длительного времени. Правильное соответствие параметров рабочего места нормативным стандартам способствует сохранению здоровья и повышению эффективности работы программиста. В таблице 6.2 приведено сравнение фактических показателей рабочего места с их нормативными значениями по номограмме.

Таблица 6.2 – Сравнение значений показателей рабочего места

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Фактические значения** | | | **Нормативные значения по номограмме** | | |
| **Высота рабочей поверхнос-ти** | **Высота сидения** | **Пространст-во для ног** | **Высота рабочей поверхнос-ти** | **Высота сидения** | **Пространст-во для ног** |
| 75,5 см | 45 см | Глубина -47,5 см | 70 см | 45 см | Глубина -65 см |

Проанализировав таблицу 6.2, был сделан вывод, что показатели высоты данного рабочего места соответствуют нормативным показателям.

Нормирование тяжести трудового процесса производят согласно Методики проведения специальной оценки условий труда, утвержденной приказом Министерства труда и социальной защиты РФ от 24 января 2014 г. № 33н (прил. А) [12]. При нормировании тяжести трудового процесса учитывается пол человека и при оценки учитываются следующие показатели:

1. Физическая динамическая нагрузка;
2. Масса поднимаемого и перемещаемого груза вручную;
3. Стереотипные рабочие движения;
4. Статическая нагрузка;
5. Рабочая поза;
6. Наклоны корпуса;
7. Перемещение в пространстве.

Оценка тяжести трудового процесса представлена в таблице 6.3.

Таблица 6.3 – Оценка тяжести трудового процесса

|  |  |  |  |
| --- | --- | --- | --- |
| **Показатели  тяжести трудового процесса** | **Фактические**  **значения показателя тяжести трудового процесса** | **Предельно допустимые значения показателя тяжести трудового процесса** | **Класс (подкласс) условий труда**  **по тяжести**  **трудового процесса** |
| Масса поднимаемого и перемещаемого груза вручную | 4 кг | До 30 кг [6] | Оптимальный |
| Стереотипные рабочие движения | 29000 единиц | 40000 единиц [6] | Допустимый |
| Рабочая поза | Свободная поза комфортна и может быть изменена в любой момент по усмотрению работающего. | Периодическое, до 25 % времени смены, нахождение в неудобном и (или) фиксированном положении. Нахождение в положении «стоя» до 60 % времени рабочего дня (смены) [6]. | 1 класс условий труда |
| Наклоны корпуса | 20 единиц | 51-100 единиц [6] | 1 класс условий труда |
| Перемещение в пространстве | До 3 км по горизонтали | 8 км [6] | 1 класс условий труда |

Проанализировав таблицу 6.2, был сделан вывод, что условия труда по тяжести трудового процесса на данном рабочем месте являются оптимальными.

6.4 Электробезопасность

Под электробезопасностью понимается система мероприятий, направленных на предотвращение вредного или опасного поражения работников электрическим током, электрической дугой и т.п.

В помещении коворкинга «Точка кипения» отсутствует повышенная опасность поражения электрическим током, так как кабинет имеет класс опасности 1, проведена электрическая сеть со стандартным напряжением в 220В, проводка полностью заизолирована, оголенные провода и поврежденные электрические розетки не обнаружены, удлинители снабжены стандартным предохранителем. В случае происхождения аварийной ситуации помещение будет обесточено.

При неисправности какого-либо оборудования необходимо обесточить устройство и вызвать специалиста. Самостоятельно принимать дальнейшие действия запрещается, в противном случае неизбежно поражение электрическим током.

Для защиты работника от поражения электрическим током в системах электрификации устанавливается заземление. Защитное заземление – это особое, преднамеренное соединение с землей металлических нетоковедущих частей для защиты человека от поражения электричеством.

Схема типового заземления, установленного в университете согласно ГОСТ 12.1.019-2017 [10] представлена на рисунке 6.4.

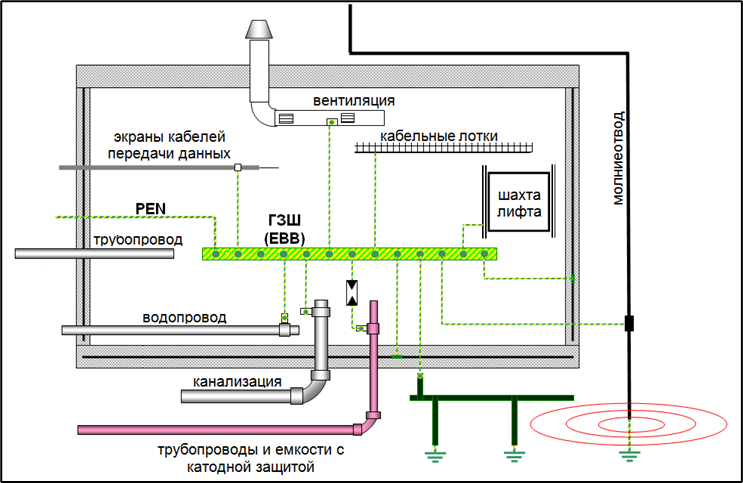


Рисунок 6.4 – Схема заземления

Рабочее помещение соответствует стандартам, прописанным в ГОСТ 12.1.019-2017 [10], что позволяет работникам продолжать трудовую деятельность без угрозы для здоровья.

6.5 Пожарная безопасность

Пожарная безопасность регламентируется Закон РФ от 22 июля 2008 г. N 123-ФЗ [11]. Пожар – неконтролируемое горение, несущее опасность для всех работников помещений. Причинами пожара могут послужить такие неисправности, как: неисправность проводки, неисправность электронагревательных устройств. Также причинами возгорания может стать человеческий фактор: халатное отношение работников к пожарной безопасности, умышленный и неумышленный поджог.

В рассматриваемом помещении для защиты от пожара установлены датчики дыма, представленные на рисунке 6.5. Принцип работы датчиков дыма заключается в том, что посылаемый луч в приборе при наличии дыма рассеивается, сигнализируя системе о наличии задымления.



* + - 1. Рисунок 6.5 – Датчик дыма «ИПД-3.2»

В коридорах университета расположены пожарные шиты, предусмотренные на случай возникновения возгорания. В каждом корпусе на одном этаже установлено три таких щита. Внутреннее содержимое в каждом представлено огнетушителем и шлангом. На рисунке 6.6 проиллюстрирован один из установленных щитов.



Рисунок 6.6 – Пожарный щиток

На рисунке 6.7(а) изображен огнетушитель ОУ-2, на рисунке 6.7(б) изображен пожарный рукав, которые установлены в вышеописанные пожарные щитки.

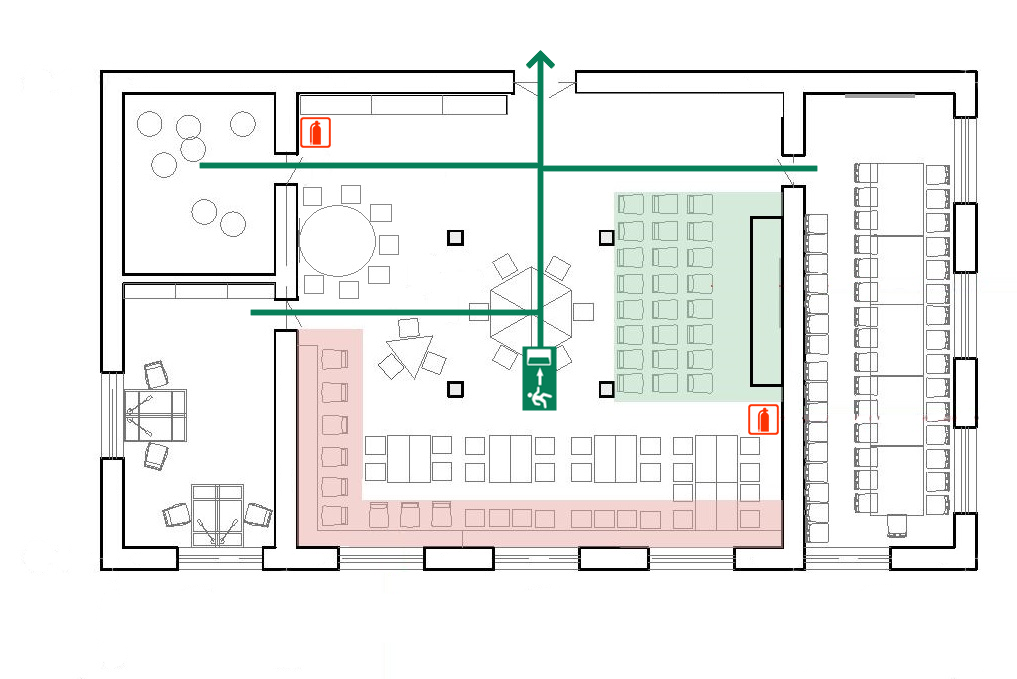


а) б)

* + - 1. Рисунок 6.7 – Огнетушитель ОУ-2 и пожарный рукав

Внутри огнетушителя ОУ – 2 находится углекислота и углекислый газ в жидкой форме под давлением от 5,7 до 15 Мпа, вырываясь наружу мгновенно сбивает пламя, лишая его кислорода, вследствие этого огонь не может далее гореть. Пожарный рукав имеет стандартную длину 20 м.

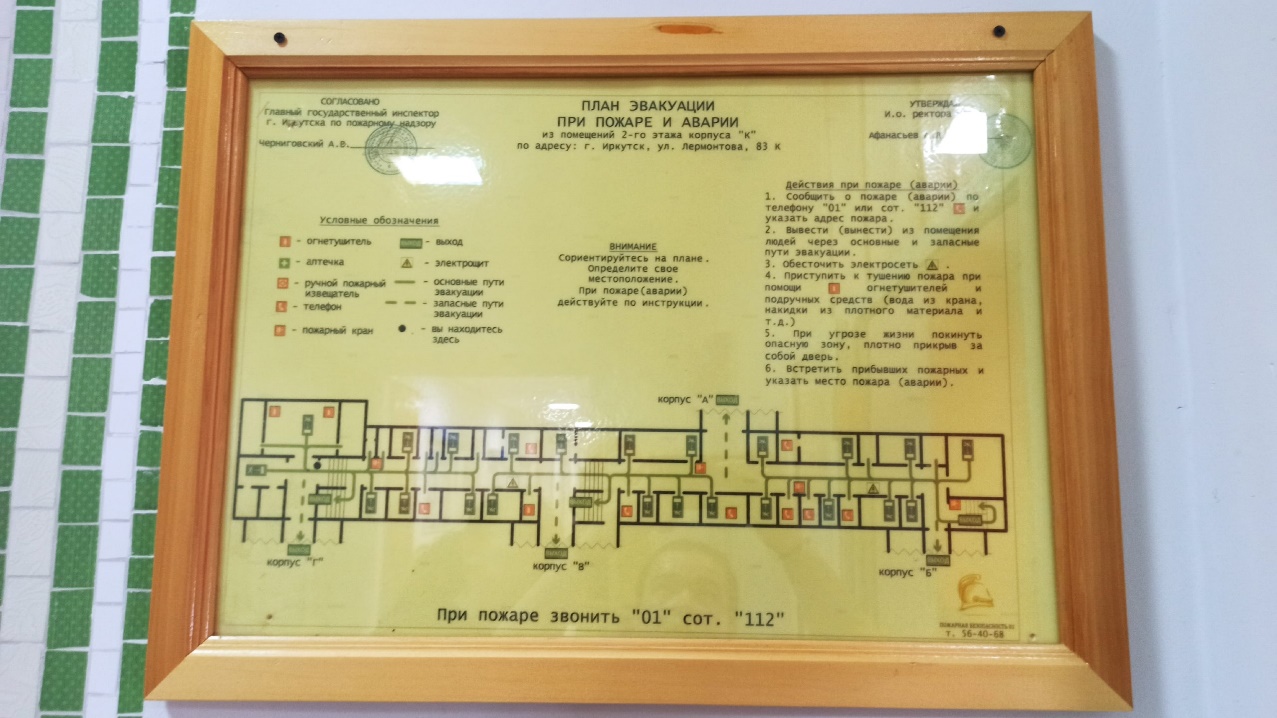
На случай возникновения пожароопасной ситуации в центральном зале Игошина расположены огнетушители. На рисунке 6.8 изображена схема эвакуации из помещения.



* + - 1. Рисунок 6.8 – Схема эвакуации из помещения

6.6 Обеспечение безопасности в условиях ЧС

В случае возникновения чрезвычайных ситуаций социального характера или пожара на каждом этаже учебных корпусов здания установлена схема эвакуации в случаях чрезвычайных ситуаций. На рисунке 6.8 представлена схема эвакуации, расположенная перед входом в помещение «Точки кипения».



* + - 1. Рисунок 6.8 – Схема эвакуации

В случае возникновения чрезвычайной ситуации люди, находящиеся в данном помещении, должны без спешки и давки проследовать по лестнице к выходу корпуса Г здания университета.

Здание соответствует всем необходимым стандартам для проведения учебных занятий.

6.7 Вывод по главе

Из проведенного анализа в рамках данного раздела, можно сделать вывод о соответствии рабочего пространства помещения коворкинга «Точка Кипения» всем необходимым требованиям для проведения трудовой деятельности. Помещение оснащено всем необходимым для комфортной работы и проведения мероприятий. Эргономика помещения соответствует необходимым параметрам для выполнения работ. На случай происхождения чрезвычайных ситуаций помещение оснащено огнетушителями. При необходимости эвакуации в коридоре около входа в коворкинг установлена схема, сообщающая маршрут эвакуации из здания.

Заключение

Список использованных источников

1. Метод построения словарей предметных областей для извлечения фактов из текстов на естественном языке [Электронный ресурс] URL: https://cyberleninka.ru/article/n/metod-postroeniya-slovarey-predmetnyh-oblastey-dlya-izvlecheniya-faktov-iz-tekstov-na-estestvennom-yazyke (дата обращения 27.04.2023);

2. Ирниту – Рейтинг [Электронный ресурс] URL: https://www.istu.edu/studentu/stipendiya/reyting (дата обращения 28.04.2023);

3. RUNET О проекте [Электронный ресурс] URL: https://runet-id.com/page/info/about (дата обращения 29.04.2023);

4. Leader-ID [Электронный ресурс] URL: https://leader-id.ru/intro (дата обращения 02.05.2023);

5. ФГАИС «Молодежь России» - Мероприятия [Электронный ресурс] URL: https://myrosmol.ru/measures (дата обращения 03.05.2023).

1. Безопасность жизнедеятельности (лекционный курс) [Электронный ресурс] URL: <https://cyberleninka.ru/article/n/bezopasnost-zhiznedeyatelnosti-lektsionnyy-kurs/viewer>. (дата обращения 10.04.23)

2. Постановление Правительства РФ от 7 ноября 2019 г. N 1421 [Электронный ресурс] URL: <https://base.garant.ru/72979284/>. (дата обращения 10.04.23)

3. СП 14.13330.2010. Строительство в сейсмических районах. [Электронный ресурс] URL: https://docs.cntd.ru/document/1200000291. (дата обращения 10.04.23)

4. ИРНИТУ – Точка кипения – ИРНИТУ [Электронный ресурс] URL: https://www.istu.edu/deyatelnost/molodezhnaya\_politika/nauka/tk. (дата обращения 11.04.23)

5. ИРНИТУ – Отдел развития научно-исследовательской и предпринимательской деятельности студентов [Электронный ресурс] URL: <https://www.istu.edu/deyatelnost/nauka/nauchnoe_upravlenie/nirs>. (дата обращения 11.04.23)

6. Р 2.2.2006-05. Руководство, по гигиенической оценке, факторов рабочей среды и трудового процесса. Критерии и классификация условий труда. [Электронный ресурс] URL: https://docs.cntd.ru/document/1200040973. (дата обращения 11.04.23)

7. ГОСТ 12.0.003-2015. Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация. [Электронный ресурс] URL: https://docs.cntd.ru/document/1200136071. (дата обращения 12.04.23)

8. СанПиН 1.2.3685-21. Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания. [Электронный ресурс] URL: https://docs.cntd.ru/document/573500115. (дата обращения 12.04.23)

9. СанПиН 52.13330.2016. Естественное и искусственное освещение. [Электронный ресурс] URL: https://docs.cntd.ru/document/456054197. (дата обращения 13.04.23)

10. ГОСТ 12.1.019-2017. Система стандартов безопасности труда. Электробезопасность. Общие требования и номенклатура видов защиты. [Электронный ресурс] URL: https://docs.cntd.ru/document/1200161238. (дата обращения 13.04.23)

11. Федеральный закон "Технический регламент о требованиях пожарной безопасности" от 22.07.2008 N 123-ФЗ. [Электронный ресурс] URL: https://docs.cntd.ru/document/902111644. (дата обращения 13.04.23)

12. Приказ Минтруда России от 24.01.2014 N 33н. [Электронный ресурс] URL: <https://www.consultant.ru/document/cons_doc_LAW_158398/>. (дата обращения 13.04.23)