

Аннотация

Тема: «Разработка мобильного приложения «Регистрация на мероприятия для ИС «Рейтинг студентов».

Ключевые слова: информационные технологии, мобильное приложение, рейтинговая стипендия.

Выполнил: Вовиков Даниил Евгеньевич, 09.03.02 «Информационные системы и технологии на предприятиях / в машиностроении»

Руководитель: Черкашин Евгений Александрович, кандидат технических наук, доцент.

Первый раздел посвящён анализу предметной области:

1. Рассмотрены система начисления рейтинговой стипендии.
2. Проанализированы учет мероприятий.
3. Описана исследуемая предметная область.
4. Выполнена постановка задачи.
5. Выдвинуты требования к ролям и правам доступа.
6. Описано функциональное назначение системы.
7. Сформированы функциональные требования к системе.
8. Проведен обзор аналогов.
9. Анализ базы данных информационной системы «Рейтинг студента».
10. Выдвинуты требования графического интерфейса приложения.

Во втором разделе проектируется мобильное приложение. В рамках проектирования было выполнено:

1. Обоснованы выбранные средства разработки и проектирования.
2. Предложена архитектура подсистемы.
3. Спроектированы классы данных.
4. Описаны сценарии использования.
5. Спроектирован графический интерфейс.

Третий раздел посвящён реализации мобильного приложения. Были описаны реализованные функции, приведена спецификация классов.

В четвертом разделе проведено тестирование работоспособности мобильного приложения с помощью фокус-группы.

В пятом разделе представлены экономические расчеты по разработке подсистемы.

В шестом разделе приводятся результаты работы по организации безопасности жизнедеятельности студентов, участвующих в мероприятиях.

Основные выводы по результатам проведенной работы: изучен процесс регистрации студентов на мероприятия, выполнен анализ предметной области, сформулирован перечень требований для приложения, выполнено проектирование программного обеспечения, реализовано приложение.

Объем пояснительной записки – 83 страниц.

Объем приложений – 18 страницы.

Количество рисунков – 62.

Количество таблиц – 9.

Объем списка литературы – 32.

Содержание

Введение.....	7
1 Анализ предметной области	9
1.1 Система начисления рейтинговой стипендии.....	9
1.2 Учет мероприятий.....	10
1.3 Описание предметной области	11
1.4 Постановка задачи	13
1.5 Требования к ролям и правам доступа	14
1.6 Функциональное назначение системы.....	16
1.7 Формирование функциональных требований пользователей к системе.....	21
1.8 Обзор аналогов	22
1.9 Анализ базы данных информационной системы «Рейтинг студента»...	24
1.10 Требования графического интерфейса приложения	26
1.11 Вывод по главе	28
2 Проектирование системы	29
2.1 Обоснование использования программных средств разработки	29
2.2 Архитектура подсистемы	31
2.2.1 Общая архитектура системы	31
2.2.2 Архитектура мобильного приложения	32
2.3 Проектирование классов данных	35
2.4 Описание сценариев использования	37
2.4.1 Спецификация вариантов использования «Просмотреть список мероприятий»	38
2.4.2 Спецификация варианта использования «Зарегистрироваться на мероприятие».....	38
2.4.3 Спецификация варианта использования «Подтвердить присутствие участника»	39
2.5 Проектирование графического интерфейса	39
2.5.1 Проектирование графического интерфейса для роли «Студент».....	41
2.5.2 Проектирование графического интерфейса для роли «Ответственный за мероприятие»	46
2.5.3 Проектирование графического интерфейса непривязанных к конкретной роли.....	52
3 Реализация системы	56
3.1 Реализация функции «Просмотреть мероприятие»	56
3.2 Реализация функции проверки интернет-подключения	56
3.3 Реализация функции «Сформировать QR-код»	57
3.4 Реализация функции «Подтвердить присутствие участника»	58
3.5 Реализация функции «Сохранить QR-код в pdf-файл».....	60
3.6 Реализация функции «Отправить pdf-файл в мессенджер»	60
3.7 Реализация функции «Добавить мероприятие в календарь смартфона».....	61
3.8 Реализация функции «Отменить регистрацию на мероприятие»	62
3.9 Реализация функции смены языка приложения	63
3.10 Реализация функции смены графической темы приложения	64
3.11 Реализация функции «Авторизоваться с помощью логина и пароля».....	64

3.12 Реализация функции «Зарегистрироваться на мероприятие»	65
3.13 Реализация функции «Просмотреть список зарегистрированных участников»	65
4 Тестирование системы	66
5 Экономическое обоснование	69
5.1 Расчет единовременных затрат разработчика	69
5.2 Расчет затрат по экономическим элементам	69
5.3 Расчет экономической выгоды по времени	71
6 Безопасность жизнедеятельности	72
6.1 Краткая характеристика рабочего помещения	72
6.2 Характеристика опасных и вредных производственных факторов	73
6.3 Эргономические требования к организации рабочего помещения в Коворкинг "Точка кипения"	76
6.4 Электробезопасность	77
6.5 Пожарная безопасность	78
6.6 Обеспечение безопасности в условиях ЧС	80
6.7 Вывод по главе	80
Заключение	81
Список использованных источников	82
Приложение А	84
Приложение Б	88
Приложение В	101

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 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 процессов 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 [1].

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 деятельности, 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 ответственному 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.

1.3 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 [2]. 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	
	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()
	Учебная группа	
	Статус присутствия	
Статус пользователя	Участник	
	Ответственный	
	Зритель	
Статус присутствия	Присутствие	Отметить присутствие()
QR-код	Uid пользователя	Просмотреть QR-код()
		Сформировать QR-код()

Продолжение таблицы 1.2

Классы	Свойства	Методы
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 которого являются студенты и ответственные за мероприятия. Для студентов приложение должно предоставлять возможность просмотра информации о мероприятии, регистрации на мероприятие 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 подтверждать фактическое присутствие участника на мероприятии. Это должно осуществляться путем сканирования QR-кода студента 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 студентов на мероприятия, подтверждении присутствия и других действий, 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 функциональности и правах доступа для разных типов пользователей. За основу разграничения доступа использован ГОСТ Р 59383-2021 [3].

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

Продолжение таблицы 1.3

Роль	Подсистема	Функция
Сту- дент	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 меро- приятия	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
Ответ- ствен- ный за меро- прия- тие	Мероприятия	просмотр списка предстоящих мероприятий
		выбор мероприятия
		регистрация на мероприятие в качестве ответ- ственного
		просмотр подробного описания выбранного мероприятия
		фильтрация мероприятий
		поиск определенного мероприятия по названию
	Пользователь	возможность авторизации с помощью системы «Кампус»
		возможность авторизации с помощью логина и пароля
		возможность выхода из учетной записи пользователя

Продолжение таблицы 1.3

Роль	Подсистема	Функция
Ответственный за мероприятие	Подтверждение присутствия	выбор необходимого мероприятия
		считывание уникального кода участника студента
		подтверждение присутствия 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 [4]. 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 с регистрацией и подтверждением присутствия на мероприятиях.
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.

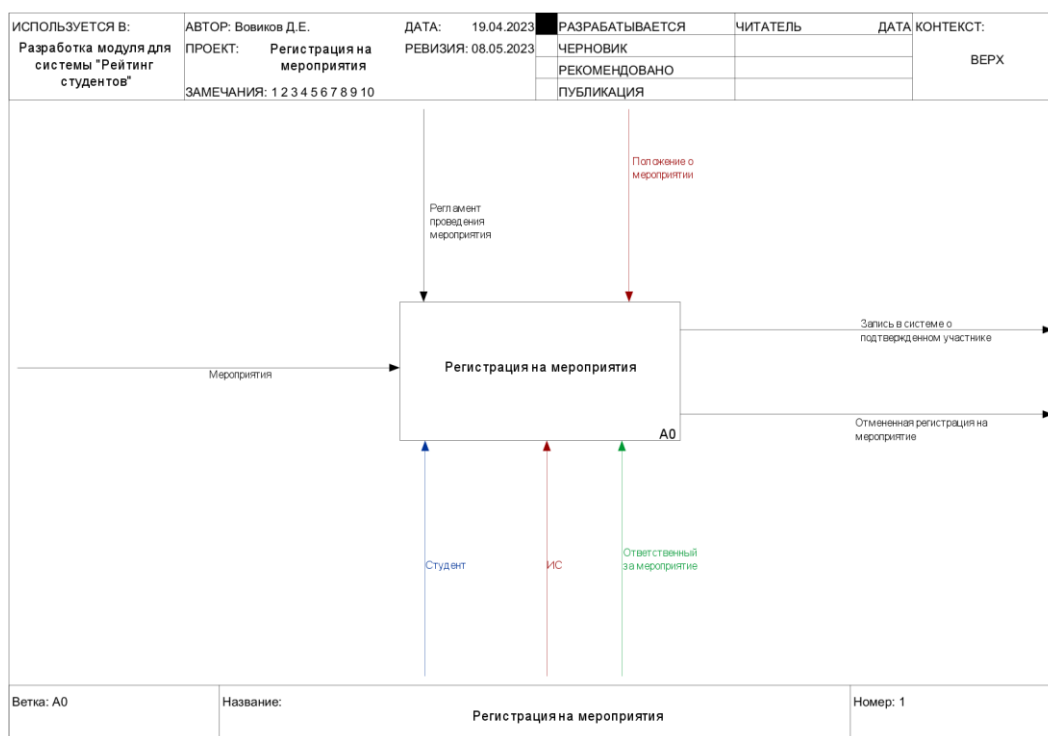


Рисунок 1.2 – Блок «Регистрация на мероприятия»

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.

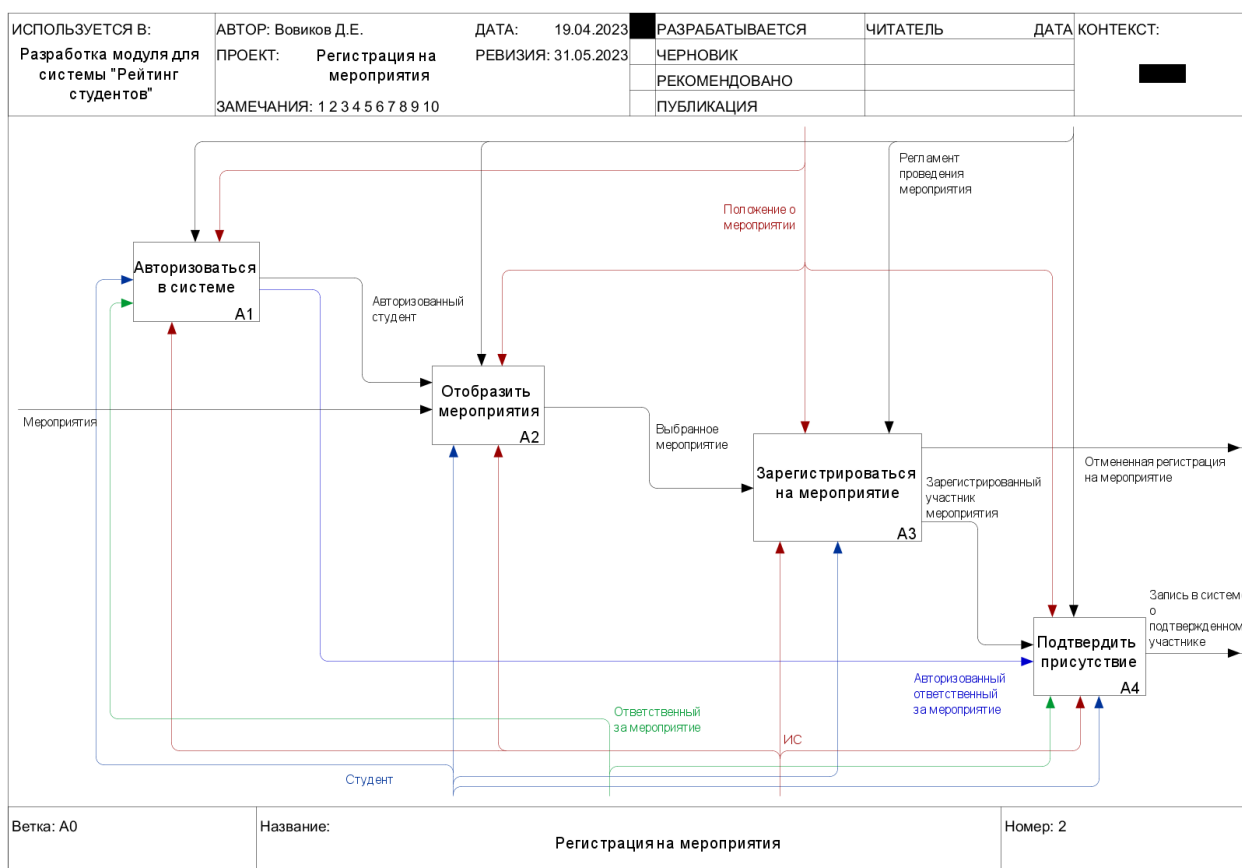


Рисунок 1.3 – Блок «Регистрация на мероприятия»

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.

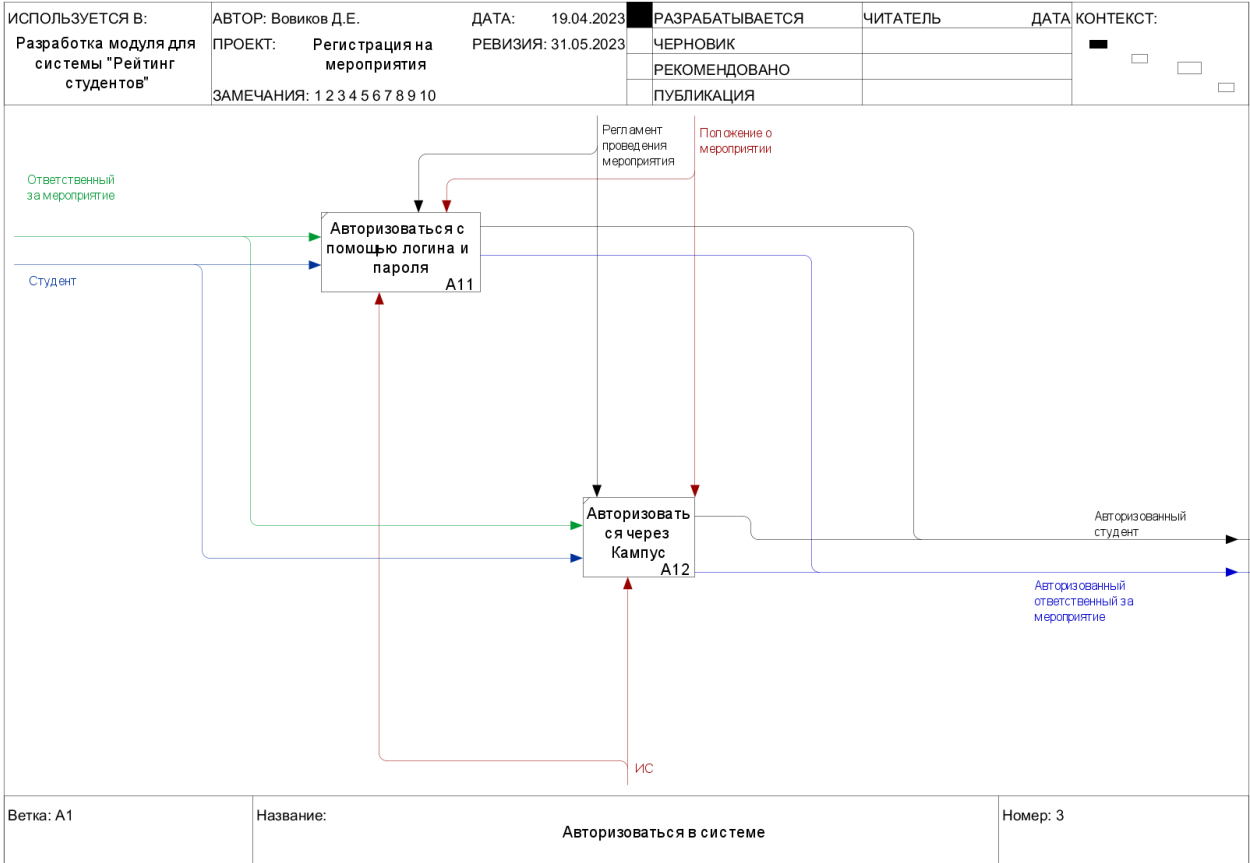


Рисунок 1.4 – Декомпозиция блока «Авторизоваться в системе»

На рисунке 1.5 представлена декомпозиция процесса A2 «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.

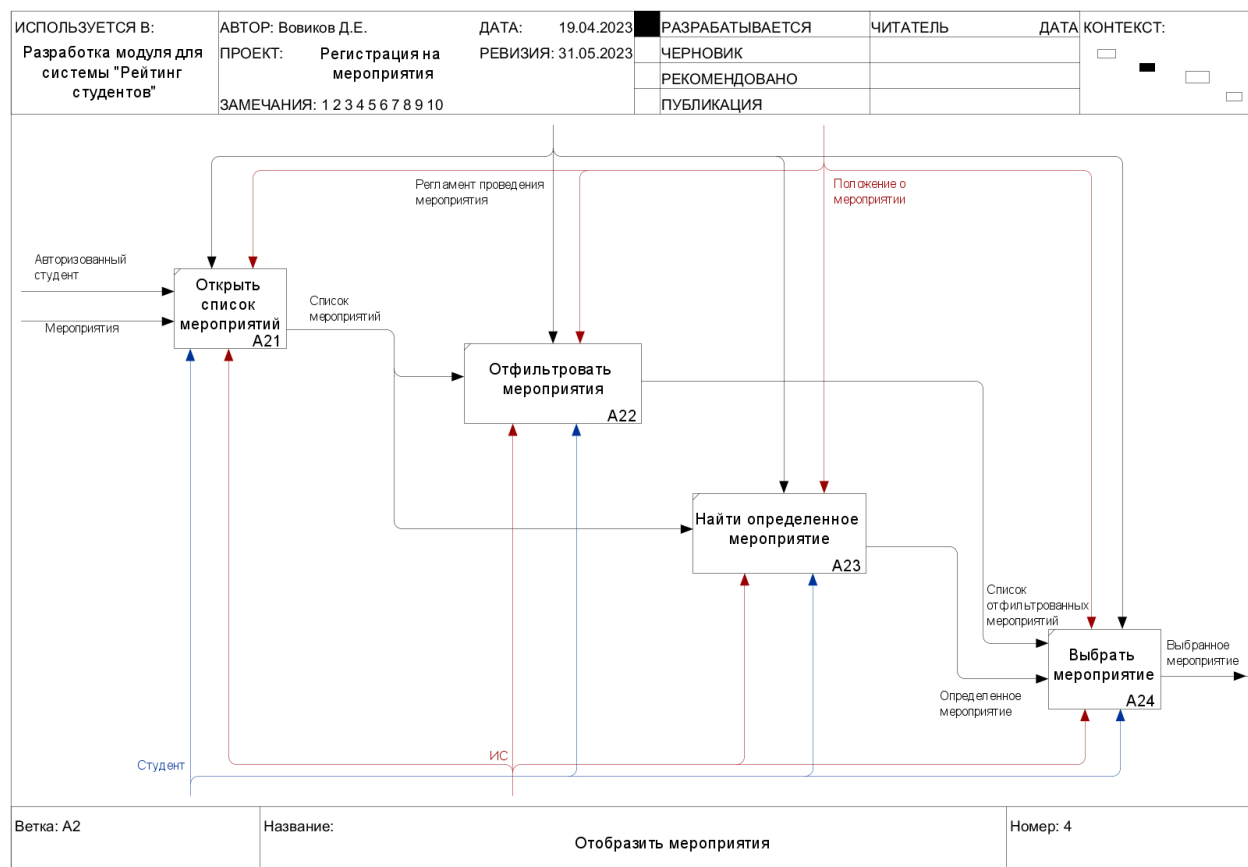


Рисунок 1.5 – Декомпозиция блока «Отобразить мероприятия»

На рисунке 1.6 представлена декомпозиция процесса A3 «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 ранее 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.

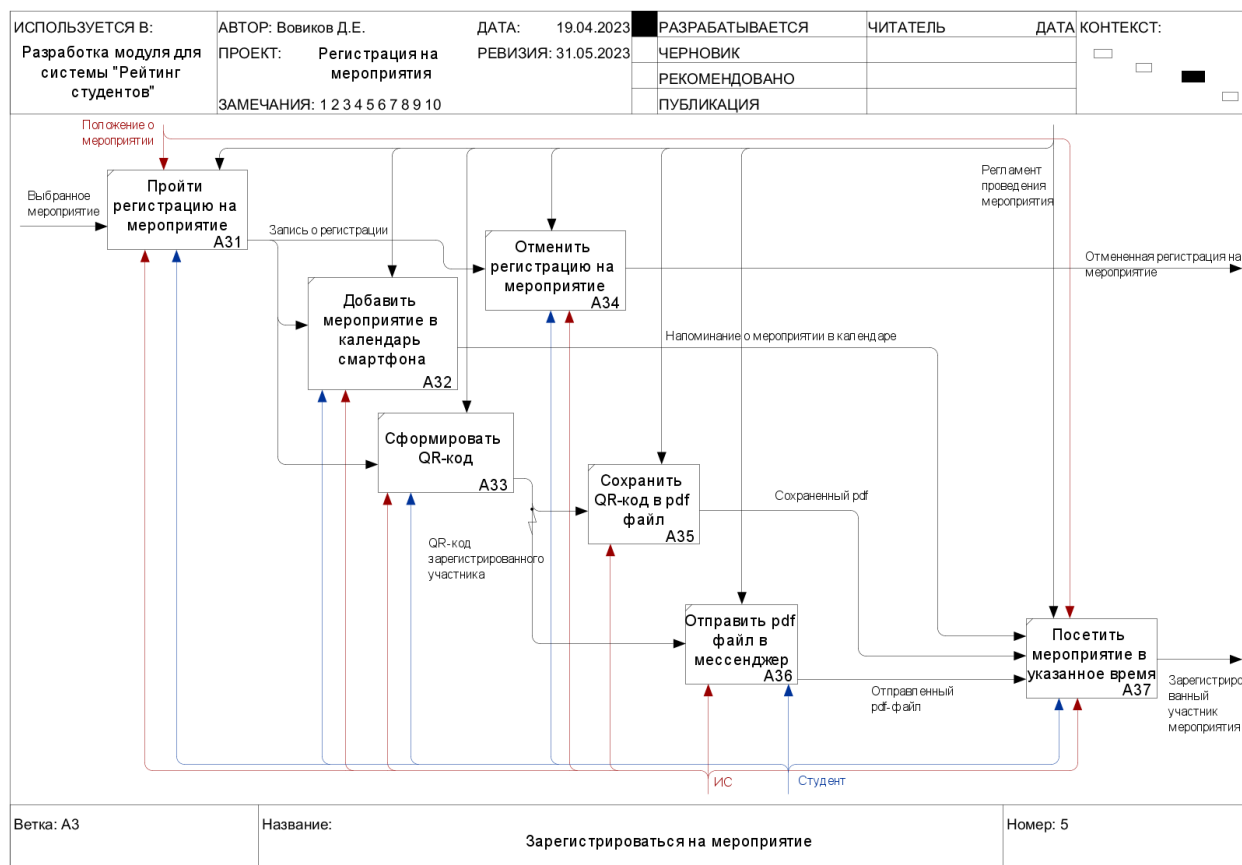


Рисунок 1.6 – Декомпозиция блока «Зарегистрироваться на мероприятие»

На рисунке 1.7 представлена детализация блока A4 «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.

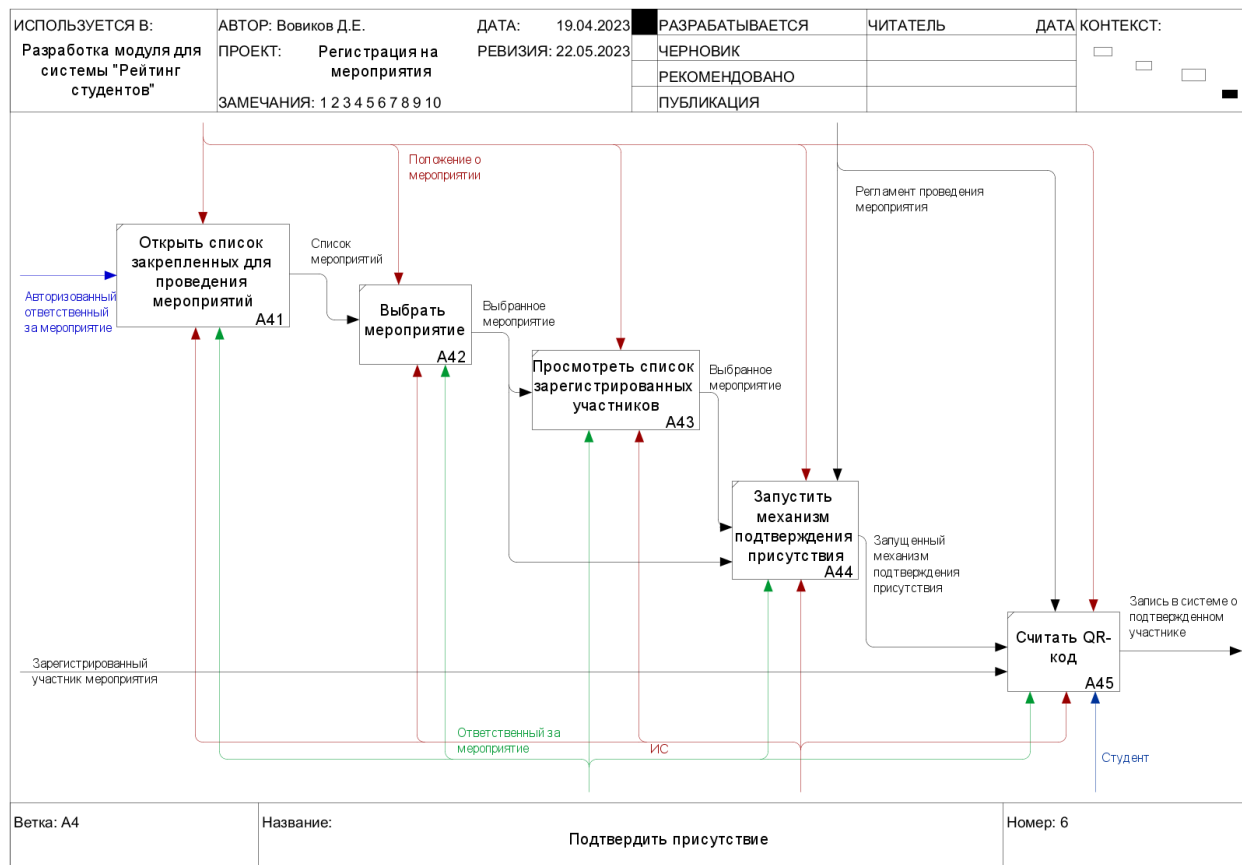


Рисунок 1.7 – Декомпозиция блока «Подтвердить присутствие»

1.7 Формирование функциональных требований пользователей к системе

Для описания функциональности разрабатываемого программного модуля были выделены функции разрабатываемой системы. В системе обозначены 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. Описание выполнено с помощью диаграмм использования [5].

На рисунке 1.8 представлены функции разрабатываемой системы.

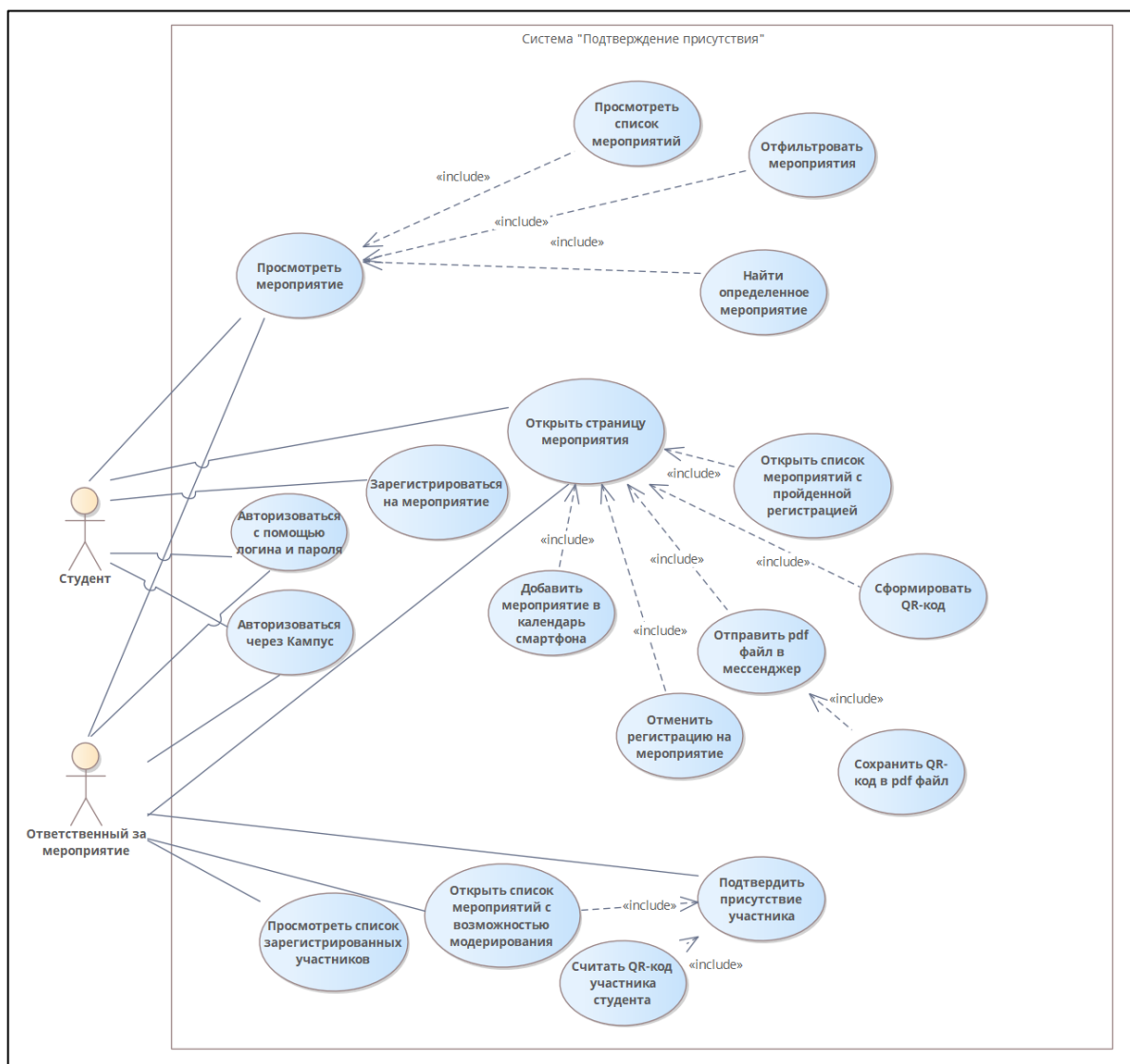


Рисунок 1.8 – Функции разрабатываемой системы

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 [6]. 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 [7].

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 [8].

1.9 Анализ базы данных информационной системы «Рейтинг студента»

Логическая модель 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.

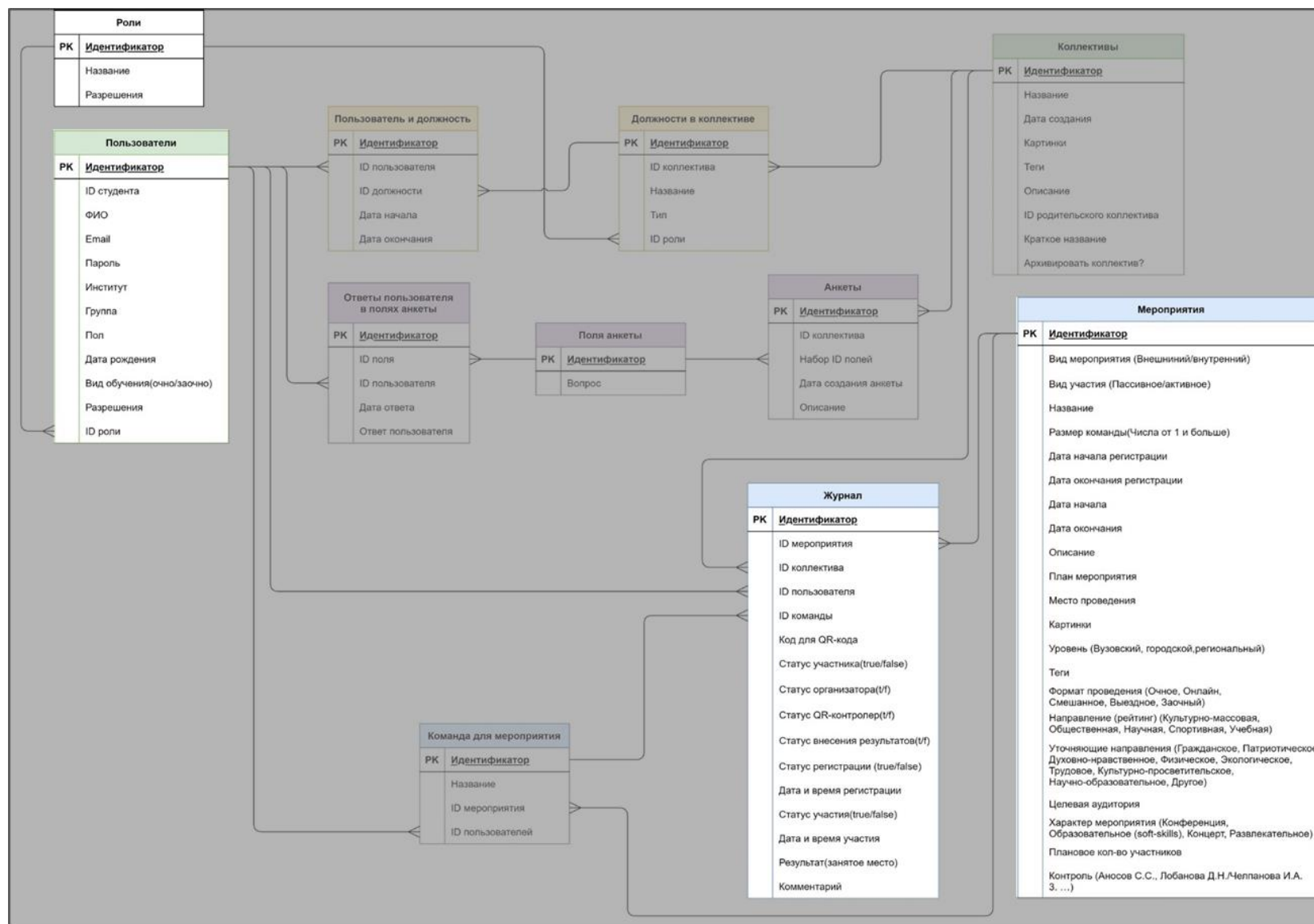


Рисунок 1.9 – Логическая модель данных ИС «Рейтинг студента»

1.10 Требования графического интерфейса приложения

Для реализации графического интерфейса спроектирован набор экранов.

На рисунке 1.10. 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.

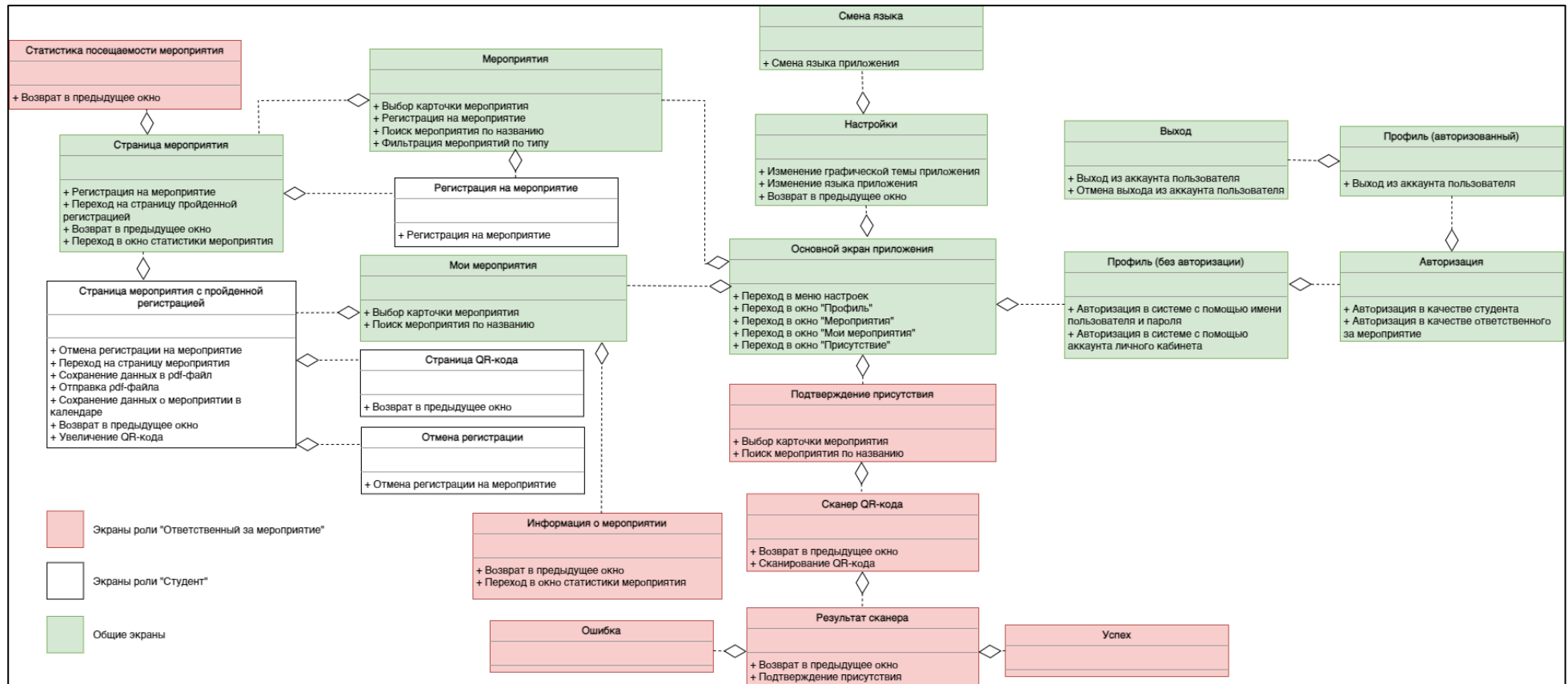


Рисунок 1.10 – Диаграмма интерфейсных экранов приложения

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 функциональность приложения для двух ролей – студент и ответственный за мероприятия. Для студента необходимо реализовать функциональность 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 [9].

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 [10].

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 [11].

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 [12].

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 [13].

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 [14].

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 [15].

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 [16].

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 данных 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.

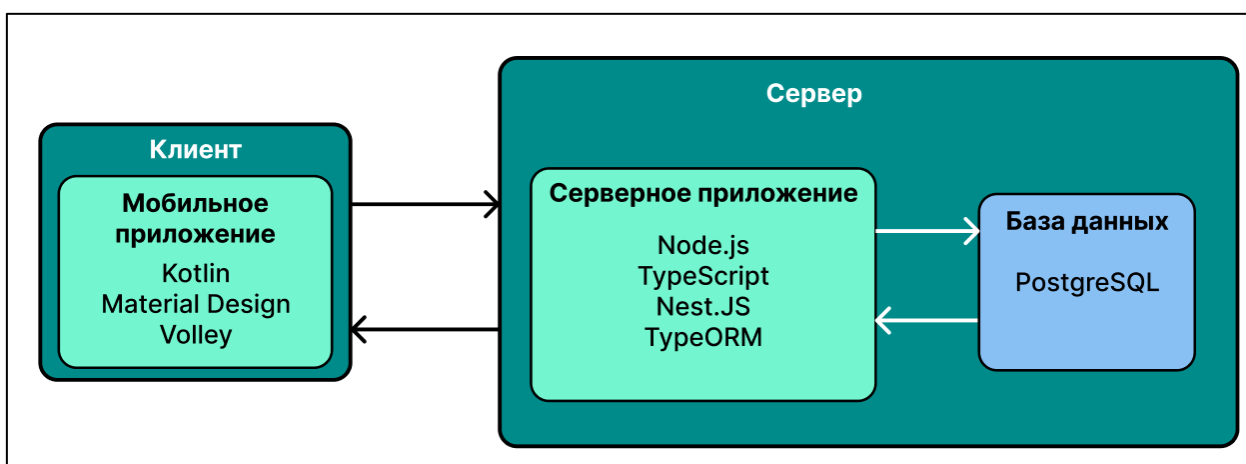


Рисунок 2.1 – Архитектура системы

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 [17].

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 пройденным регистрацией.

8. MainActivity – основное окно приложения.

9. EventStatsActivity – Activity для отображения статистики мероприятия.

10. NetworkErrorActivity – Activity отображения информации об ошибке 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.

Многослойная архитектура преобразована в пакеты классов, каждый из которых представляет тот или иной слой. Интерфейсный слой (UI Layer) разделен на два пакета: Activity и Fragment. Слой адаптеров (Adapter Layer) представлен пакетом Adapter. Слой данных (Data Layer), состоящий из набора Data-классов для хранения данных, представлен в виде пакета Model. Слой Helper Layer полностью реализован в пакете Helper. Общая диаграмма пакетов представлена на рисунке 2.3. Диаграммы пакетов представлены в приложении А.

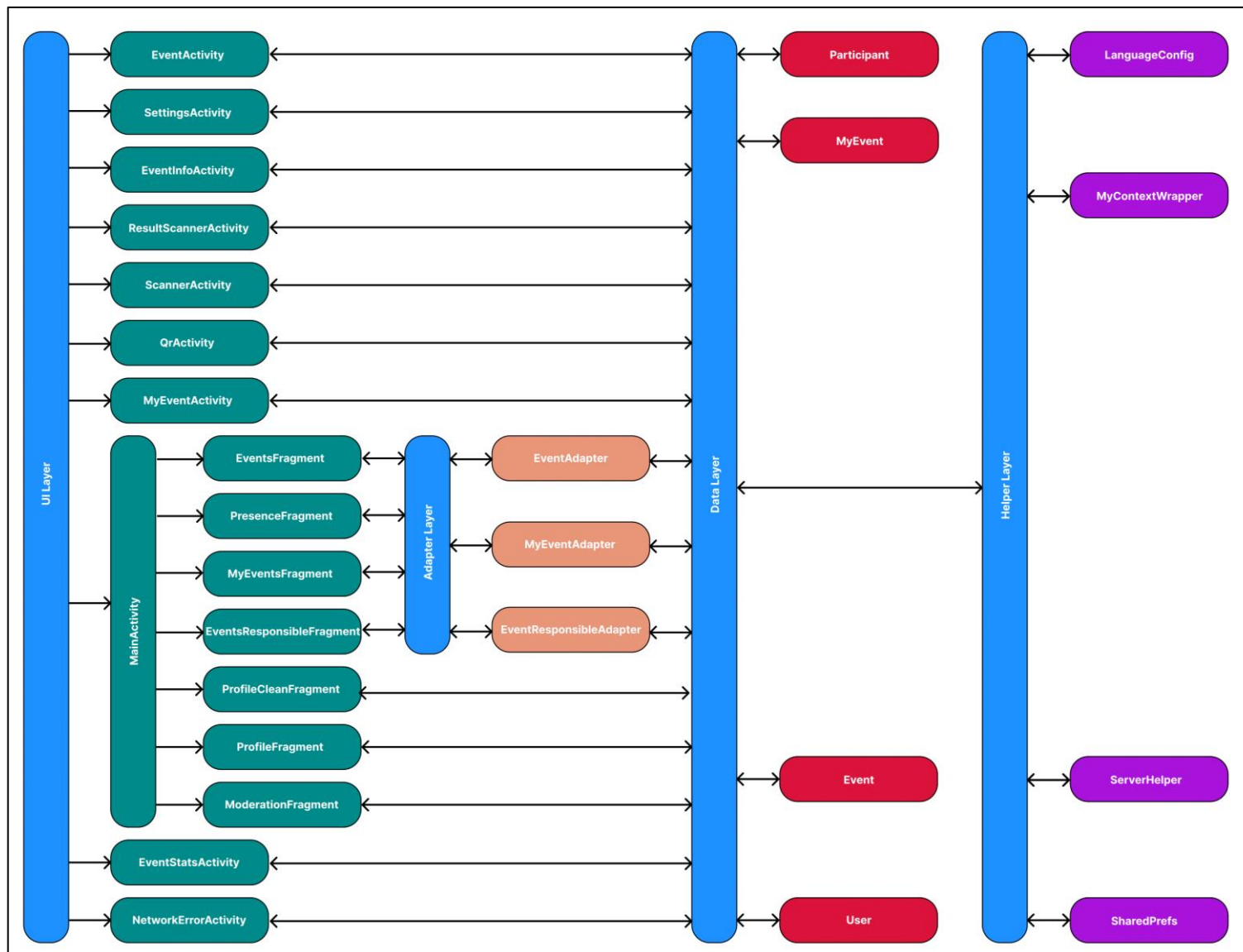


Рисунок 2.2 – Архитектура мобильного приложения

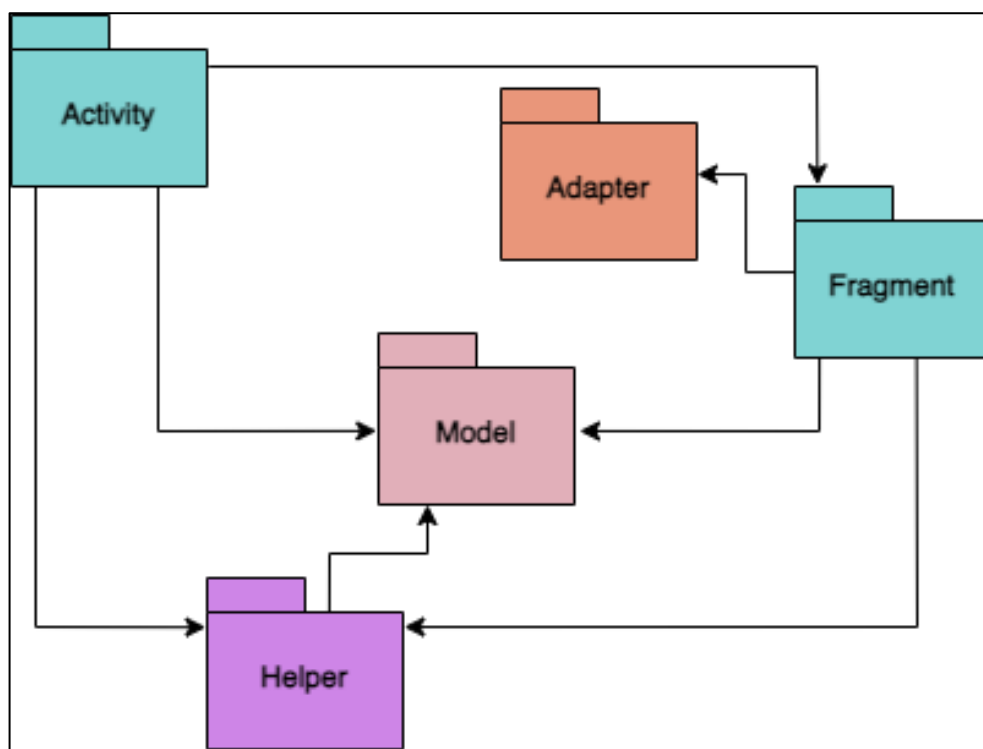


Рисунок 2.3 – Архитектура пакетов мобильного приложения

2.3 Проектирование классов данных

Классы данных описывают систему управления мероприятиями и участниками в мобильном приложении и предназначены для хранения данных сущностей [18].

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 2.4 представлена схема классов данных, в таблице 2.1 представлено описание полей.

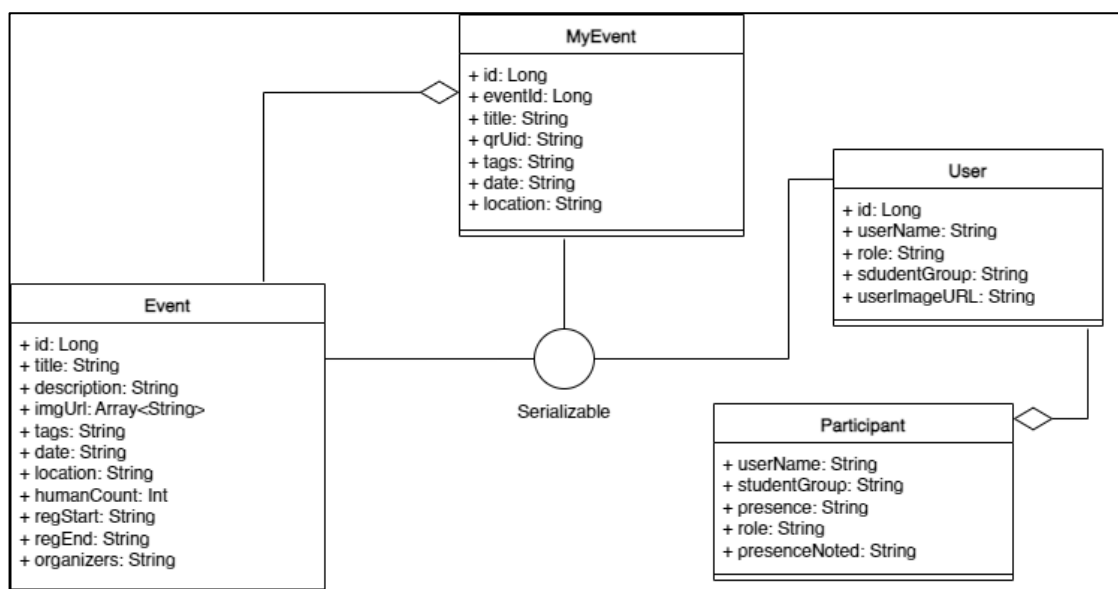


Рисунок 2.4 – Классы данных приложения

Таблица 2.1 – Описание полей Data-классов

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

Продолжение таблицы 2.1

Название	Уровень доступа	Тип данных	Свойство	Назначение
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
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. Описание выполнено с помощью диаграмм последовательностей [19].

Ниже представлены диаграммы спецификаций функций мобильного приложения.

2.4.1 Спецификация вариантов использования «Просмотреть список мероприятий»

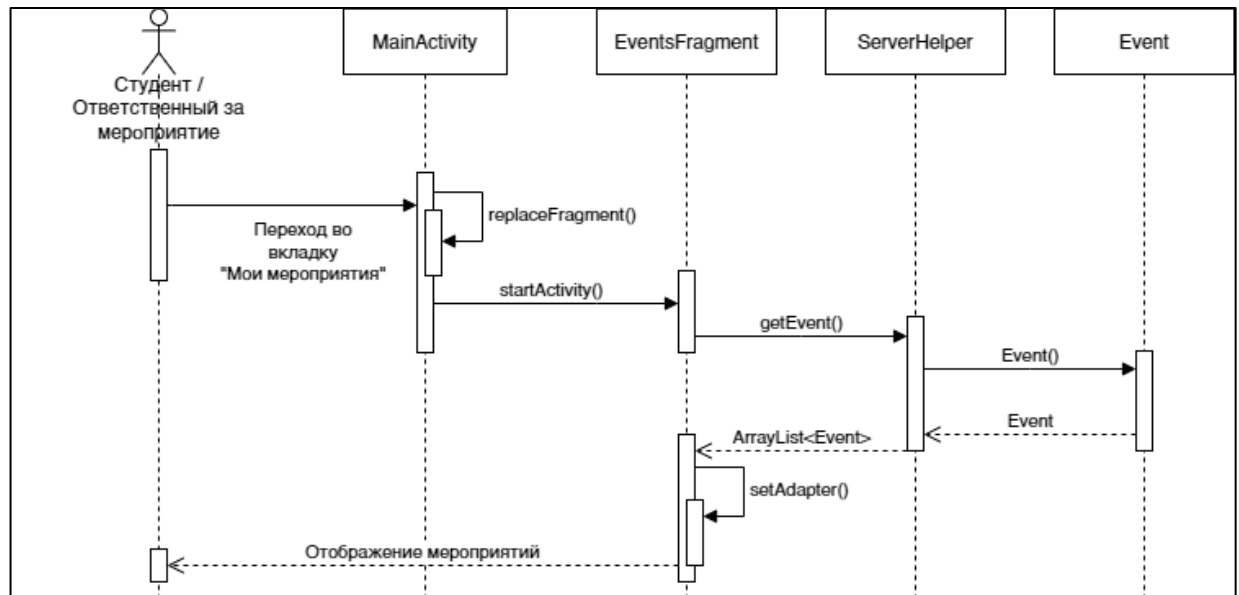


Рисунок 2.5 – «Просмотреть мероприятие»

2.4.2 Спецификация варианта использования «Зарегистрироваться на мероприятие»

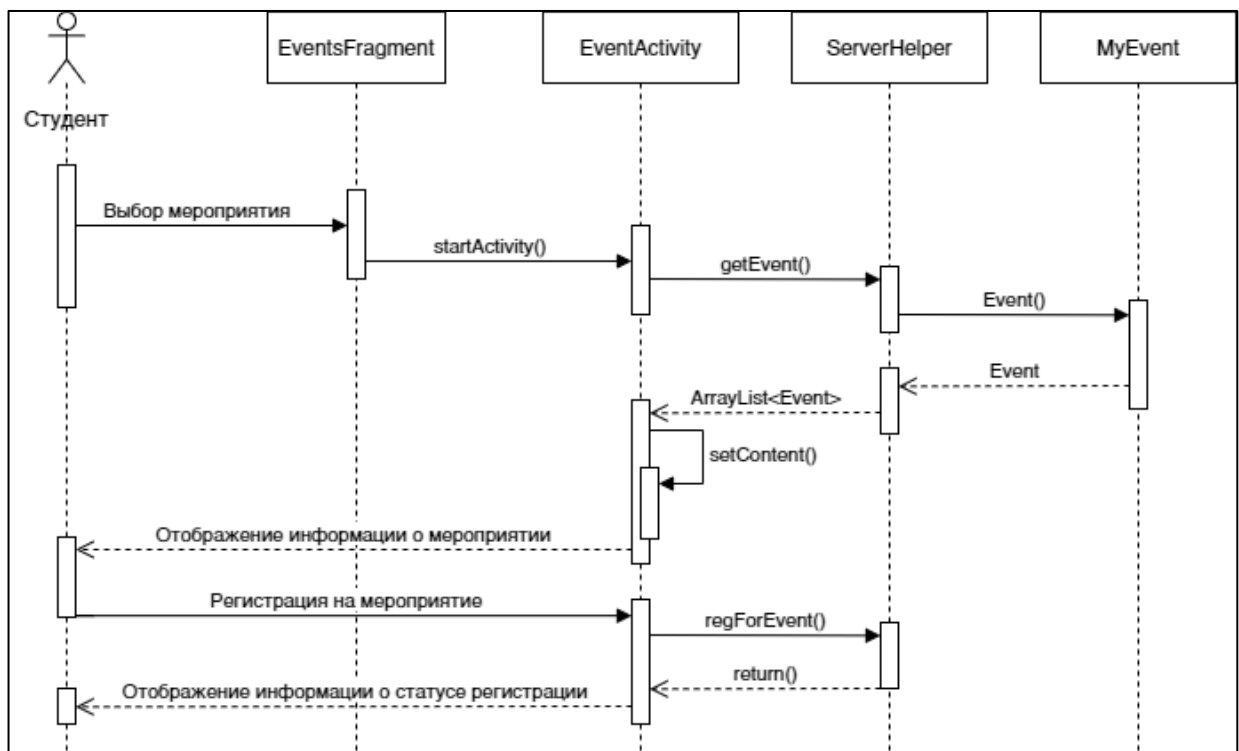


Рисунок 2.6 – «Зарегистрироваться на мероприятие»

2.4.3 Спецификация варианта использования «Подтвердить присутствие участника»

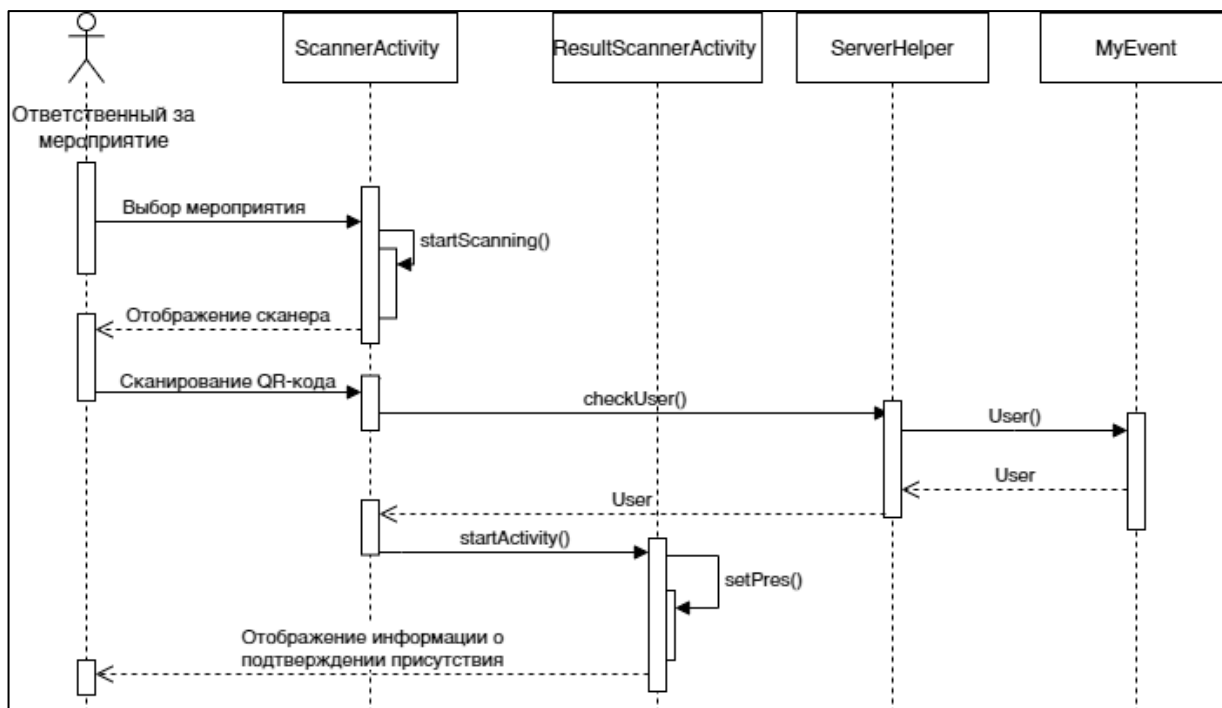


Рисунок 2.7 – «Подтвердить присутствие участника»

2.5 Проектирование графического интерфейса

Для реализации графического интерфейса спроектирован набор интерфейсных классов. На рисунке 2.8. представлена структура интерфейсных классов приложения. Зеленым цветом помечены общие для ролей «Студент» и «Ответственный за мероприятие» классы. Белым – классы интерфейса для роли «Студент», красным – для роли «Ответственный за мероприятие». Изображение содержит название классов и тип их взаимодействия. Подробное описание и представление элементов представлено на рисунках с 2.9 по 2.23 [20].

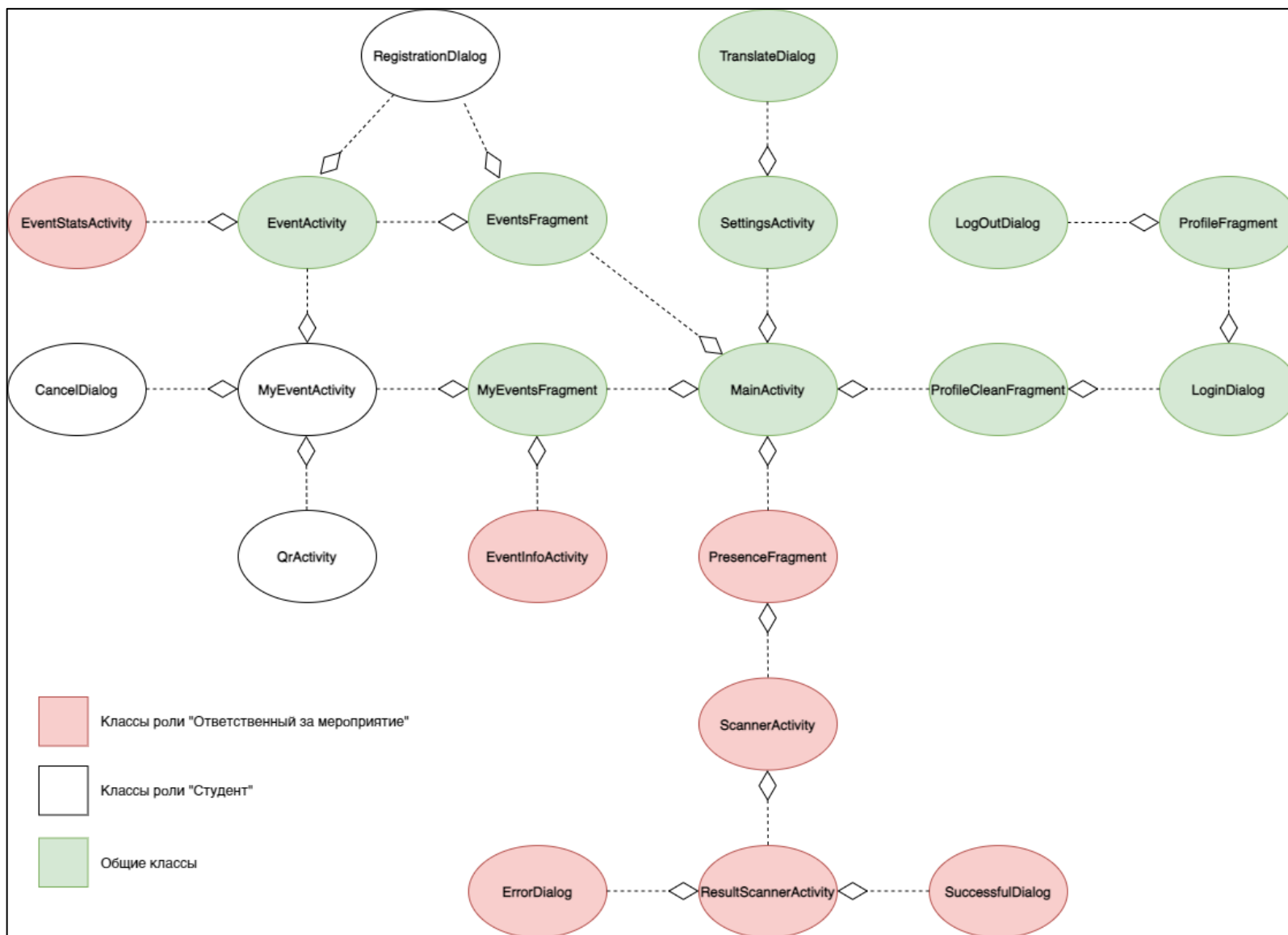


Рисунок 2.8 – Структура интерфейсных классов приложения

2.5.1 Проектирование графического интерфейса для роли «Студент»

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 пройденной авторизации в систему и при пройденной регистрации на мероприятие.

Макет данного экрана представлен на рисунке 2.9.

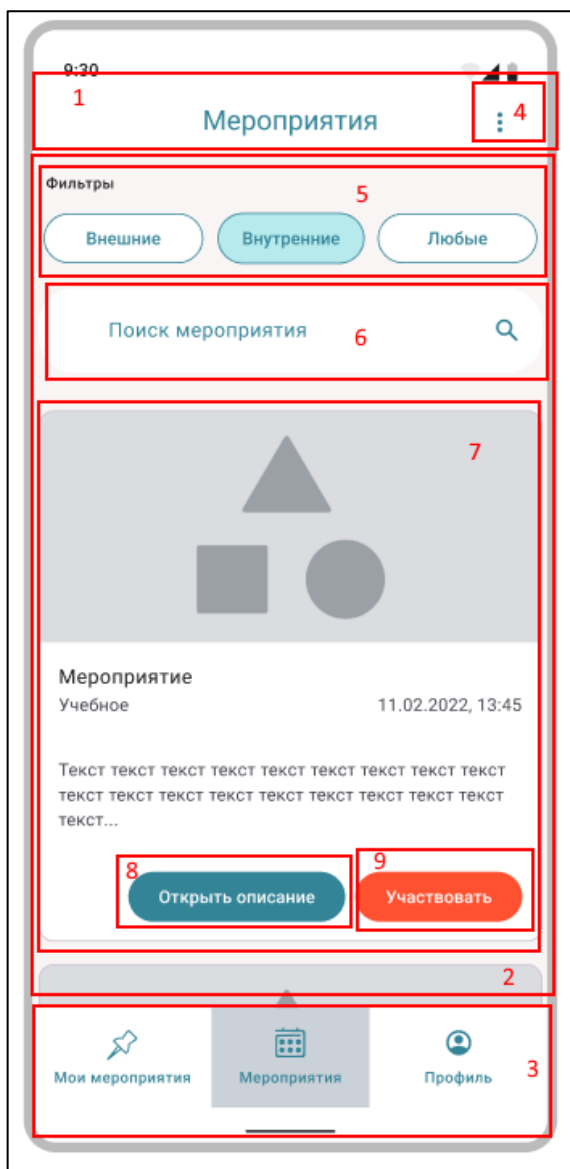


Рисунок 2.9 – Макет основного экрана приложения

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 2.12).

Макет данного экрана представлен на рисунке 2.10.

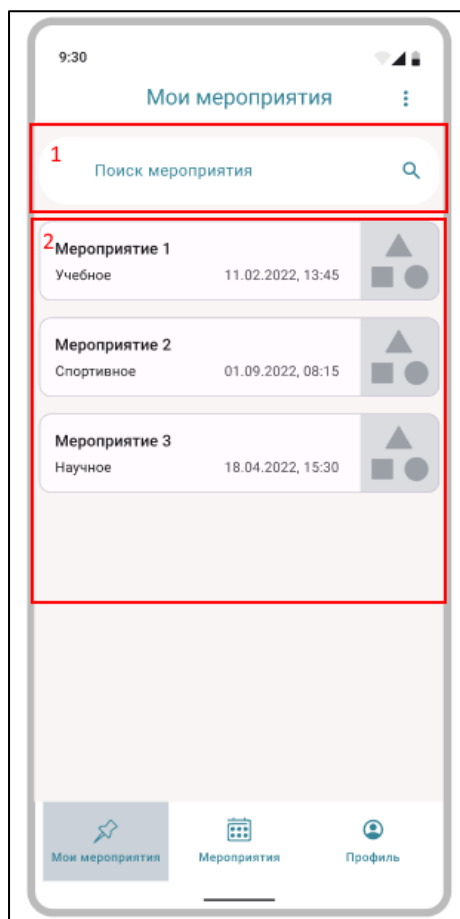


Рисунок 2.10 – Макет экрана «Мои мероприятия»

Описание экрана «Мероприятие». Данный экран вызывается при нажатии кнопки «Открыть описание» элемента 8 карточки мероприятия (см. рисунок 2.9). 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 2.12).

Макет данного экрана представлен на рисунке 2.11.

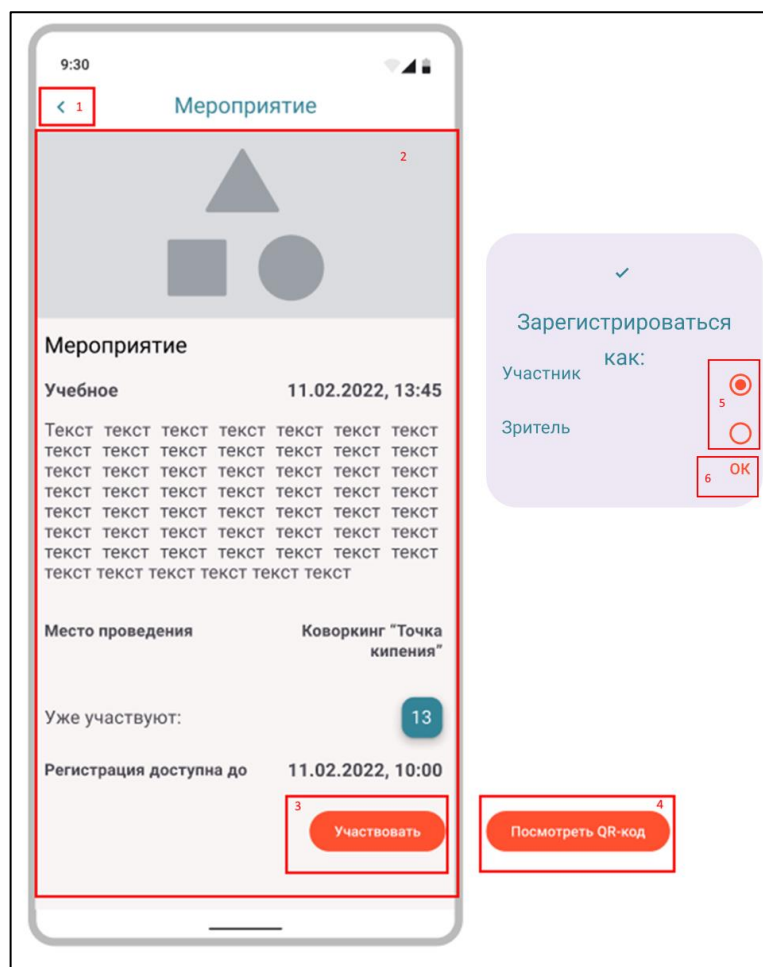


Рисунок 2.11 – Макет экрана «Мероприятие»

Описание экрана мероприятия с пройденной регистрацией. Данный экран вызывается при нажатии кнопки «Посмотреть QR-код» элемента 4 экрана «Мероприятие» (см. рисунок 2.11) или выбора компонента из списка элемента 2 экрана «Мои мероприятия» (см. рисунок 2.10). 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 2.11).

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 2.9). При нажатии на кнопку «Нет» диалоговое окно отмены регистрации закроется, экран мероприятия с пройденной регистрацией останется активным.

Макет данного экрана представлен на рисунке 2.12.

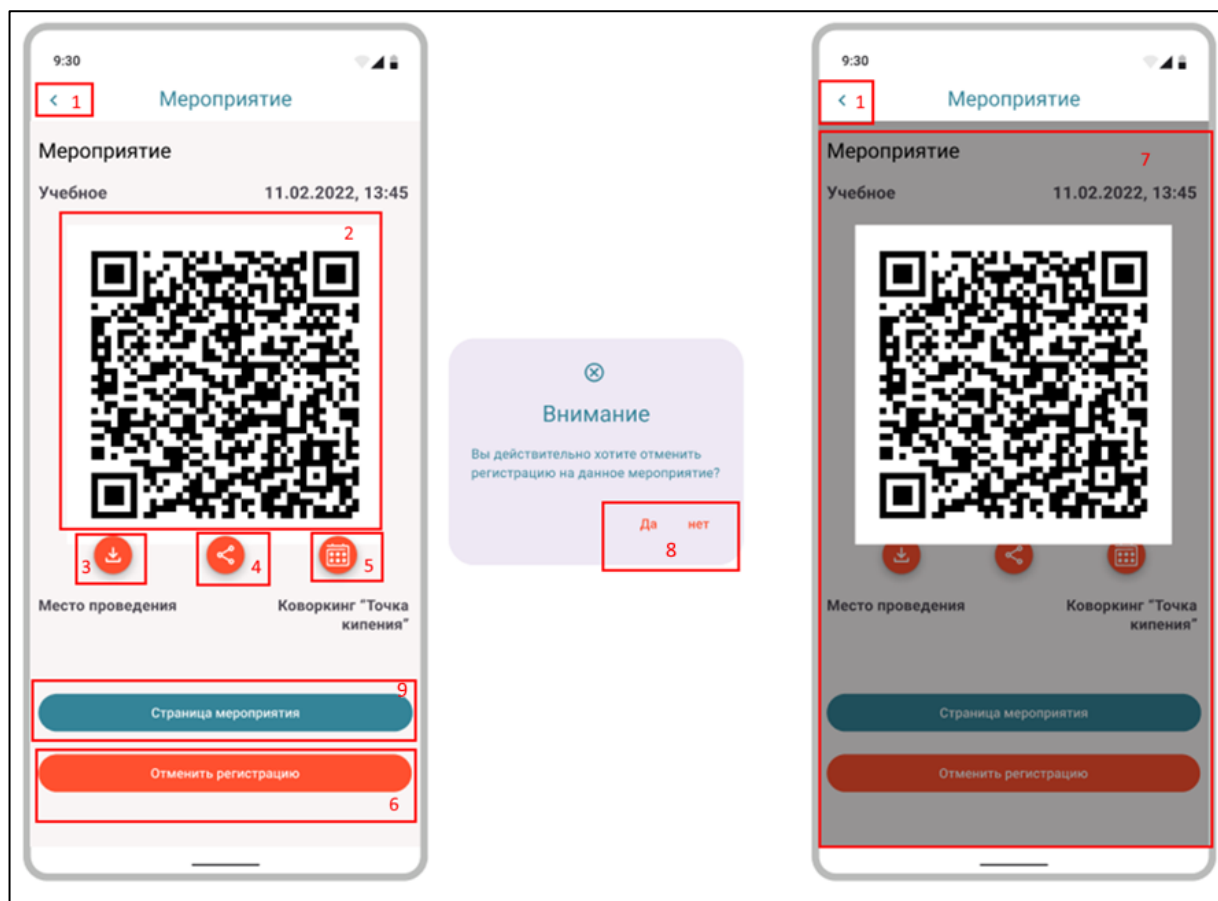


Рисунок 2.12 – Макет экрана мероприятия с пройденной регистрацией

2.5.2 Проектирование графического интерфейса для роли «Ответственный за мероприятие»

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 2.13.

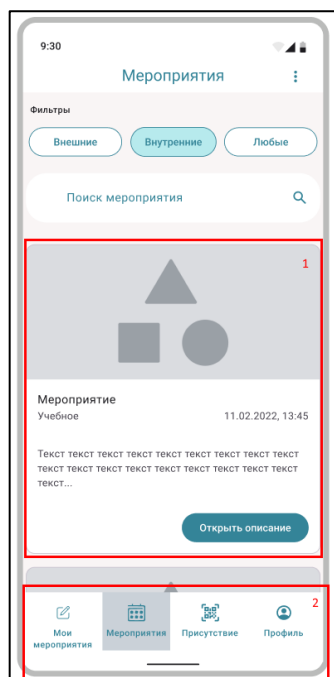


Рисунок 2.13 – Макет основного экрана приложения

Описание экрана «Мероприятие». Данный экран вызывается при нажатии кнопки «Открыть описание» кнопки «Открыть описание» карточки мероприятия (см. рисунок 2.13). 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 2.14.

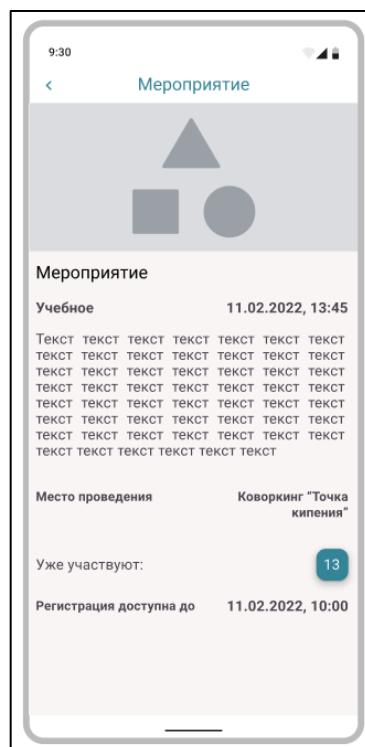


Рисунок 2.14 – Макет экрана «Мероприятие»

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 2.16).

Макет данного экрана представлен на рисунке 2.15.



Рисунок 2.15 – Макет экрана «Подтверждение присутствия»

Описание экрана сканера QR-кодов. Данный экран вызывается при выборе объекта списка в экране «Подтверждение присутствия» **элемента 2** (см. рисунок 2.15). 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 собой 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 собой 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 2.16.

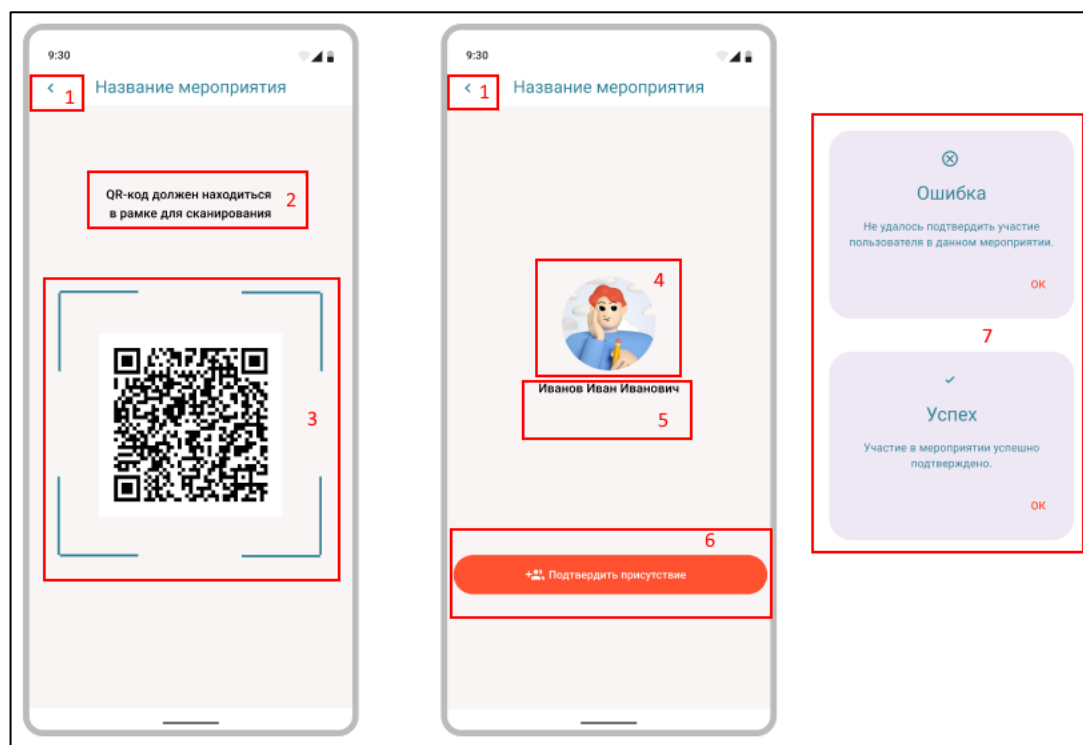


Рисунок 2.16 – Макет экрана сканера QR-кода

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:

3. **Элемент 1** – строка поиска мероприятия. Данный компонент отвечает за поиск определенного мероприятия по критерию «Название мероприятия».

4. **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 2.18).

Макет данного экрана представлен на рисунке 2.17.

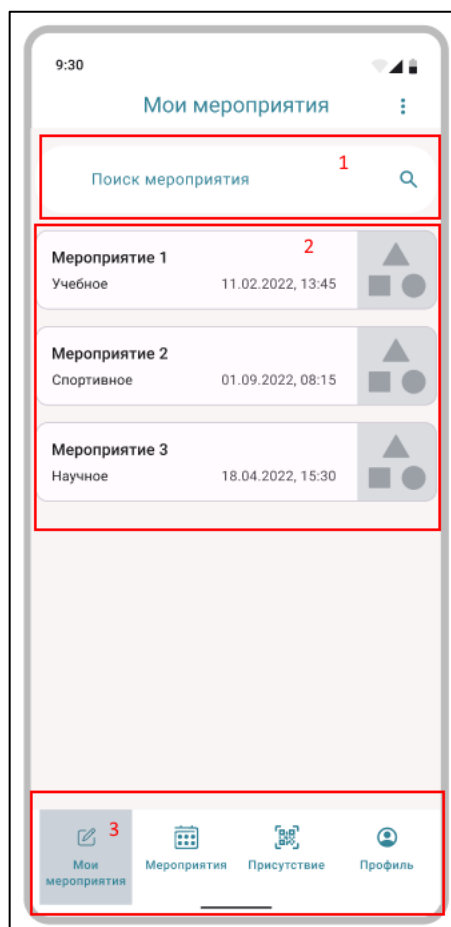


Рисунок 2.17 – Макет экрана «Мои мероприятия»

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

3. **Элемент 3** – кнопка «Статистика мероприятия». При нажатии на данный элемент открывается экран статистики мероприятия (см. рисунок 2.19).

Макет данного экрана представлен на рисунке 2.18.

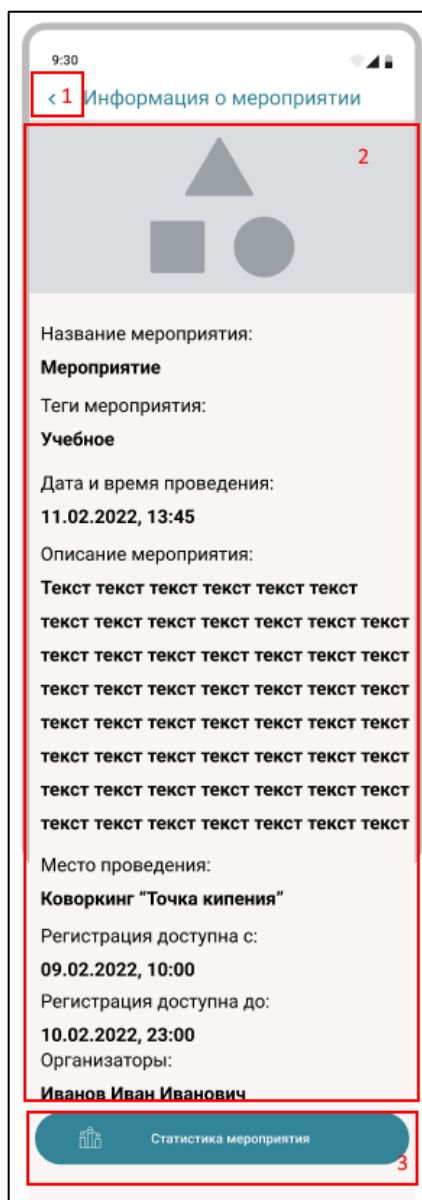


Рисунок 2.18 – Макет экрана «Информация о мероприятии»

Описание экрана статистики мероприятия. Данный экран вызывается при нажатии на кнопку «Статистика мероприятия» элемента 3 экрана «Информация о мероприятии» (см. рисунок 2.18). 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 2.19.

№	ФИО Студента	Группа	Роль	Присутствие отметил	Присутствует
1	Коновалов Егор Яковлевич	ИСТ-21-1	Зритель	Шариков П. П.	Да
2	Ильин Григорий Борисович	УУА-16-1	Участник	-	Нет
3	Осинова Анна Ильинична	ТХА6-21-1	Зритель	Шариков П. П.	Да
4	Чижова София Лукинична	УАС-19-2	Участник	-	Нет
5	Гаврилов Владимир Ильич	ЛУТм-21-1	Зритель	Шариков П. П.	Да
6	Иванова Стефания Егоровна	ИСТ-22-2	Зритель	Шариков П. П.	Да

Рисунок 2.19 – Макет экрана статистики мероприятия

2.5.3 Проектирование графического интерфейса непривязанных к конкретной роли

Описание экрана «Профиль». Данный экран вызывается при выборе объекта списка в экране «Профиль» (см. рисунок 2.13 – для ответственного, см. рисунок 2.9 – для студента). Данный экран состоит из следующего набора элементов:

1. **Элемент 1** – текстовое поле для отображения информации о необходимости авторизации для использования полной функциональности 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 2.20.

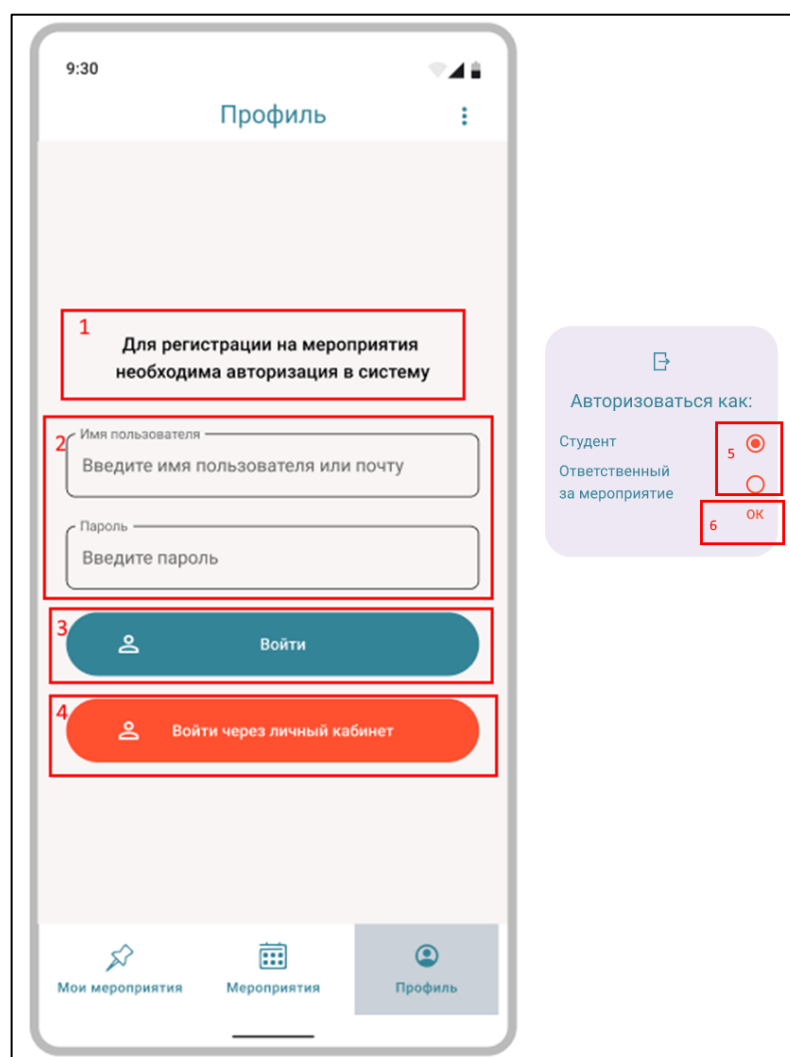


Рисунок 2.20 – Макет экрана «Профиль»

Описание экрана «Профиль» (авторизованный). Данный экран заменяет собой экран «Профиль» без авторизации (см. рисунок 2.20). 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 2.21.

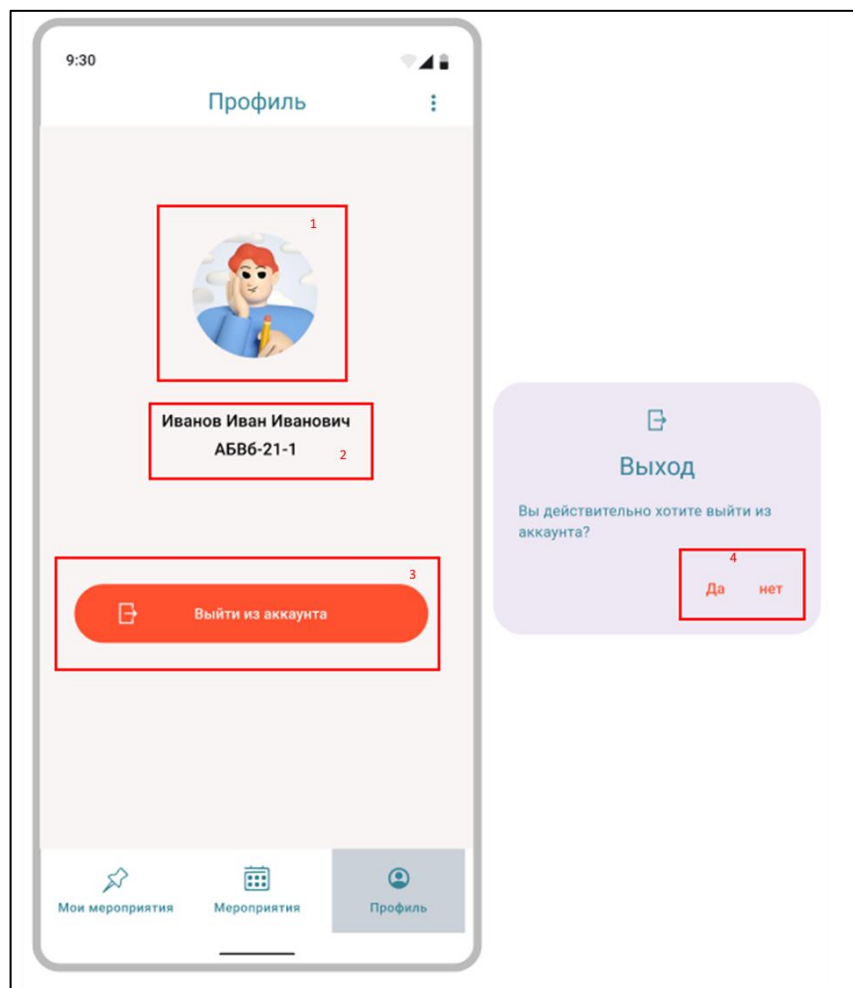


Рисунок 2.21 – Макет экрана «Профиль» (авторизованный)

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 2.23.

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 2.22.

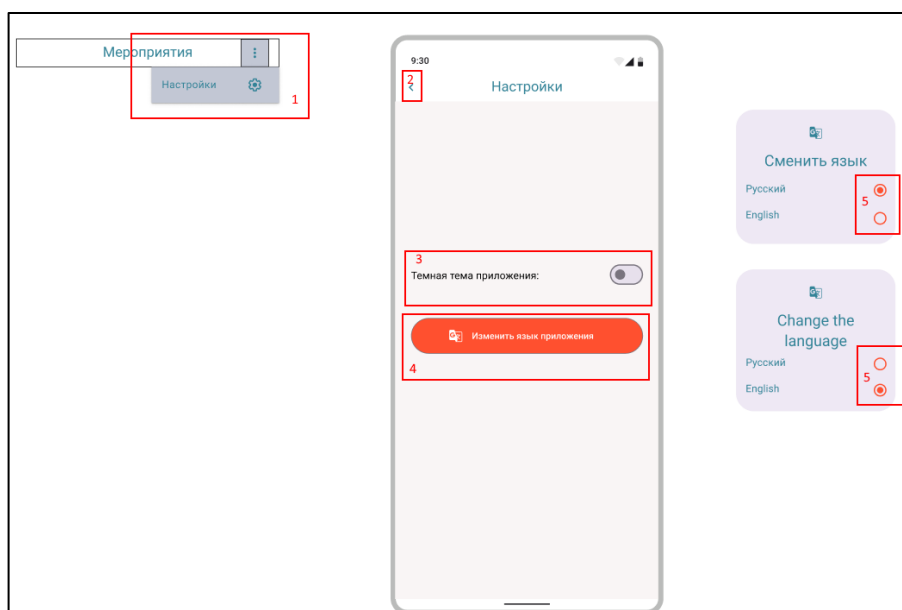


Рисунок 2.22 – Макет экрана «Настройки»

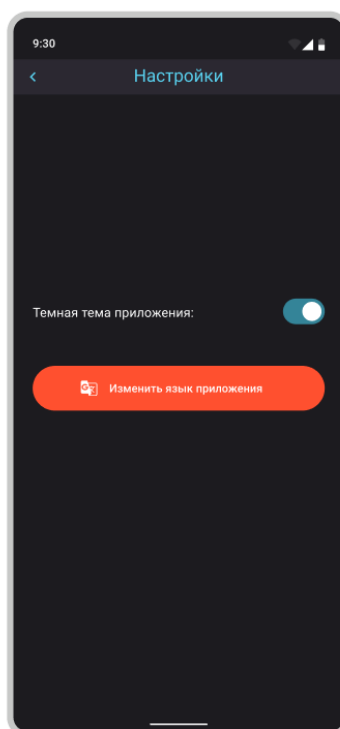


Рисунок 2.23 – Макет экрана «Настройки» (темная тема)

3 Реализация системы

Спецификация всех классов представлена в приложении Б.

3.1 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 3.1).



Рисунок 3.1 – Страница мероприятия

Фрагмент листинга кода для выполнения функции

```
private fun setContent() {  
    headerText.text = event.title  
    tagListText.text = event.tags  
    eventDateText.text = event.date  
    bodyEventText.text = event.description  
    eventLocationText.text = event.location  
    human_count_text.text = event.humanCount.toString()  
    setImages()  
    eventLayout.visibility = View.VISIBLE  
    progressBar.visibility = View.INVISIBLE  
}
```

3.2 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 3.2). Реализация функции осуществляется в классе `ServerHelper`.

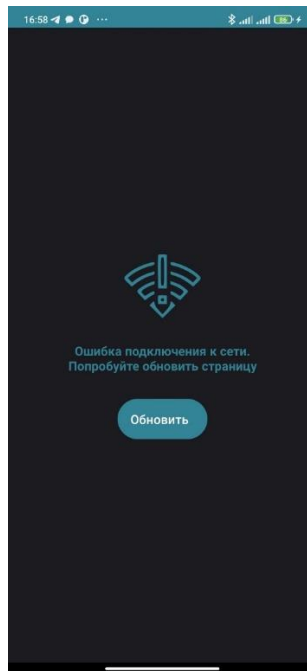


Рисунок 3.2 – Ошибка подключения к сети

Фрагмент листинга кода для выполнения 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
}
```

3.3 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 3.3). Функция реализуется в классе MyEventActivity.

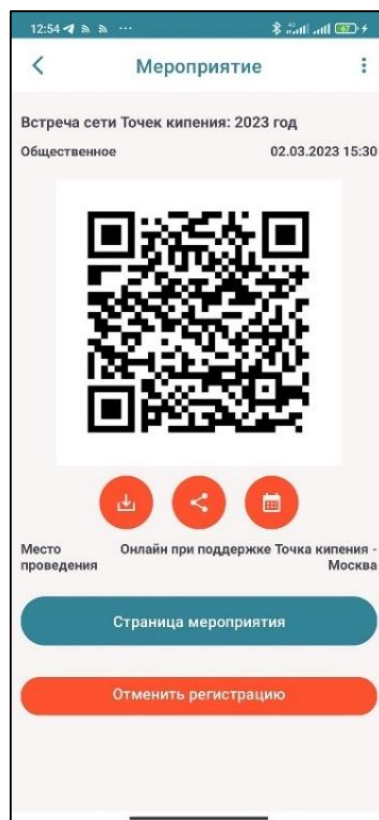


Рисунок 3.3 – Генерация QR-кода

Фрагмент листинга кода для выполнения 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)
    }
}
```

3.4 Реализация функции «Подтвердить присутствие участника»

Данная функция вызывается при считывании QR-кода студента сканером. Для реализации используется класс `ScannerActivity` (см. рисунок 3.4) и `ResultScannerActivity` (см. рисунок 3.5).



Рисунок 3.4 – Подтверждение присутствия

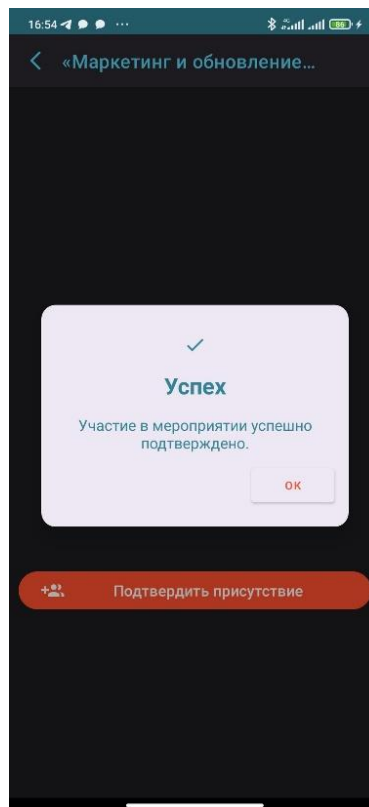


Рисунок 3.5 – Подтверждение присутствия

Фрагмент листинга кода для выполнения функции

```
private fun startScanning() {
    // Callbacks
    codeScanner.decodeCallback = DecodeCallback {
        runOnUiThread {
            if(serverHelper.checkUser(it.text, event.id) == true) {
                val i = Intent(this, ResultScannerActivity::class.java)
                i.putExtra("user", it.text)
            }
        }
    }
}
```



```

        i.putExtra("event_title", event.eventTitle)
        startActivity(i)
    }
    else{
        Toast.makeText(this, "QR-код не распознан...",
            Toast.LENGTH_SHORT).show()
    }
}
}
codeScanner errorCallback = ErrorCallback { // or ErrorCallback.SUPPRESS
    runOnUiThread {
        Toast.makeText(this, "Camera initialization error: ${it.message}",
            Toast.LENGTH_LONG).show()
    }
}
}
}

```

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

Мобильное приложение сохраняет необходимую информацию о мероприятии в виде pdf-файла (см. рисунок 3.6). Функция выполняется в классе MyEventActivity, запуск осуществляется с помощью кнопки сохранения.



Рисунок 3.6 – Pdf-файл информации о мероприятии

3.6 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 3.7). Функция реализуется в классе MyEventActivity.

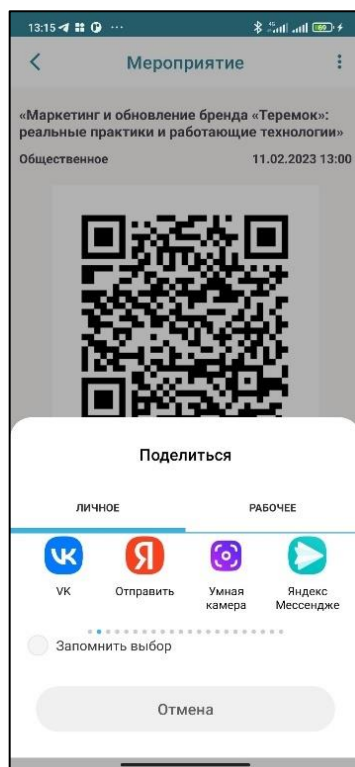


Рисунок 3.7 – Отправка pdf-файла

3.7 Реализация функции «Добавить мероприятие в календарь смартфона»

Приложение подготавливает данные о мероприятии и вызывает Intent выбора приложения-календаря (см. рисунок 3.8.). Выбранное приложение получает набор данных и формирует новое событие в день проведения мероприятия (см. рисунок 3.9). Функция реализуется в классе MyEventActivity.

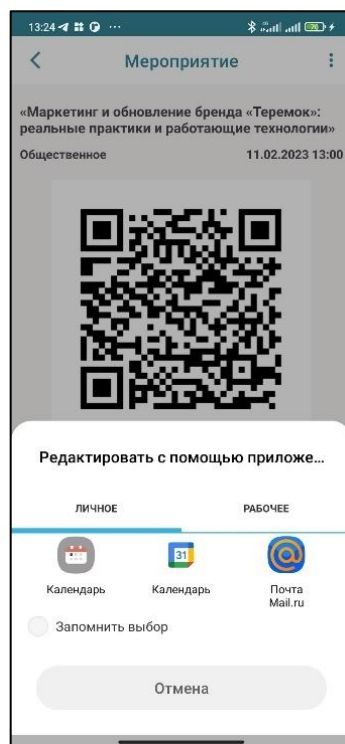


Рисунок 3.8 – Вызов приложения-календаря устройства

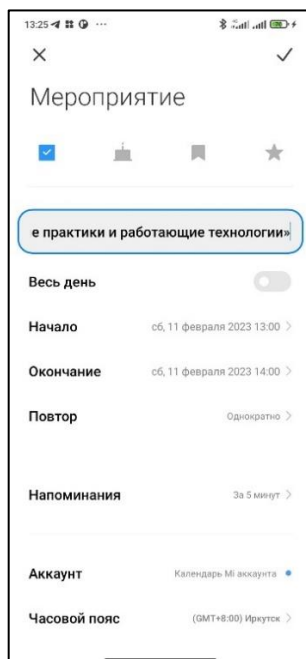


Рисунок 3.9 – Создание нового события в календаре

3.8 Реализация функции «Отменить регистрацию на мероприятие»

При нажатии на соответствующую кнопку на странице мероприятия вызывается диалог-предупреждение об отмене регистрации на мероприятии (см. рисунок 3.10).

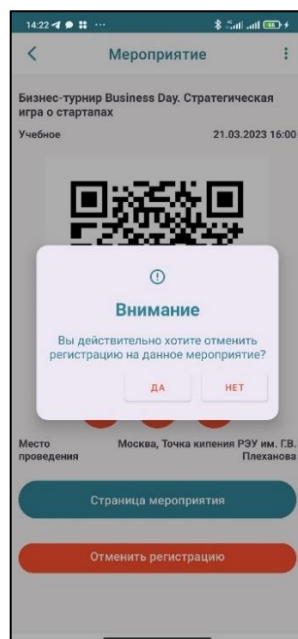


Рисунок 3.10 – Отмена регистрации на мероприятие

3.9 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 3.11).

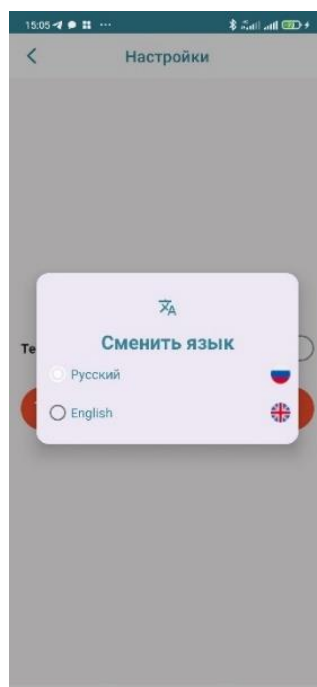


Рисунок 3.11 – Диалоговое окно смены языка

3.10 Реализация функции смены графической темы приложения

Смена графической темы приложения осуществляется за счет изменения состояния переключателя (см. рисунок 3.12). Функция реализована в классе SharedPrefs, вызов диалогового окна в классе SettingsActivity.

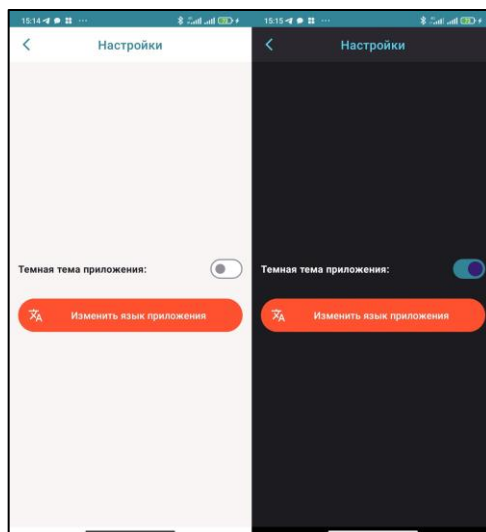


Рисунок 3.12 – Смена графической темы

3.11 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 3.13). Функция реализована в классе SharedPrefs, вызов диалогового окна в классе ProfileCleanFragment.

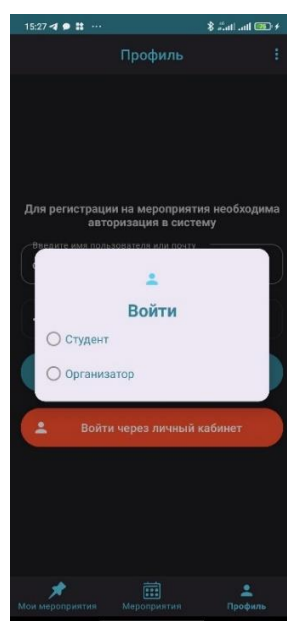


Рисунок 3.13– Авторизация в приложении

3.12 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 3.14)

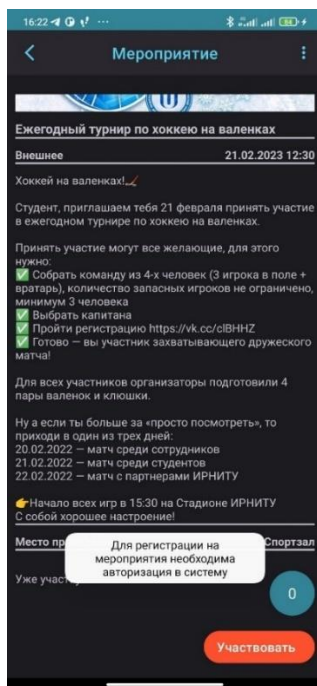


Рисунок 3.14– Регистрация на мероприятие

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).

№	ФИО Студента	Группа	Роль	Присутствие отметил	Присутствует
1	Иванов Иван Иванович	ИМп-12-2	участник	Иванов А.И	Да
2	Иванов Иван Иванович	ИМп-12-2	участник	Иванов А.И	Да
3	Иванов Иван Иванович	ИМп-12-2	участник	Иванов А.И	Да
4	Иванов Иван Иванович	ИМп-12-2	участник	Иванов А.И	Да

Рисунок 3.15 – Статистика мероприятия

4 Тестирование системы

Тестовое приложение было представлено фокус-группе, состоящей из студентов ИРНИТУ, которые получили сборку для проведения тестирования и анкету для предоставления обратной связи. Целью тестирования было определение уровня работоспособности приложения на различных типах устройств и версиях операционной системы Android, а также оценка функциональности, стабильности и удобства использования.

Участникам фокус-группы было предложено заполнить анкету, в которой они должны были указать тип устройства, версию Android, обозначить выявленные баги или ошибки при использовании, указать уровень стабильности приложения при слабом подключении к сети, а также оценить дизайн и интуитивность взаимодействия с графическим интерфейсом. Анкета опроса представлена в приложении В.

После сбора результатов опроса была проведена аналитика данных, позволившая получить подробную сводку по каждому из вопросов. Исходя из этой подробной сводки, можно отметить следующее:

1. Подавляющее число студентов проводило тестирование приложения на смартфоне (см. рисунок 4.1).

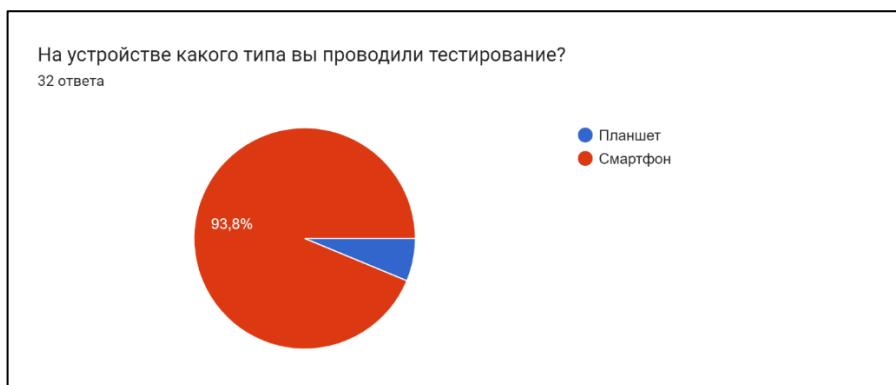


Рисунок 4.1 – Тип устройства

2. Большинство пользователей использовали смартфоны с операционной системой Android 10, но также были ответы, связанные с более старыми (Android 5.0 Lollipop, Android 6.0 Marshmallow) и новыми версиями (см. рисунок 4.2).

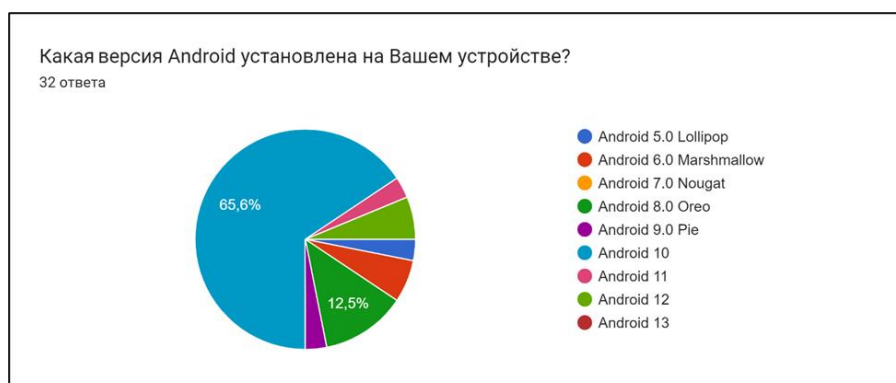


Рисунок 4.2 – Версия Android на устройстве

3. Некоторые пользователи столкнулись с багами, ошибками или аварийным завершением работы приложения, но большинство отметило, что не встретилось с данными проблемами (см. рисунок 4.3).

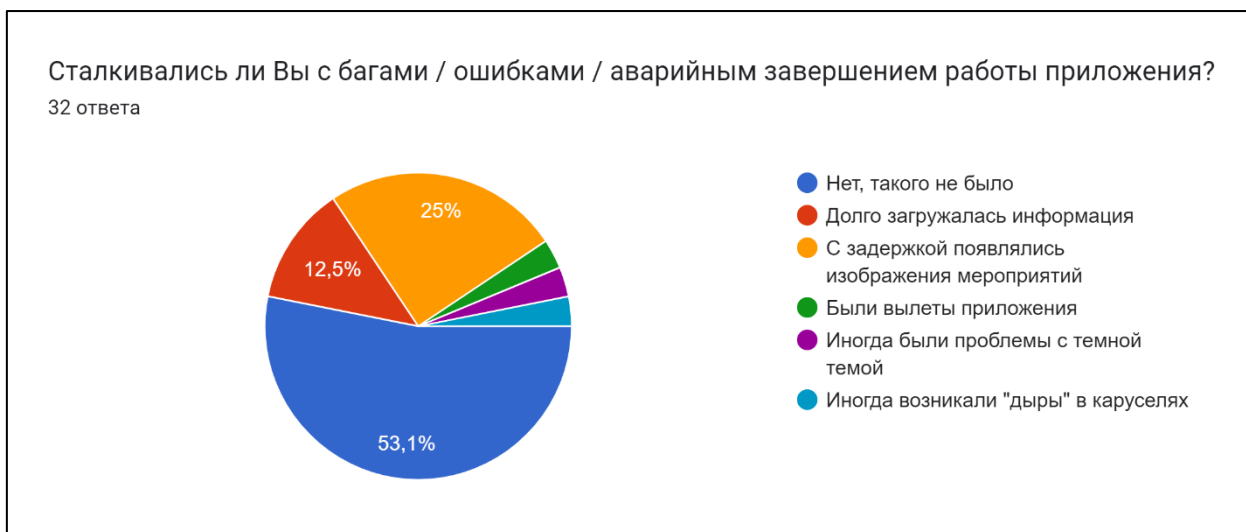


Рисунок 4.3 – Наличие ошибок в приложении

4. Большинство пользователей отметили, что приложение стабильно работает даже при слабом подключении к сети (см. рисунок 4.4).

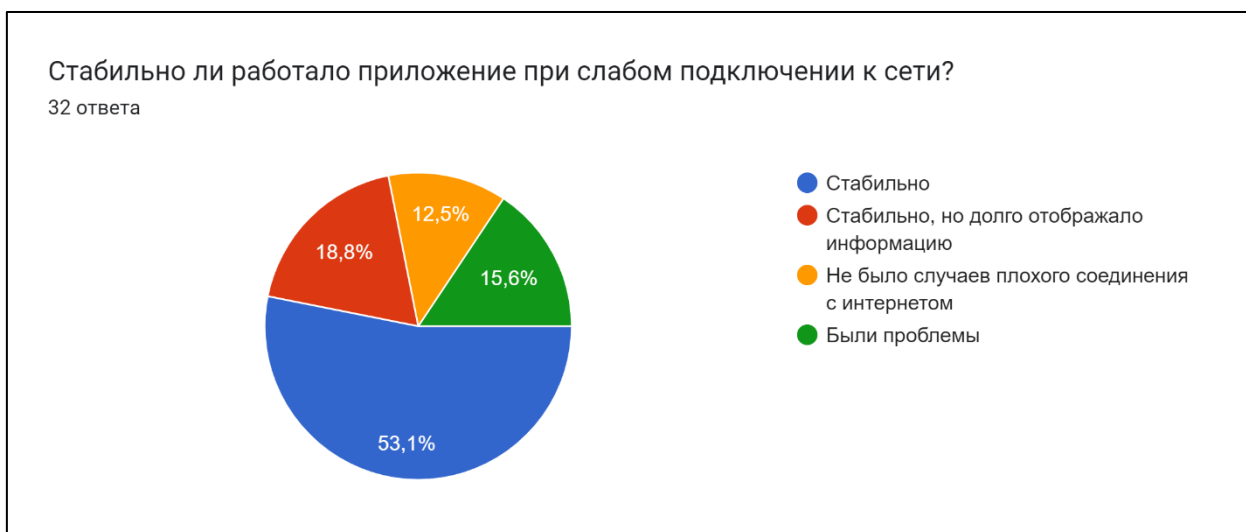


Рисунок 4.4 – Стабильность работы при слабом подключении к сети

5. Оценки дизайна приложения варьировались от 1 до 5, но наибольшее количество пользователей оценило дизайн на уровне 4 или 5 из 5 (см. рисунок 4.5).

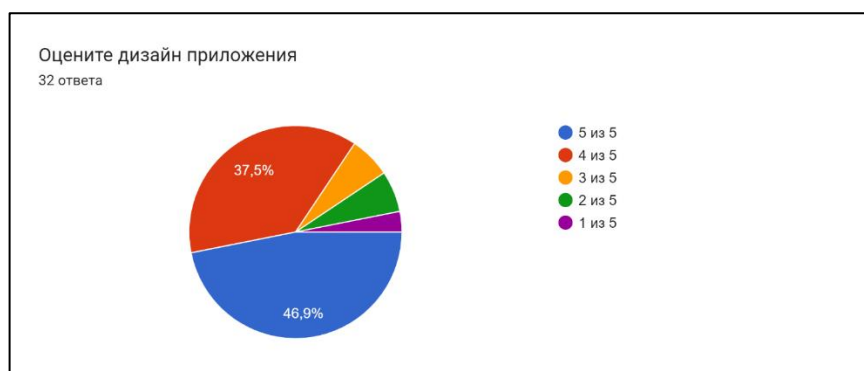


Рисунок 4.5 – Дизайн графического интерфейса

6. Оценки интуитивности взаимодействия с графическим интерфейсом также были разнообразными, но большинство пользователей оценило данный аспект на уровне 4 или 5 из 5 (см. рисунок 4.6).

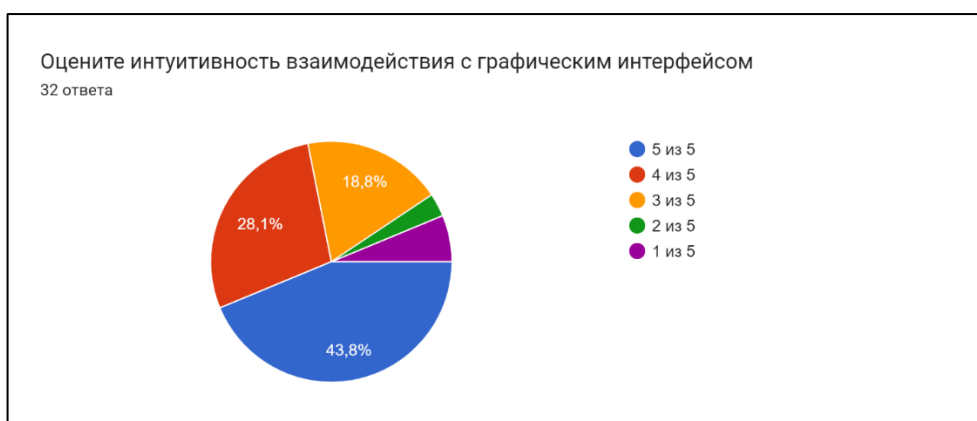


Рисунок 4.6 – Интуитивность взаимодействия

После анализа результатов опроса, были сделаны следующие выводы:

1. Загрузка информации – некоторые пользователи сообщили о длительной загрузке информации при использовании приложения.
2. Баги и ошибки – некоторые участники столкнулись с багами, ошибками или аварийным завершением работы приложения.
3. Работа при слабом подключении к сети – большинство пользователей отметили, что приложение работало стабильно даже при слабом подключении к сети, но некоторые все же сталкивались с проблемами.
4. Оценка дизайна – оценки дизайна приложения варьировались, но в целом пользователи высоко оценили его.
5. Интуитивность взаимодействия – оценки интуитивности взаимодействия с графическим интерфейсом также были разнообразными, но большинство пользователей оценило данный аспект положительно.

Эти выводы позволяют понять основные сильные и слабые стороны приложения, которые могут быть учтены при дальнейшей разработке и улучшении.

5 Экономическое обоснование

5.1 Расчет единовременных затрат разработчика

Единовременные затраты разработчика программного обеспечения могут быть рассчитаны с учетом множества факторов. Важно учесть количество часов, которое требуется для выполнения каждого этапов создания приложения и затраченную на их выполнение трудоемкость. Данные этапы включают в себя: анализ требований, проектирование архитектуры, разработку кода и тестирование итогового приложения. В таблице 5.1 представлена оценка трудоемкости этапов разработки мобильного приложения.

Таблица 5.1 – Стадии разработки мобильного 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

Затраты при разработке мобильного приложения для выпускной квалификационной работы отнесены к единовременным затратам.

5.2 Расчет затрат по экономическим элементам

К затратам по экономическим элементам на создание программного обеспечения относятся следующие составляющие:

1. Материальные затраты – включают стоимость всех физических компонентов, необходимых для создания приложения. В эту категорию входят компьютеры, серверы, сетевое оборудование, периферийные устройства (например, принтеры, сканеры), коммуникационные кабели и другое оборудование, которое требуется для функционирования системы. В таблице 5.2 представлены материальные затраты на разработку.

Таблица 5.2 – Материальные затраты

Наименование	Цена	Количество	Стоимость
Набор канцелярии	240 руб	1	240
Печать (1 лист)	5 руб	100	500
Пакет интернета	250 руб	1	250
Итого			990

2. Затраты на электроэнергию – создание и поддержание информационной системы требует электроэнергии для питания всех компонентов, включая компьютеры, мониторы и другие устройства. Сумма этих затрат зависит от времени работы и стоимости электроэнергии.

В процессе проектирования и разработки мобильного приложения для работы использовался ноутбук Acer Aspire A715-71G. Потребление электроэнергии ноутбуком – 0,135 кВт/ч.

Стоимость 1кВт электроэнергии составляет 1,42 руб.

Стоимость машинного часа $C_{мч}$ работы рассчитывается по формуле (1):

$$C_{мч} = P_э \times C_э = 0,135 \frac{\text{кВт}}{\text{час}} \times 1,42 \frac{\text{руб.}}{\text{кВт}} = 0.1917 \frac{\text{руб.}}{\text{час}}, \quad (1)$$

где $P_э$ – потребление электроэнергии, кВт/ч;

$C_э$ – стоимость электроэнергии, рубль/кВт.

Время работы ЭВМ, $T_{эвм}$ рассчитывается по формуле (2):

$$T_{эвм} = D \times H = 36 \text{ сут.} \times 6 \text{ час} = 216 \text{ час.}, \quad (2)$$

где D – количество рабочих дней, сутки;

H – количество часов, час.

Себестоимость электроэнергии рассчитывается по формуле (3):

$$C_{эл} = T_{эвм} \times C_{мч} = 216 \text{ час} \times 0,1917 \frac{\text{руб.}}{\text{час}} = 41.4072 \text{ руб.}, \quad (3)$$

где $T_{эвм}$ – время работы ЭВМ, час;

$C_{мч}$ – стоимость машинного часа, руб./час.

По данным сайта Job Filter средняя заработная плата программиста-стажера составляет 40000 рублей в месяц [21]. Затраты на оплату труда $C_{от}$ рассчитываются по формуле (4):

$$C_{от} = 3П_{мес} \times T = 40000 \text{ руб.} \times 2 \text{ мес.} = 80000 \text{ руб.}, \quad (4)$$

где $3П_{мес}$ – ежемесячная заработная плата, рубль;

T – количество рабочего времени, месяц.

По результатам вышеприведённых расчётов составлена смета затрат на разработку мобильного приложения (см. таблицу 5.3).

Таблица 5.3 – Смета затрат

Элемент затрат	Сметная стоимость, руб
Материальные затраты	990 руб
Затраты на электроэнергию	41.4072 руб
Затраты на оплату труда	80000 руб
Итого	81031.4072 руб

5.3 Расчет экономической выгоды по времени

Для расчета временной выгоды необходимо вычислить временные затраты ответственного за мероприятие на подтверждение присутствия студента. При входе студента ответственный за мероприятие сверяет данные участника со списком зарегистрированных студентов. При наличии студента в списке ответственный просит участника расписаться в нем напротив своей фамилии. Данный процесс занимает в среднем до 5-ти минут в зависимости от количества пришедших участников и ответственных за мероприятие. Зрители, пришедшие посмотреть на мероприятие, не учитываются. Количество участников каждого мероприятия является индивидуальным значением. Среднее число участников мероприятия равняется 30-ти.

Для нахождения времени, затрачиваемого на подтверждения присутствия студентов рассчитывается по формуле (5):

$$T_p = \frac{K_{уч} \times T_{подтв}}{K_{отв}}, \quad (5)$$

где $K_{уч}$ – количество участников, единица;

$T_{подтв}$ – время подтверждения присутствия, минута;

$K_{отв}$ – количество ответственных на мероприятии, единица.

На каждом мероприятии находятся минимум 3 человека, занимающиеся подтверждением присутствия студентов. Соответственно время, затраченное на регистрацию присутствия студентов рассчитывается по формуле (5):

$$T_{p1} = \frac{30 \text{ ед.} \times 5 \text{ мин.}}{3 \text{ ед.}} = 50 \text{ мин.}$$

С помощью мольного приложения, в свою очередь, ответственный за мероприятие потратит приблизительно 30 секунд или же 0.5 мин на подтверждение присутствия студента, с учетом обмена данными с сервером. Дополнительно становится возможность учета не только участников мероприятия, но и зрителей.

В таком случае время, затрачиваемое тремя ответственными на подтверждение 30-ти студентов рассчитывается по формуле (5):

$$T_{p2} = \frac{30 \text{ ед.} \times 0.5 \text{ мин.}}{3 \text{ ед.}} = 5 \text{ мин.}$$

Следовательно, временная выгода (ΔT) от внедрения приложения рассчитывается по формуле (6):

$$\Delta T = T_{p1} - T_{p2} = 50 \text{ мин.} - 5 \text{ мин.} = 45 \text{ мин.}, \quad (6)$$

где T_{p1} – время работы ответственных при ручном подтверждении, минута;

T_{p2} – время работы ответственных при использовании приложения, минута.

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 [22].

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 [23]. При построении здания учитывались требования СНиП II-7-81, включающие нормы и правила строительства зданий в сейсмических районах [24]. Соблюдение данных норм обусловлено нахождением Иркутской области в сейсмическом районе акватории озера Байкал. Схема рабочего помещения представлена на рисунке 6.1.

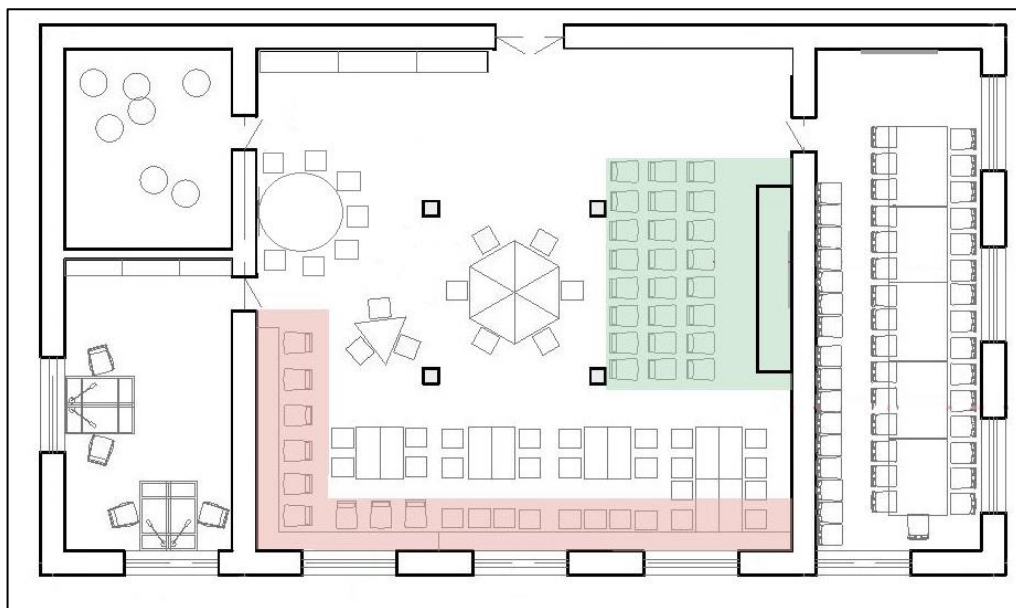


Рисунок 6.1 – Схема рабочего помещения

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 [25]. 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 [26].

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.

В соответствии с руководящими документами Р 2.2.2006-05 [27] и ГОСТ 12.0.003-2015 [28] 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
1	2	3	4
Вредные факторы			
Повышенная или пониженная освещенность	Неисправные осветительные приборы, загрязненность окон	$E_n = 1,275 \%$ $E_n = 300$ лк СанПиН 1.2.3685-21 [29] СанПиН 52.13330.2016 [30]	Заменить осветительные приборы и очистить окна от загрязнений
Повышенный уровень интенсивности звука	Кулер, оргтехника, электромагнитный шум. Внешний шум $L > 80-120$ дБ	$L = 50$ дБ СанПиН 1.2.3685-21 [29]	Монтаж звукоизоляции, замена оборудования на менее шумное
Повышенная или пониженная интенсивность электромагнитного излучения	Компьютер, оргтехника	В диапазоне 5Гц – 2кГц: $E = 25$, В/м. $B = 250$ нТл, $H = 2,5$ В/м СанПиН 1.2.3685-21 [29]	Регламентированный перерыв, изоляционные экраны
Отклонение параметра микроклимата от нормируемых значений	Отсутствие вентиляции, систем кондиционирования, наличие щелей в окнах, неисправность отопления	Категории работ 1b Холодный период: $t = 21-23^\circ\text{C}$ $\varphi = 60-40\%$ $v = 0,1$ м/с, t поверхностей = $20-24^\circ\text{C}$. Тёплый период: $t = 22-24^\circ\text{C}$ $\varphi = 60-40\%$, $v = 0,1$ м/с t , поверхностей = $21-25^\circ\text{C}$. СанПиН 1.2.3685-21 [29]	Установка дополнительных обогревателей и кондиционеров, вентиляторов
Повышенный показатель напряженности трудового процесса	Плотность сигналов (световых, звуковых) и сообщений в среднем за 1 час работы	допустимо 76 – 175 Р 2.2.2006 – 05 [27]	Использовать средства индивидуальной защиты: специальные очки для работы за компьютером

Продолжение таблицы 6.1

1	2	3	4
Повышенный показатель напряженности трудового процесса	Стереотипные рабочие движения. При локальной нагрузке (с участием мышц кистей и пальцев рук)	Оптимально до 20000 однотипных движений за смену Р 2.2.2006 – 05 [27]	Смена позы, регламентированный перерыв
	Рабочая поза	Удобная поза, возможность смены рабочего положения тела. Нахождение в позе стоя до 40% времени смены Р 2.2.2006 – 05 [27]	Смена позы и регламентированный перерыв
	Сенсорные нагрузки: плотность сигналов и сообщений в среднем за 1 час работы	допустимо 76 – 175 оптимально до 75 Р 2.2.2006 – 05 [27]	Использовать средства индивидуальной защиты: специальные очки для работы за компьютером
	Сложность работы, ответственность	Перерывы регламентированы, достаточной продолжительности: 7% и более рабочего времени Р 2.2.2006 – 05 [27]	Наличие регламентированных перерывов и их продолжительность
Опасные факторы			
Опасность поражения электрическим током	Оголенные провода, неисправности электропроводки и оборудования	U=220В I=0,6-1,5мА (ток самосвобождения) ГОСТ 12.1.019-2017 [31]	Наличие изоляций токопроводящих поверхностей. Наличие защитного 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 [32]	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 6.2 продемонстрирована схема нормативных показателей высоты рабочего места.

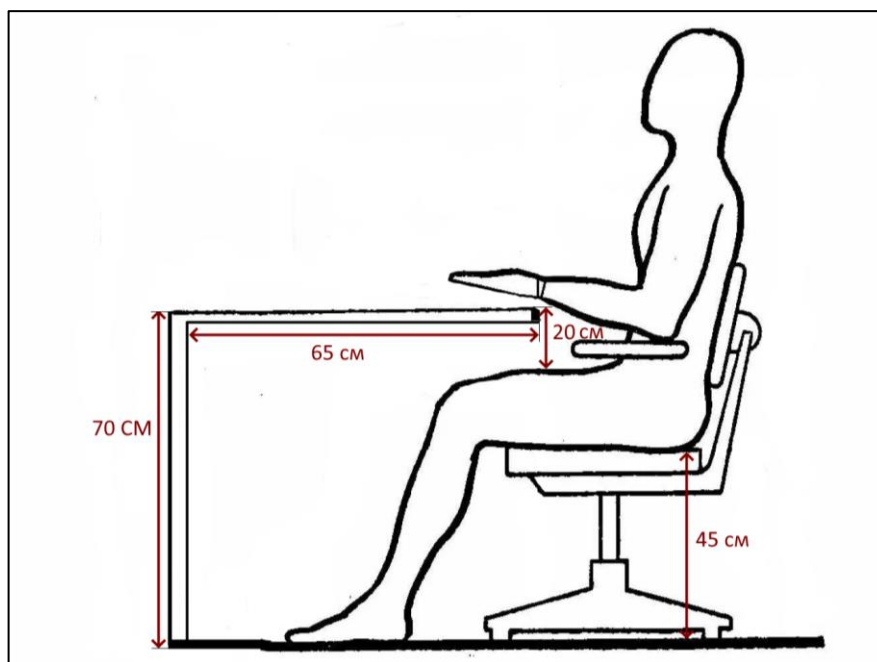


Рисунок 6.2 – Схема нормированных показателей рабочего места

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 поверхности	Высота сидения	Пространство для ног	Высота рабочей поверхности	Высота сидения	Пространство 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.

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 [31] представлена на рисунке 6.3.

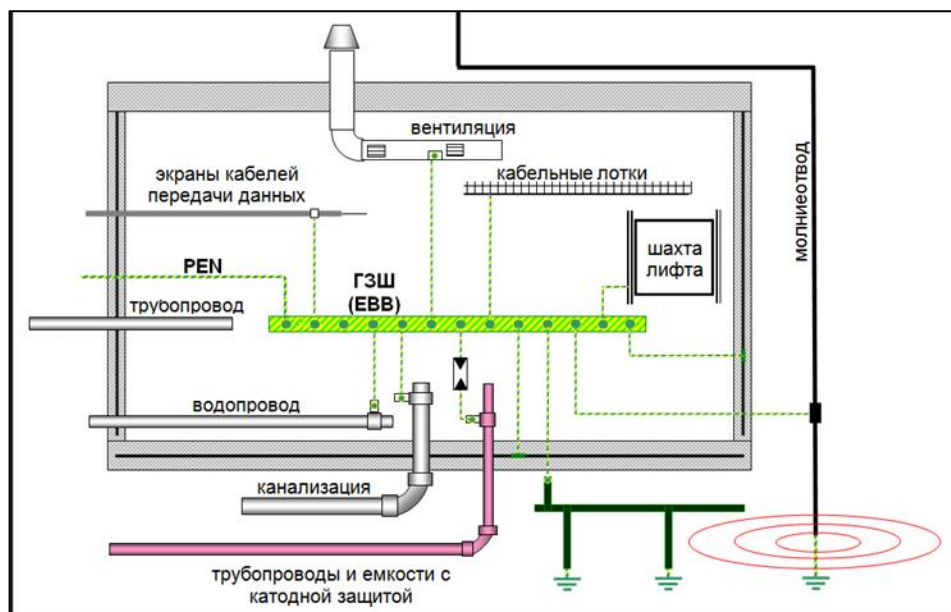


Рисунок 6.3 – Схема заземления

Рабочее помещение соответствует стандартам, прописанным в ГОСТ 12.1.019-2017 [31], 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 [32]. 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 6.4. Принцип работы датчиков дыма заключается в том, что посылаемый луч в приборе при наличии дыма рассеивается, сигнализируя системе о наличии задымления.



Рисунок 6.4 – Датчик дыма «ИПД-3.2»

В коридорах университета расположены пожарные щиты, 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 6.5 проиллюстрирован один из установленных щитов.



Рисунок 6.5 – Пожарный щиток

На рисунке 6.6 изображен огнетушитель ОУ-2, на рисунке изображен пожарный рукав, которые установлены в вышеописанные пожарные щитки.



Рисунок 6.6 – 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 6.7 изображена схема эвакуации из помещения.

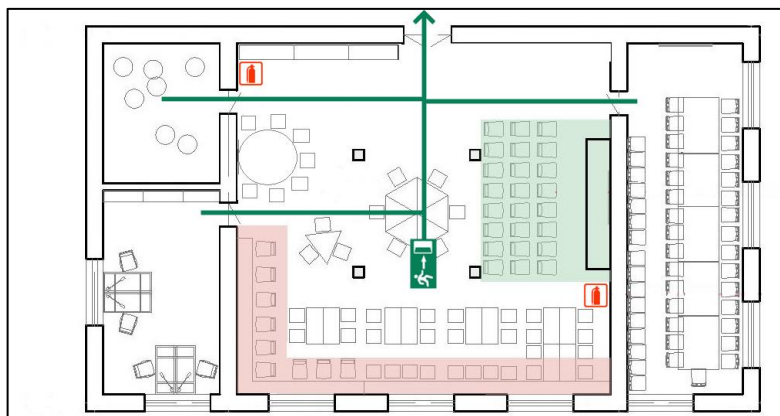


Рисунок 6.7 – Схема эвакуации из помещения

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».



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

В результате выполнения выпускной квалификационной работы был проведен анализ проблем учета посещаемости мероприятий студентами ИРНИТУ, поставлены основные цели и задачи.

В ходе выполнения поставленных задач был произведен анализ проблем процесса регистрации студентов на мероприятия и последующего подтверждения фактического присутствия.

Разработанное мобильное приложение является частью информационной системы «Рейтинг студента».

В результате были решены следующие задачи:

1. Изучен процесс регистрации студентов на мероприятия.
2. Выполнен анализ предметной области.
3. Сформулирован перечень требований для мобильного приложения.
4. Выполнено проектирование мобильного приложения.
5. Реализовано мобильное приложение, состоящее из двух модулей: для ролей «Студент» и «Ответственный за мероприятие».

Для студентов приложение предоставляет возможность просматривать информации о мероприятии, регистрироваться на них и получать уникальные QR-коды участников мероприятия и зрителей.

Для ответственных за мероприятия приложение предлагает возможность считывания QR-кода и последующего подтверждения фактического присутствия студентов на мероприятии с помощью камеры устройства.

Разработанное мобильное приложение повышает эффективность процесса регистрации на мероприятия и уменьшает время подтверждения фактического присутствия студентов на 45 минут. Следовательно, процесс оптимизирован, цель достигнута.

Список использованных источников

1. Рейтинг // ИРНИТУ. – URL: <https://www.istu.edu/studentu/stipendiya/rejting> (дата обращения: 27.04.2023).
2. Кунгурцев А.Б. Метод построения словарей предметных областей для извлечения фактов из текстов на естественном языке / Кунгурцев А.Б., Бородавкин С.Н., Голуб А.П. // Восточно-Европейский журнал передовых технологий. – 2010. – №43. – С. 32-36.
3. ГОСТ Р 59383-2021. Информационные технологии. Методы и средства обеспечения безопасности. Основы управления доступом.
4. Владимцев Н.В. Принцип моделирования бизнес-процессов в стандарте IDEF0 / Владимцев Н.В., Извольская И.В. // Экономический анализ: теория и практика. – 2008. – №114. – С. 11-17.
5. Новиков Ф.А. Учебно-методическое пособие по дисциплине «Анализ и проектирование на UML». / Новиков Ф.А. – СПб: СПбГУ ИТМО, 2007. – 286 с.
6. О ПРОЕКТЕ // RUNET. – URL: <https://runet-id.com/page/info/about> (дата обращения: 29.04.2023).
7. Зачем вам Leader-ID // Leader-ID. – URL: <https://leader-id.ru/intro> (дата обращения: 02.05.2023).
8. Мероприятия // ФГАИС «Молодежь России». – URL: <https://myrosmol.ru/measures> (дата обращения: 03.05.2023).
9. Kotlin Programming Language // kotlinlang. – URL: <https://kotlinlang.org/> (дата обращения: 04.05.2023).
10. Material Design // Material Design. – URL: <https://m3.material.io/> (дата обращения: 04.05.2023).
11. Volley overview // Volley. – URL: <https://google.github.io/volley/> (дата обращения: 04.05.2023).
12. draw.io Documentation // draw.io. – URL: <https://www.drawio.com/doc/> (дата обращения: 05.05.2023).
13. RAMUS // RAMUS. – URL: <https://ramussoftware.com/> (дата обращения: 05.05.2023).
14. Документация по GitHub. // GitHub – URL: <https://docs.github.com/ru> (дата обращения: 05.05.2023).
15. Meet Android Studio bookmark_border // Android Developers. – URL: <https://developer.android.com/studio/intro> (дата обращения: 06.05.2023).
16. The collaborative interface design tool // Figma. – URL: <https://www.figma.com> (дата обращения: 06.05.2023).
17. Курганова А. Г. Процесс выбора архитектуры для мобильного приложения. / Курганова А. Г. // StudNet. – 2021. – №6. – С. 386-397.
18. Data classes // Kotlin Documentation. – URL: <https://kotlinlang.org/docs/data-classes.html> (дата обращения: 06.05.2023).
19. Фаулер М. UML Основы. Краткое руководство по стандартному языку объектного моделирования / Фаулер М. – 3 изд. – СПб: Символ-Плюс, 2004. – 184 с. - ISBN 5-93286-060-X.

20. Сергеев С.Ф. Введение в проектирование интеллектуальных интерфейсов: Учебное пособие. / Сергеев С.Ф., Падерно П.И., Назаренко Н.А. – СПб: СПбГУ ИТМО, 2011. – 108 с.

21. Сколько зарабатывает программист стажер в Иркутске // Фильтр работ. – URL: <https://irkutsk.jobfilter.ru/career/программист-стажер> (дата обращения: 02.06.2023).

22. Чуйкова Л. Ю. Безопасность жизнедеятельности (лекционный курс) / Чуйкова Л. Ю., Чуйков Ю., С. // Астраханский вестник экологического образования. – 2016. – №38. – С. 113-172.

23. Постановление Правительства РФ от 7 ноября 2019 г. № 1421. «Об утверждении требований к антитеррористической защищенности объектов (территорий) Министерства науки и высшего образования Российской Федерации и подведомственных ему организаций, объектов (территорий), относящихся к сфере деятельности Министерства науки и высшего образования Российской Федерации, формы паспорта безопасности этих объектов (территорий) и признании утратившими силу некоторых актов Правительства Российской Федерации».

24. СП 14.13330.2010. Строительство в сейсмических районах.

25. Точка кипения – ИРНИТУ. // ИРНИТУ – URL: https://www.istu.edu/deyatelnost/molodezhnaya_politika/nauka/tk (дата обращения: 11.04.2023).

26. Отдел развития научно-исследовательской и предпринимательской деятельности студентов // ИРНИТУ. – URL: https://www.istu.edu/deyatelnost/nauka/nauchnoe_upravlenie/nirs (дата обращения: 11.04.2023).

27. Р 2.2.2006-05. Руководство, по гигиенической оценке, факторов рабочей среды и трудового процесса. Критерии и классификация условий труда.

28. ГОСТ 12.0.003-2015. Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация.

29. СанПиН 1.2.3685-21. Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания.

30. СанПиН 52.13330.2016. Естественное и искусственное освещение.

31. ГОСТ 12.1.019-2017. Система стандартов безопасности труда. Электробезопасность. Общие требования и номенклатура видов защиты.

32. Федеральный закон «Технический регламент о требованиях пожарной безопасности» от 22.07.2008 N 123-ФЗ.

Приложение А

Диаграмма классов по пакетам

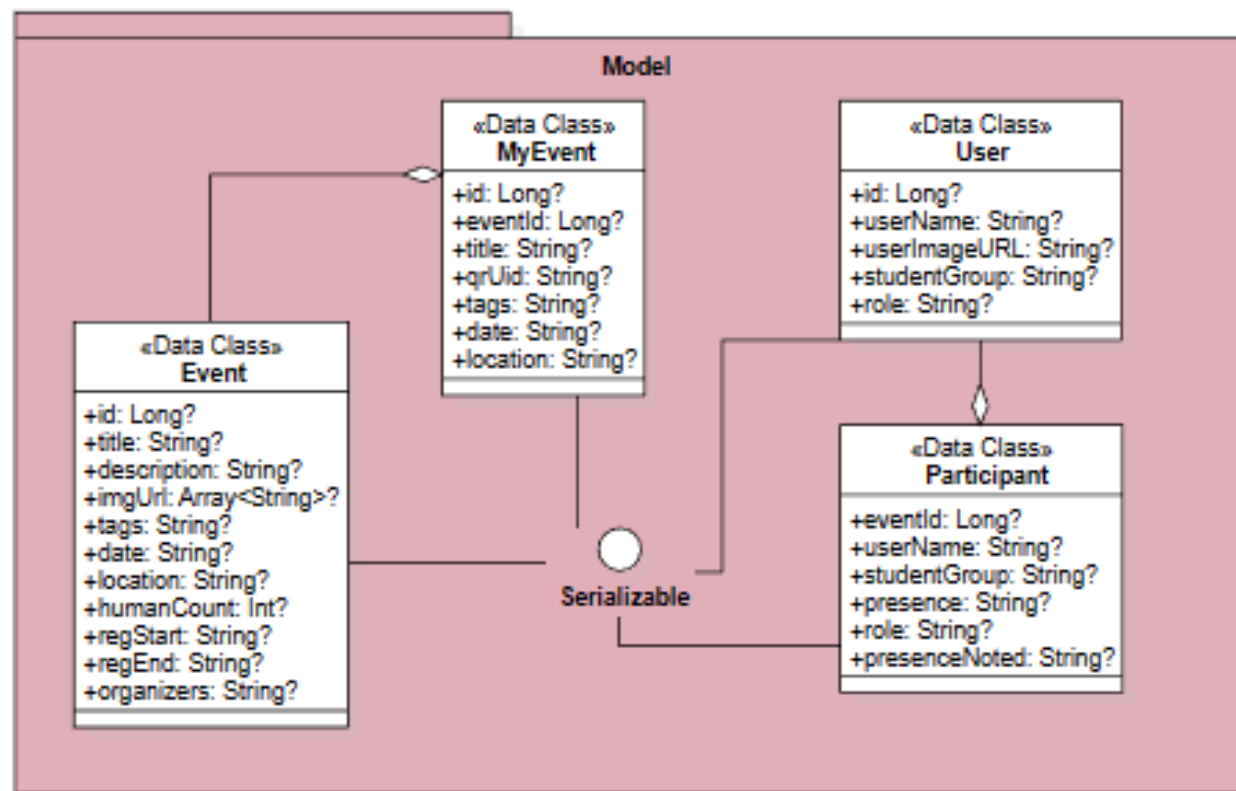


Рисунок А.1 – Диаграмма классов пакета Model

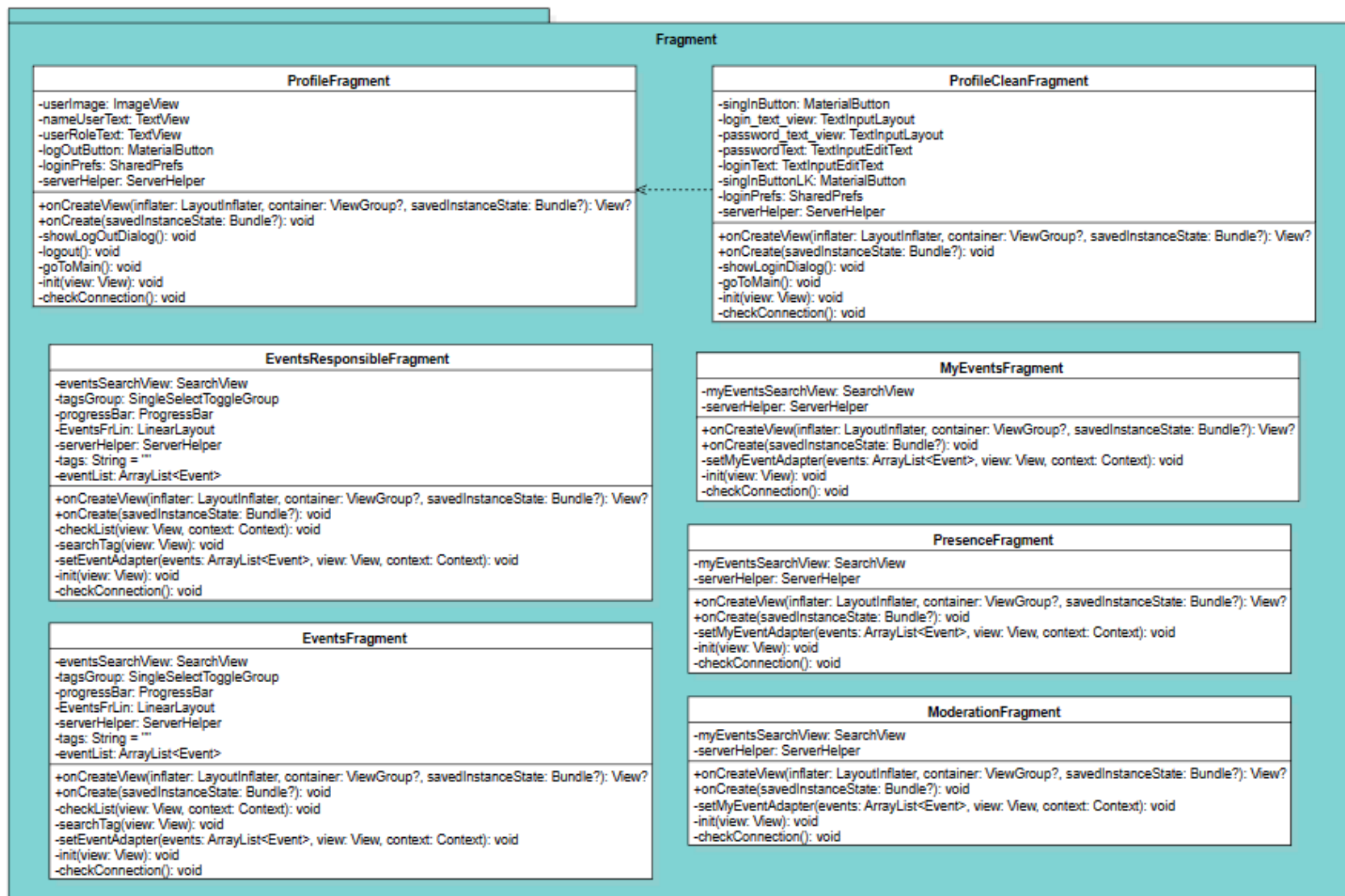


Рисунок А.2 – Диаграмма классов пакета Fragment

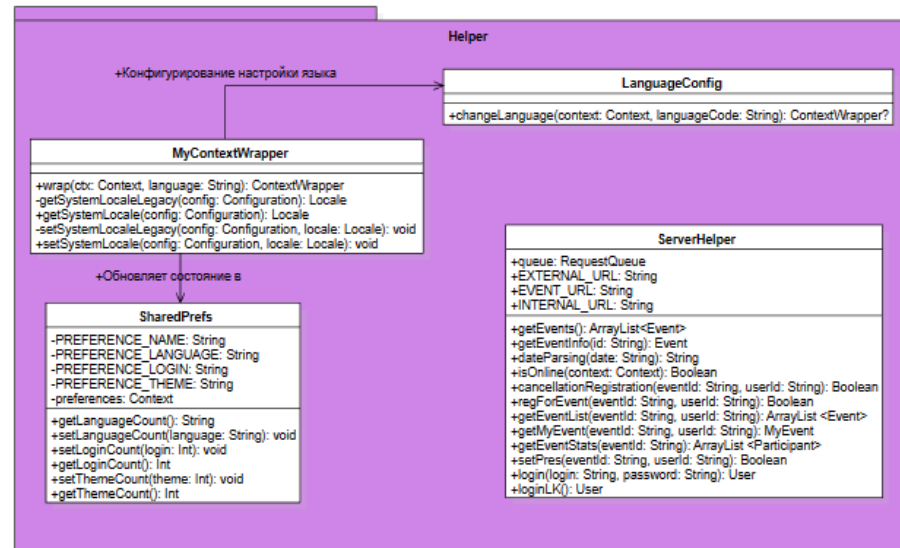


Рисунок А.3 – Диаграмма классов пакета Helper

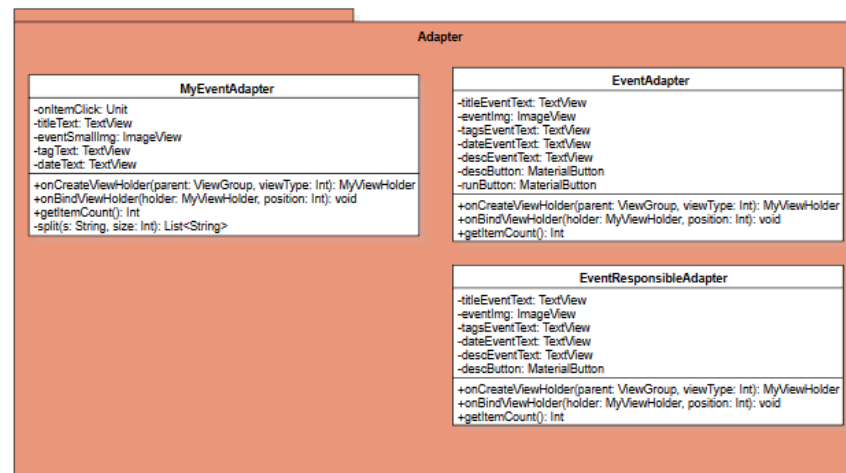


Рисунок А.4 – Диаграмма классов пакета Adapter

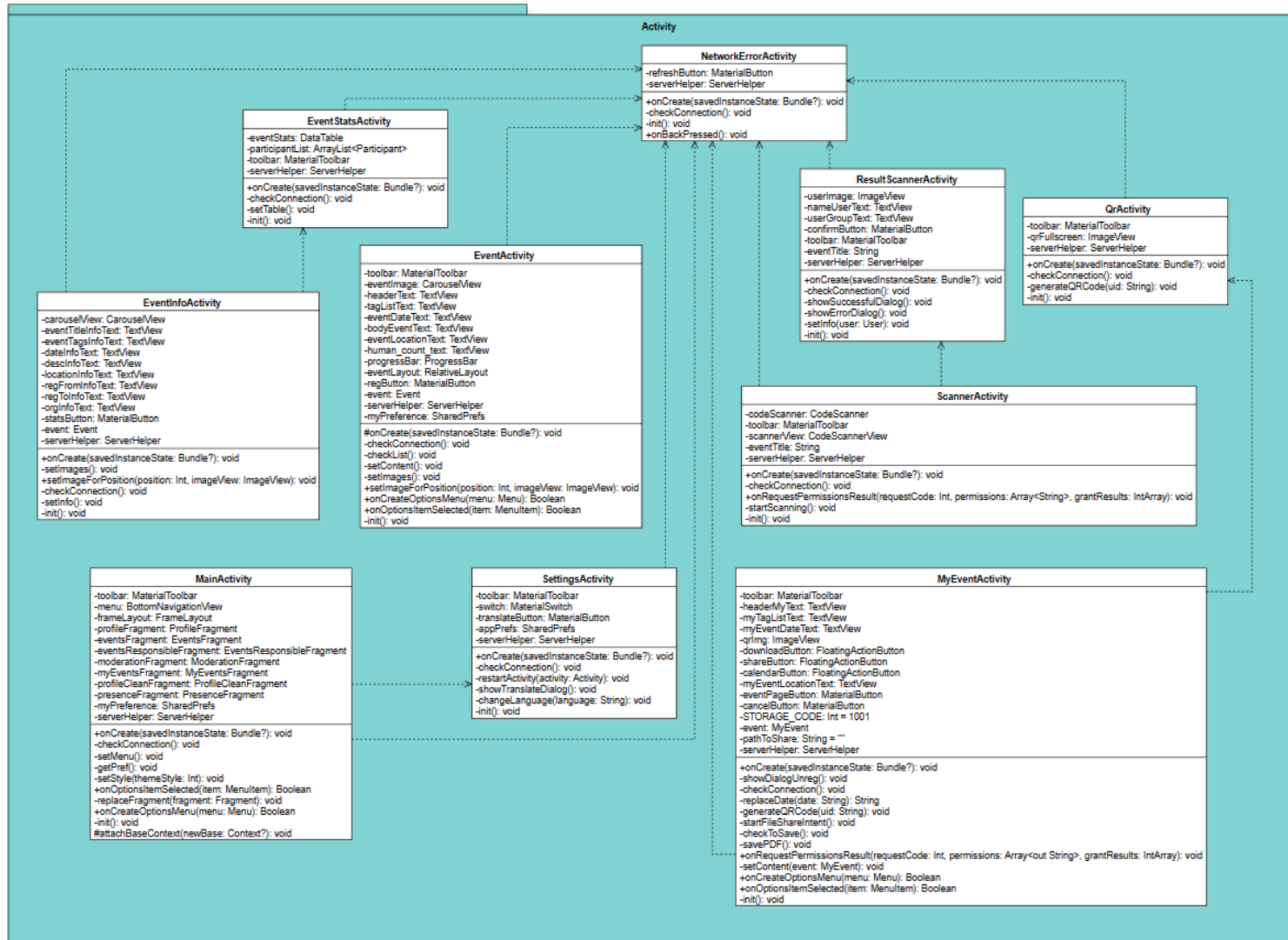


Рисунок А.5 – Диаграмма классов пакета Activity

Приложение Б

Описание реализованных классов

Таблица Б.1 – Спецификация класса EventActivity

Название	Тип	Уровень доступа	Аргументы	Назначение
onCreate	void	protected	savedInstanceState: Bundle?	Выполнение статической настройки Activity: создание View, привязка элементов управления
checkConnection	void	private	—	Проверка доступности интернет-соединения
setContent	void	private	—	Заполнение компонентов информацией
setImages	void	private	—	Настройка карусели изображений
setImageForPosition	void	public	position: Int, imageView: ImageView	Загрузка изображений в карусель
onCreateOptionsMenu	Boolean	public	menu: Menu	Создание контекстного меню
onOptionsItemSelected	Boolean	public	item: MenuItem	Слушатель элементов меню
init	void	private	—	Инициализация компонентов Activity

Таблица Б.2 – Спецификация класса EventInfoActivity

Название	Тип	Уровень доступа	Аргументы	Назначение
onCreate	void	protected	savedInstanceState: Bundle?	Выполнение статической настройки Activity: создание View, привязка элементов управления
setImages	void	private	–	Настройка карусели изображений
setImageForPosition	void	public	position: Int, imageView: ImageView	Загрузка изображений в карусель
checkConnection	void	private	–	Проверка доступности интернет-соединения
setInfo	void	private	–	Заполнение компонентов информацией
init	void	private	–	Инициализация компонентов Activity

Таблица Б.3 – Спецификация класса EventStatsActivity

Название	Тип	Уровень доступа	Аргументы	Назначение
onCreate	void	protected	savedInstanceState: Bundle?	Выполнение статической настройки Activity: создание View, привязка элементов управления
checkConnection	void	private	–	Проверка доступности интернет-соединения
setTable	void	private	–	Заполнение таблицы данными
init	void	private	–	Инициализация компонентов Activity

Таблица Б.4 – Спецификация класса MainActivity

Название	Тип	Уровень доступа	Аргументы	Назначение
onCreate	void	protected	savedInstanceState: Bundle?	Выполнение статической настройки Activity: создание View, привязка элементов управления
checkConnection	void	private	–	Проверка доступности интернет-соединения
setMenu	void	private	–	Слушатель меню для смены Fragment
getPref	void	private	–	Получение данных из SharedPreferences
setStyle	void	private	themeStyle: Int	Установка графической темы приложения
onCreateOptionsMenu	Boolean	public	menu: Menu	Создание контекстного меню
onOptionsItemSelected	Boolean	public	item: MenuItem	Слушатель элементов меню
replaceFragment	void	private	fragment: Fragment	Смена Fragment главного экрана
init	void	private	–	Инициализация компонентов Activity
attachBaseContext	void	protected	newBase: Context?	Переопределение метода attachBaseContext для получения context и корректной смены языка

Таблица Б.5 – Спецификация класса MyEventActivity

Название	Тип	Уровень доступа	Аргументы	Назначение
onCreate	void	protected	savedInstanceState: Bundle?	Выполнение статической настройки Activity: создание View, привязка элементов управления
showDialogUnreg	void	private	–	Вызов AlertDialog отмены регистрации на мероприятие
checkConnection	void	private	–	Проверка доступности интернет-соединения
replaceDate	String	private	date: String	Изменение даты и времени для формата календаря
generateQRCode	void	private	uid: String	Генерация QR-кода из uid
startFileShareIntent	void	private	–	Создание и вызов Intent отправки файла
checkToSave	void	private	–	Проверка разрешения для записи данных в файл
savePDF	void	private	–	Запись данных в pdf-файл
onRequestPermissionsResult	void	public	requestCode: Int, permissions: Array<out String>, grantResults: IntArray	Получение разрешения на сохранение файла
setContent	void	private	–	Заполнение компонентов информацией
onCreateOptionsMenu	Boolean	public	menu: Menu	Создание контекстного меню
onOptionsItemSelected	Boolean	public	item: MenuItem	Слушатель элементов меню
init	void	private	–	Инициализация компонентов Activity

Таблица Б.6 – Спецификация класса NetworkErrorActivity

Название	Тип	Уровень доступа	Аргументы	Назначение
onCreate	void	protected	savedInstanceState: Bundle?	Выполнение статической настройки Activity: создание View, привязка элементов управления
checkConnection	void	private	–	Проверка доступности интернет-соединения
init	void	private	–	Инициализация компонентов Activity
onBackPressed	void	public	–	Переопределение метода onBackPressed для запрета возврата в предыдущую Activity

Таблица Б.7 – Спецификация класса QrActivity

Название	Тип	Уровень доступа	Аргументы	Назначение
onCreate	void	protected	savedInstanceState: Bundle?	Выполнение статической настройки Activity: создание View, привязка элементов управления
checkConnection	void	private	–	Проверка доступности интернет-соединения
generateQRCode	void	private	uid: String	Генерация QR-кода из uid
init	void	private	–	Инициализация компонентов Activity

Таблица Б.8 – Спецификация класса ResultScannerActivity

Название	Тип	Уровень доступа	Аргументы	Назначение
onCreate	void	protected	savedInstanceState: Bundle?	Выполнение статической настройки Activity, создание View
checkConnection	void	private	–	Проверка доступности интернет-соединения
showSuccessfulDialog	void	private	–	Вызов AlertDialog успешного подтверждения присутствия
showErrorDialog	void	private	–	Вызов AlertDialog неудачного подтверждения присутствия
setInfo	void	private	–	Заполнение компонентов информацией
init	void	private	–	Инициализация компонентов Activity

Таблица Б.9 – Спецификация класса ScannerActivity

Название	Тип	Уровень доступа	Аргументы	Назначение
onCreate	void	protected	savedInstanceState: Bundle?	Выполнение статической настройки Activity, создание View
checkConnection	void	private	–	Проверка доступности интернет-соединения
onRequestPermissionsResult	void	public	requestCode: Int, permissions: Array<String>, grantResults: IntArray	Получение разрешения на доступ к камере
startScanning	void	private	–	Сканирование QR-кодов
init	void	private	–	Инициализация компонентов Activity

Таблица Б.10 – Спецификация класса SettingsActivity

Название	Тип	Уровень доступа	Аргументы	Назначение
onCreate	void	protected	savedInstanceState: Bundle?	Выполнение статической настройки Activity: создание View, привязка элементов управления
checkConnection	void	private	–	Проверка доступности интернет-соединения
restartActivity	void	private	activity: Activity	Перезагрузка Activity
showTranslateDialog	void	private	–	Вызов AlertDialog смены языка
changeLanguage	void	private	language: String	Смена языка приложения
init	void	private	–	Инициализация компонентов Activity

Таблица Б.11 – Спецификация класса EventAdapter

Название	Тип	Уровень доступа	Аргументы	Назначение
onCreateViewHolder	My-View-Holder	public	parent: ViewGroup, viewType: Int	Инициализация списка RecyclerView
onBindViewHolder	void	public	holder: MyViewHolder, position: Int	Заполнение RecyclerView данными
getItemCount	Int	public	–	Получение размера списка RecyclerView

Таблица Б.12 – Спецификация класса EventResponsibleAdapter

Название	Тип	Уровень доступа	Аргументы	Назначение
onCreateViewHolder	My-View-Holder	public	parent: ViewGroup, viewType: Int	Инициализация списка RecyclerView
onBindViewHolder	void	public	holder: MyViewHolder, position: Int	Заполнение RecyclerView данными
getItemCount	Int	public	–	Получение размера списка RecyclerView

Таблица Б.13 – Спецификация класса MyEventAdapter

Название	Тип	Уровень доступа	Аргументы	Назначение
onCreateViewHolder	My-View-Holder	public	parent: ViewGroup, viewType: Int	Инициализация списка RecyclerView
onBindViewHolder	void	public	holder: MyViewHolder, position: Int	Заполнение RecyclerView данными
getItemCount	Int	public	–	Получение размера списка RecyclerView
split	List<String>	private	s: String, size: Int	Деление строки по задаваемому размеру

Таблица Б.14 – Спецификация классов EventsFragment и EventsResponsibleFragment

Название	Тип	Уровень доступа	Аргументы	Назначение
onCreate	void	protected	savedInstanceState: Bundle?	Выполнение статической настройки Activity: создание View, привязка элементов управления
onCreateView	View?	public	inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?	Настройка контроллера Fragment
checkList	void	private	view: View, context: Context	Проверка готовности данных для заполнения RecyclerView
searchTag	void	private	view: View	Фильтрация мероприятий
setEventAdapter	void	private	events: ArrayList<Event>, view: View, context: Context	Загрузка данных в RecyclerView
checkConnection	void	private	–	Проверка доступности интернет-соединения
init	void	private	view: View	Инициализация компонентов Activity

Таблица Б.15 – Спецификация классов ModerationFragment, MyEventsFragment и PresenceFragment

Название	Тип	Уровень доступа	Аргументы	Назначение
onCreate	void	protected	savedInstanceState: Bundle?	Выполнение статической настройки Activity: создание View, привязка элементов управления
onCreateView	View?	public	inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?	Настройка контроллера Fragment

Продолжение таблицы Б.15

Название	Тип	Уровень доступа	Аргументы	Назначение
setAdapter	void	private	events: ArrayList<Event>, view: View, context: Context	Загрузка данных в RecyclerView
checkConnection	void	private	—	Проверка доступности интернет-соединения
init	void	private	view: View	Инициализация компонентов Activity

Таблица Б.16 – Спецификация класса ProfileCleanFragment

Название	Тип	Уровень доступа	Аргументы	Назначение
onCreate	void	protected	savedInstanceState: Bundle?	Выполнение статической настройки Activity: создание View, привязка элементов управления
onCreateView	View?	public	inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?	Настройка контроллера Fragment
showLoginDialog	void	private	—	Вызов AlertDialog авторизации
goToMain	void	private	—	Переход на главный экран
init	void	private	view: View	Инициализация компонентов Activity
checkConnection	void	private	—	Проверка доступности интернет-соединения

Таблица Б.17 – Спецификация класса ProfileFragment

Название	Тип	Уровень доступа	Аргументы	Назначение
onCreate	void	protected	savedInstanceState: Bundle?	Выполнение статической настройки Activity: создание View, привязка элементов управления
onCreateView	View?	public	inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?	Настройка контроллера Fragment
showLoginDialog	void	private	–	Вызов AlertDialog выхода
logout	void	private	–	Выход из профиля
goToMain	void	private	–	Переход на главный экран
init	void	private	view: View	Инициализация компонентов Activity
checkConnection	void	private	–	Проверка доступности интернет-соединения

Таблица Б.18 – Спецификация класса LanguageConfig

Название	Тип	Уровень доступа	Аргументы	Назначение
changeLanguage	Context-Wrapper?	private	context: Context, language-Code: String	Смена языка приложения

Таблица Б.19 – Спецификация класса MyContextWrapper

Название	Тип	Уровень доступа	Аргументы	Назначение
wrap	Context-Wrapper	public	ctx: Context, language: String	Смена языка приложения и сохранение параметров
getSystemLocale-Legacy	Locale	private	config: Configuration	Получение языковой конфигурации приложения
getSystemLocale	Locale	private	config: Configuration	Получение языковой конфигурации приложения
setSystemLocale-Legacy	void	private	config: Configuration, locale: Locale	Установка языковой конфигурации приложения
setSystemLocale	void	private	config: Configuration, locale: Locale	Установка языковой конфигурации приложения

66

Таблица Б.20 – Спецификация класса ServerHelper

Название	Тип	Уровень доступа	Аргументы	Назначение
getEvents	ArrayList <Event>	public	—	Получение списка мероприятий
getEventInfo	Event	public	id: Long	Получение информации о мероприятии по id
dateParsing	String	private	date: String	Изменение формата даты и времени
isOnline	Boolean	public	context: Context	Проверка доступности интернет-соединения
cancellationRegistration	Boolean	public	eventId: String, userId: Long	Отмена регистрации на мероприятие
regForEvent	Boolean	public	eventId: Long, userId: Long	Регистрация на мероприятие

Продолжение таблицы Б.20

Название	Тип	Уровень доступа	Аргументы	Назначение
getEventList	ArrayList <Event>	public	eventId: Long, userId: Long	Получение списка мероприятий пользователя
getMyEvent	MyEvent	public	eventId: Long, userId: Long	Получение информации о мероприятии с пройденной регистрацией
getEventStats	ArrayList <Partici- pant>	public	eventId: Long	Получение статистики мероприятия
setPres	Boolean	public	eventId: Long, userId: Long	Подтверждение присутствия студента на мероприятии
login	User	public	login: String, password: String	Авторизация с помощью логина и пароля
loginLK	User	public	—	Авторизация с помощью Личного кабинета
checkUser	User	public	uid:String, eventId: Long	Проверка регистрации студента на мероприятие
getUser	User	public	userId: Long	Получение информации о пользователе

Приложение В

Анкета тестирования приложения

На устройстве какого типа вы проводили тестирование?

1. Планшет.
2. Смартфон.

Какая версия Android установлена на Вашем устройстве?

1. Android 5.0 Lollipop.
2. Android 6.0 Marshmallow.
3. Android 7.0 Nougat.
4. Android 8.0 Oreo.
5. Android 9.0 Pie.
6. Android 10.
7. Android 11.
8. Android 12.
9. Android 13.

Сталкивались ли Вы с багами / ошибками / аварийным завершением работы приложения?

1. Нет, такого не было.
2. Долго загружалась информация.
3. С задержкой появлялись изображения мероприятий.
4. Были вылеты приложения.
5. Другое.

Стабильно ли работало приложение при слабом подключении к сети?

1. Стабильно.
2. Стабильно, но долго отображало информацию.
3. Не было случаев плохого соединения с интернетом.
4. Были проблемы.

Оцените дизайн приложения

1. 5 из 5.
2. 4 из 5.
3. 3 из 5.
4. 2 из 5.
5. 1 из 5.

Оцените интуитивность взаимодействия с графическим интерфейсом

1. 5 из 5.
2. 4 из 5.
3. 3 из 5.
4. 2 из 5.
5. 1 из 5.