2 Проектирование системы

2.1 Обоснование использования программных средств разработки

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2.2 Архитектура подсистемы

2.2.1 Общая архитектура системы

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

Архитектура системы представлена на рисунке 2.1.

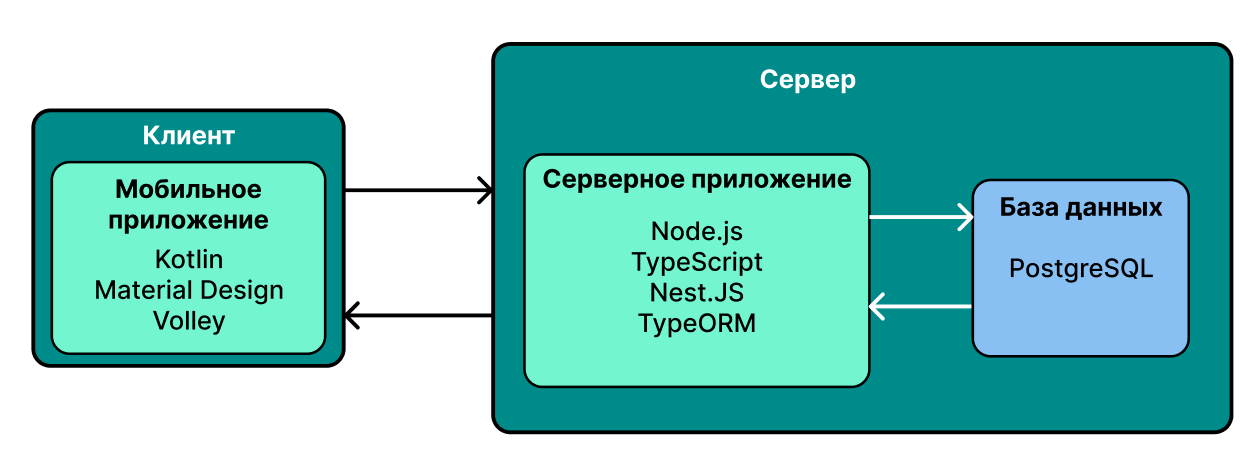


Рисунок 2.1 – Архитектура системы

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

2.2.2 Архитектура мобильного приложения

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

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

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

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

Список фрагментов в приложении включает в себя:

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

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

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

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

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

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

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

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

Архитектура мобильного приложения представлена на рисунке 2.2.

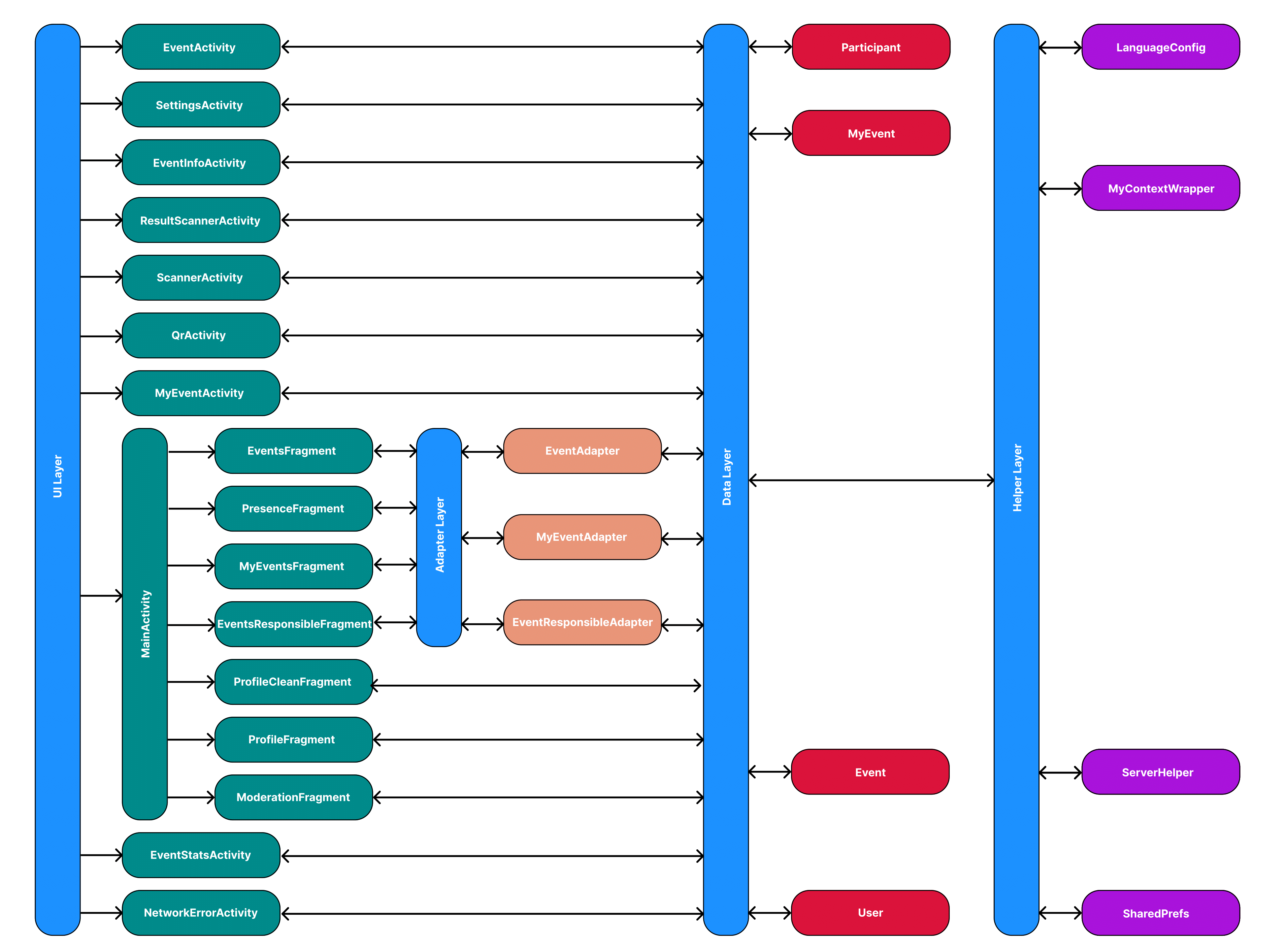


Рисунок 2.2 – Архитектура мобильного приложения

2.3 Проектирование классов данных

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

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

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

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

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

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

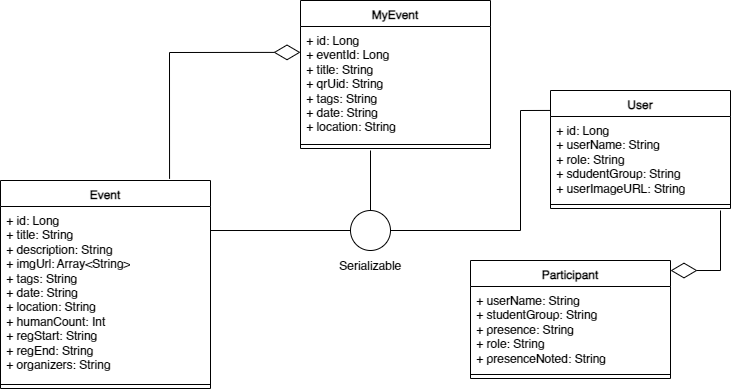


Рисунок 2.3 – Классы данных приложения

Таблица 2.1 – Описание полей Data-классов

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

Продолжение таблицы 2.1 – Описание полей Data-классов

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

Продолжение таблицы 2.1 – Описание полей Data-классов

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

2.4 Описание сценариев использования

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

Ниже представлены диаграммы спецификаций функций мобильного приложения.

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

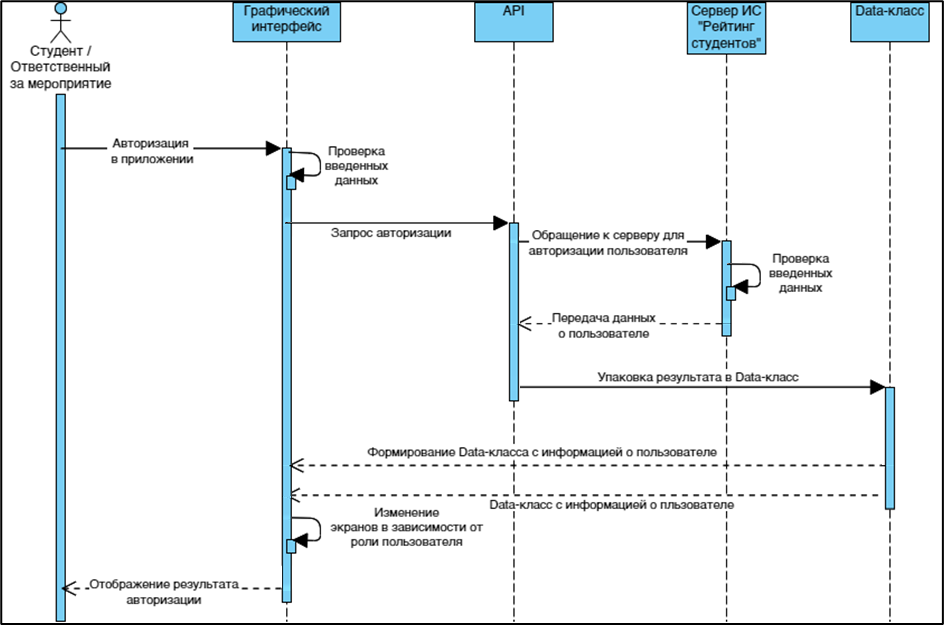


Рисунок 2.4 – «Авторизоваться с помощью логина и пароля» и «Авторизация через Кампус»

2.4.2 Спецификация варианта использования «Регистрация на мероприятие»

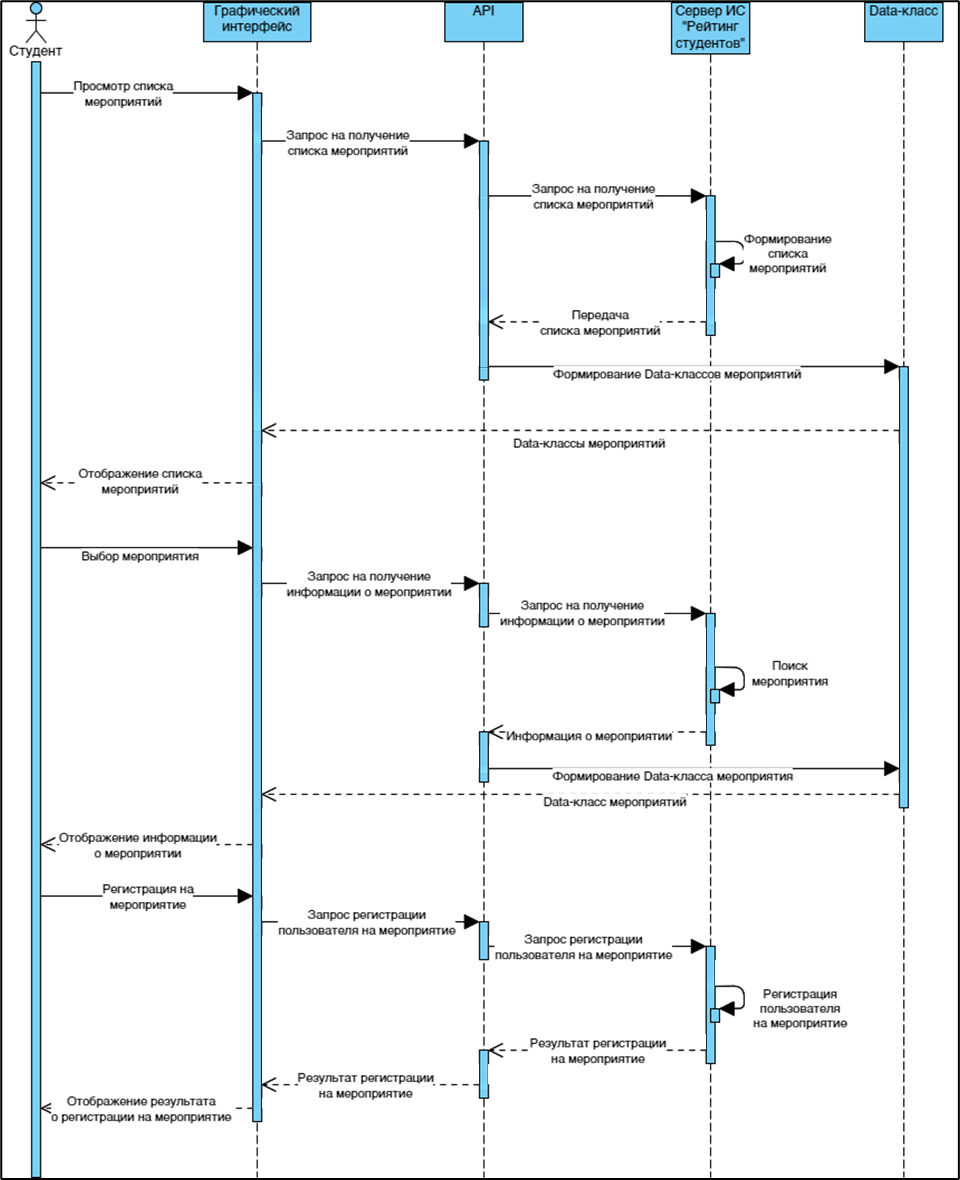


Рисунок 2.5 – «Регистрация на мероприятие»

2.4.3 Спецификация варианта использования «Подтвердить присутствие участника»

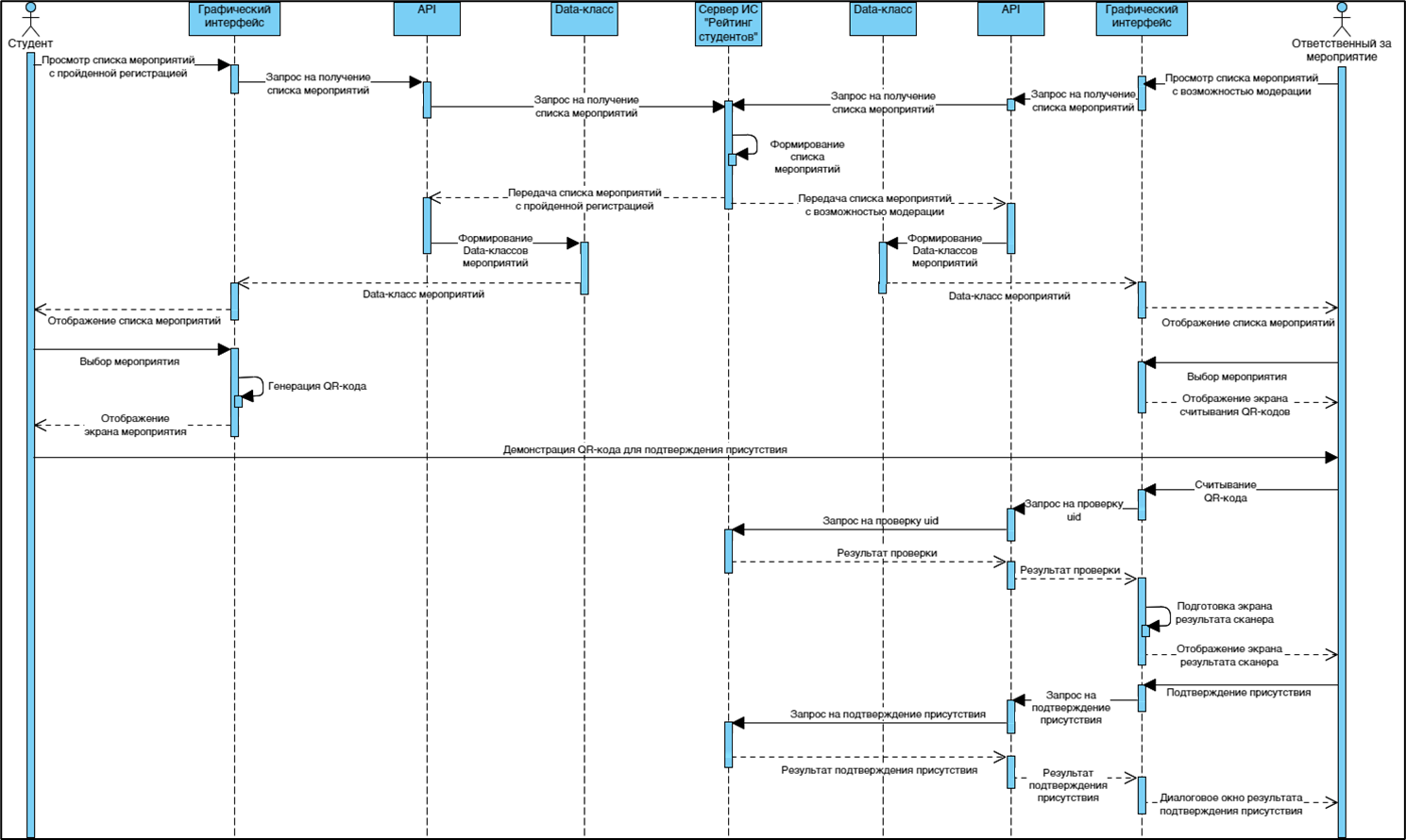


Рисунок 2.6 – «Подтвердить присутствие участника»

2.5 Проектирование графического интерфейса

Для реализации графического интерфейса спроектирован набор экранов.

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

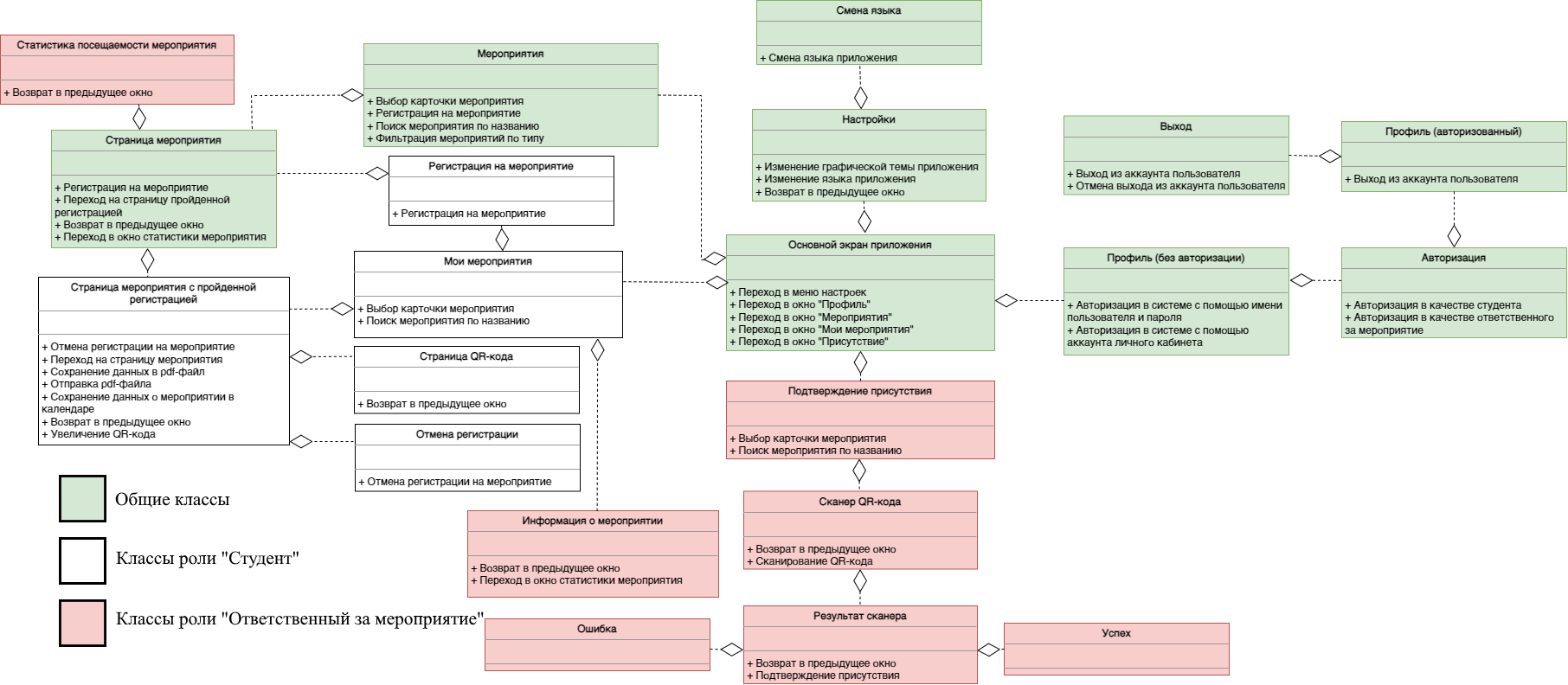


Рисунок 2.7 – Диаграмма интерфейсных классов приложения

2.5.1 Проектирование графического интерфейса для роли «Студент»

**2.5.1.1 Описание основного экрана приложения**

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

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

Макет данного экрана представлен на рисунке 2.8.

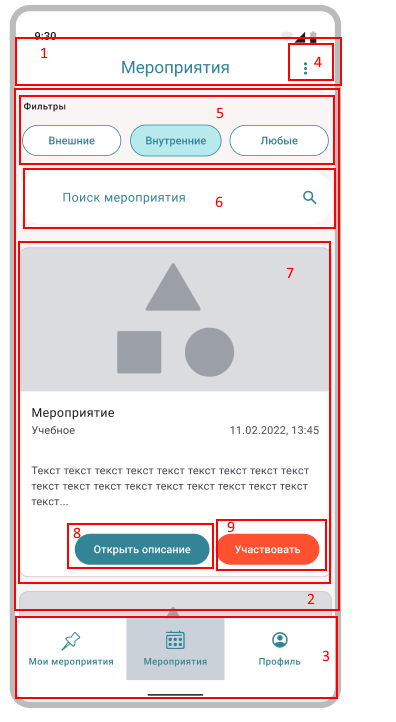


Рисунок 2.8 – Макет основного экрана приложения

**2.5.1.2 Описание экрана «Мои мероприятия»**

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

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

Макет данного экрана представлен на рисунке 2.9.

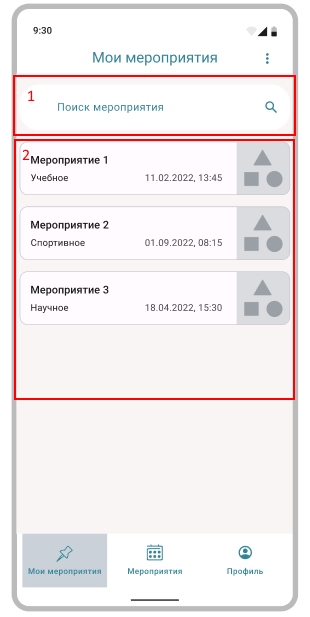


Рисунок 2.9 – Макет экрана «Мои мероприятия»

**2.5.1.3 Описание экрана «Мероприятие»**

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

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

Макет данного экрана представлен на рисунке 2.10.

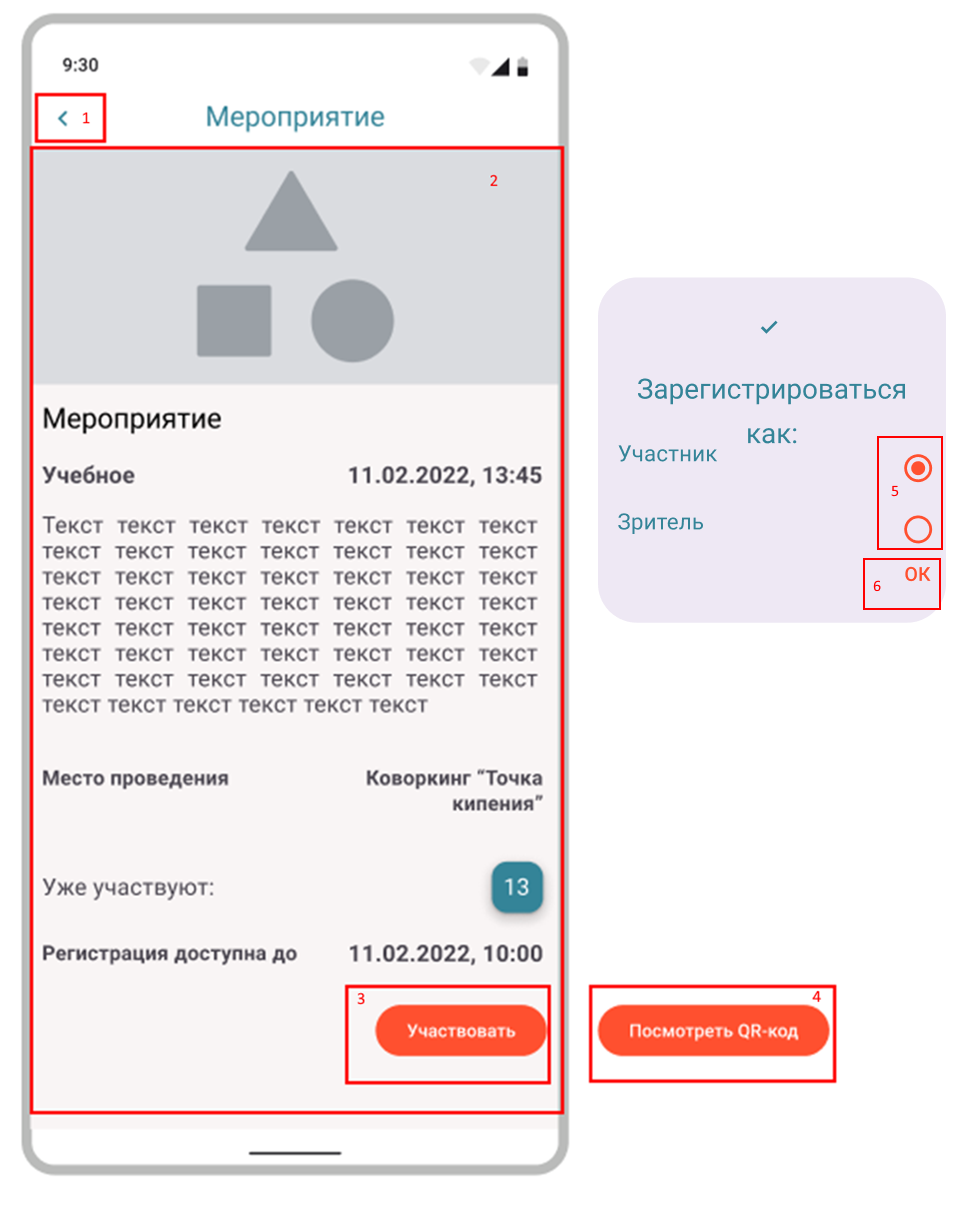


Рисунок 2.10 – Макет экрана «Мероприятие»

**2.5.1.4 Описание экрана мероприятия с пройденной регистрацией**

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

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

Макет данного экрана представлен на рисунке 2.11.

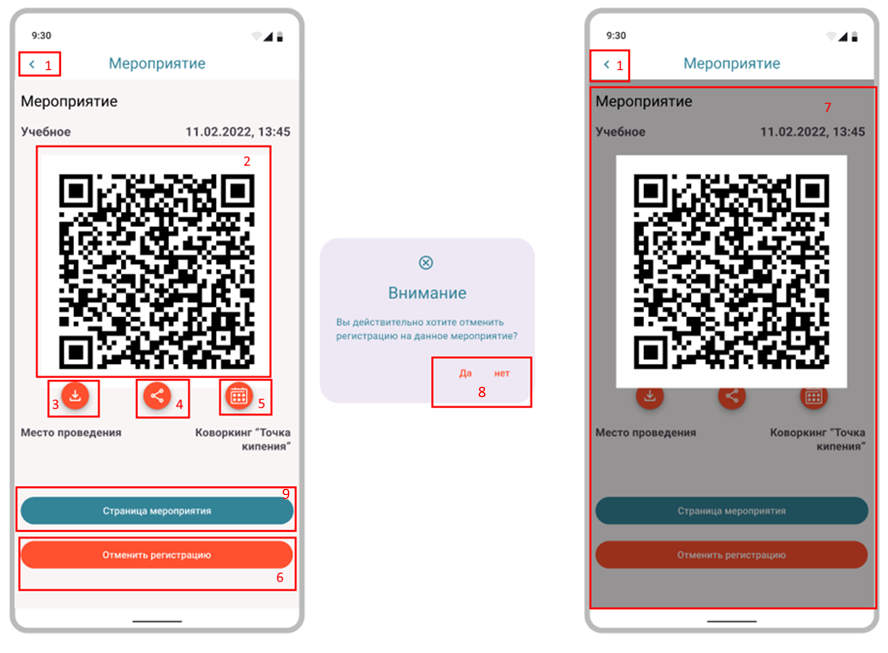


Рисунок 2.11 – Макет экрана мероприятия с пройденной регистрацией

2.5.2 Проектирование графического интерфейса для роли «Ответственный за мероприятие»

**2.5.2.1 Описание основного экрана приложения**

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

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

Макет данного экрана представлен на рисунке 2.12.

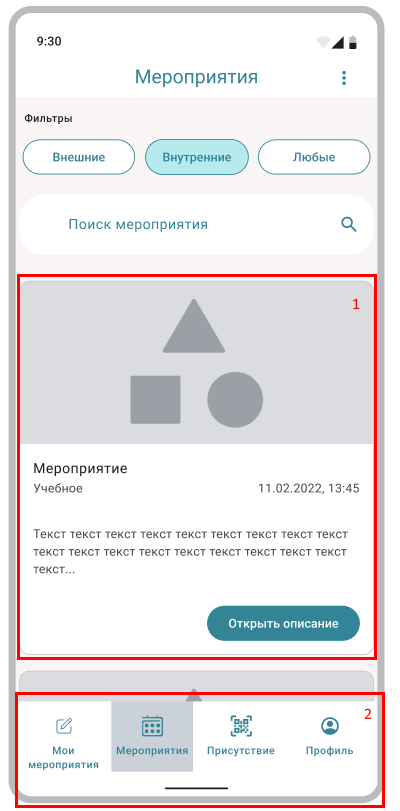


Рисунок 2.12 – Макет основного экрана приложения

**2.5.2.2 Описание экрана «Мероприятие»**

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

Макет данного экрана представлен на рисунке 2.13.

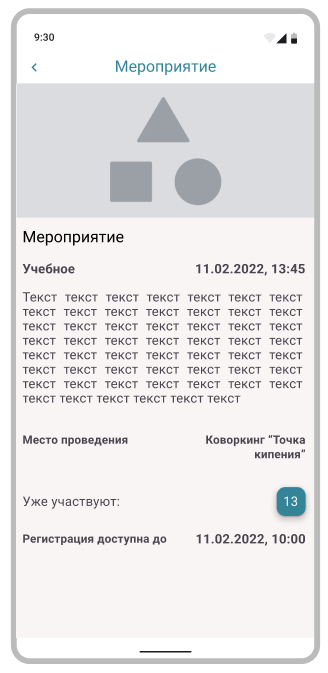


Рисунок 2.13 – Макет экрана «Мероприятие»

**2.5.2.3 Описание экрана «Подтверждение присутствия»**

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

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

Макет данного экрана представлен на рисунке 2.14.

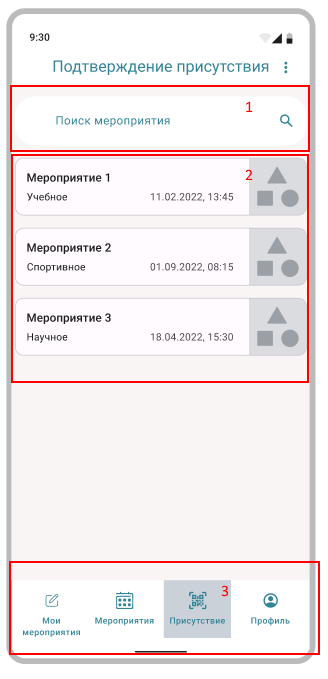


Рисунок 2.14 – Макет экрана «Подтверждение присутствия»

**2.5.2.4 Описание экрана сканера QR-кодов**

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

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

Макет данного экрана представлен на рисунке 2.15.

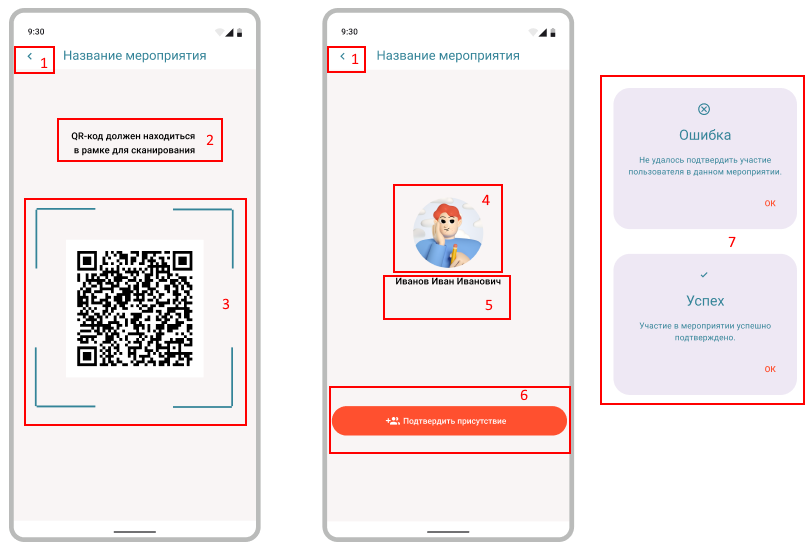


Рисунок 2.15 – Макет экрана сканера QR-кода

**2.5.2.5 Описание экрана «Мои мероприятия»**

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

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

Макет данного экрана представлен на рисунке 2.16.

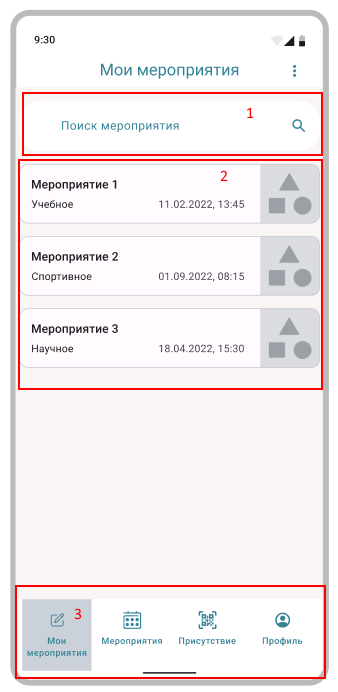


Рисунок 2.16 – Макет экрана «Мои мероприятия»

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

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

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

Макет данного экрана представлен на рисунке 2.17.

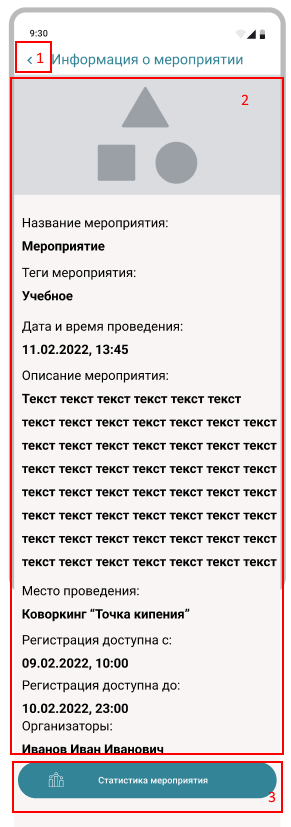


Рисунок 2.17 – Макет экрана «Информация о мероприятии»

**2.5.2.7 Описание экрана статистики мероприятия**

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

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

Макет данного экрана представлен на рисунке 2.18.



Рисунок 2.18 – Макет экрана статистики мероприятия

2.5.3 Проектирование графического интерфейса непривязанных к конкретной роли

**2.5.3.1 Описание экрана «Профиль»**

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

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

Макет данного экрана представлен на рисунке 2.19.

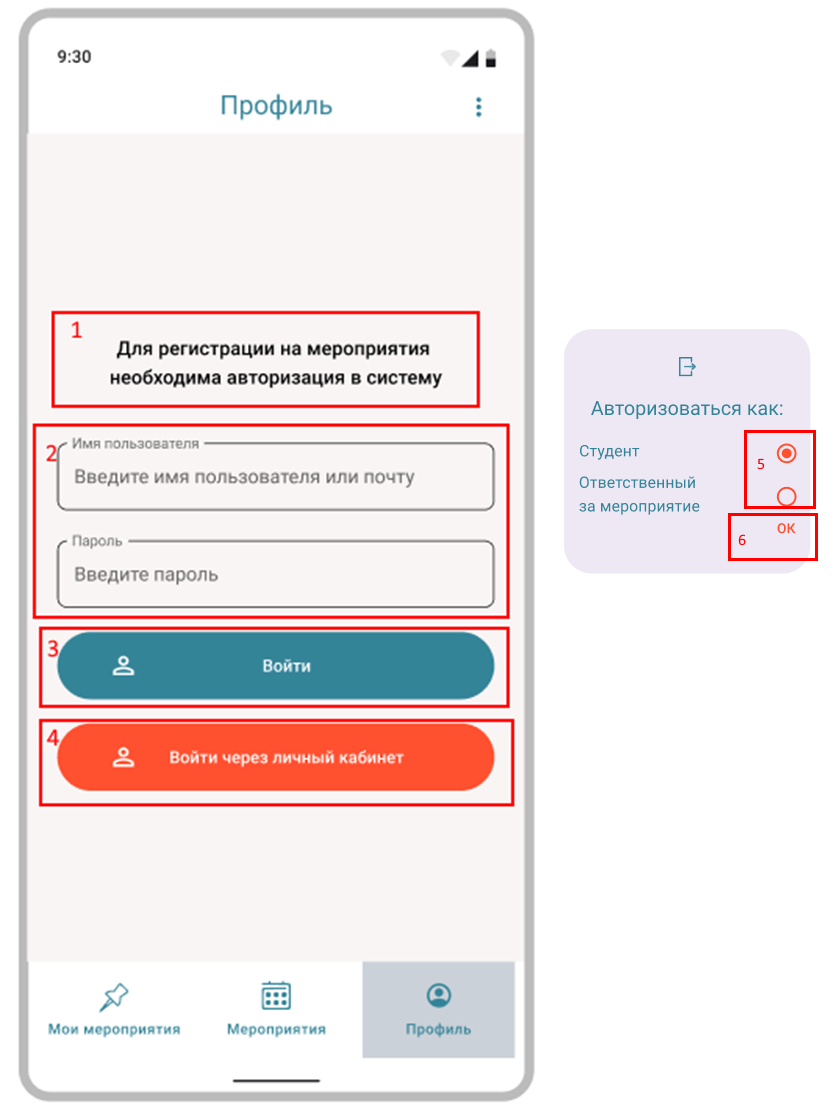


Рисунок 2.19 – Макет экрана «Профиль»

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

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

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

Макет данного экрана представлен на рисунке 2.20.

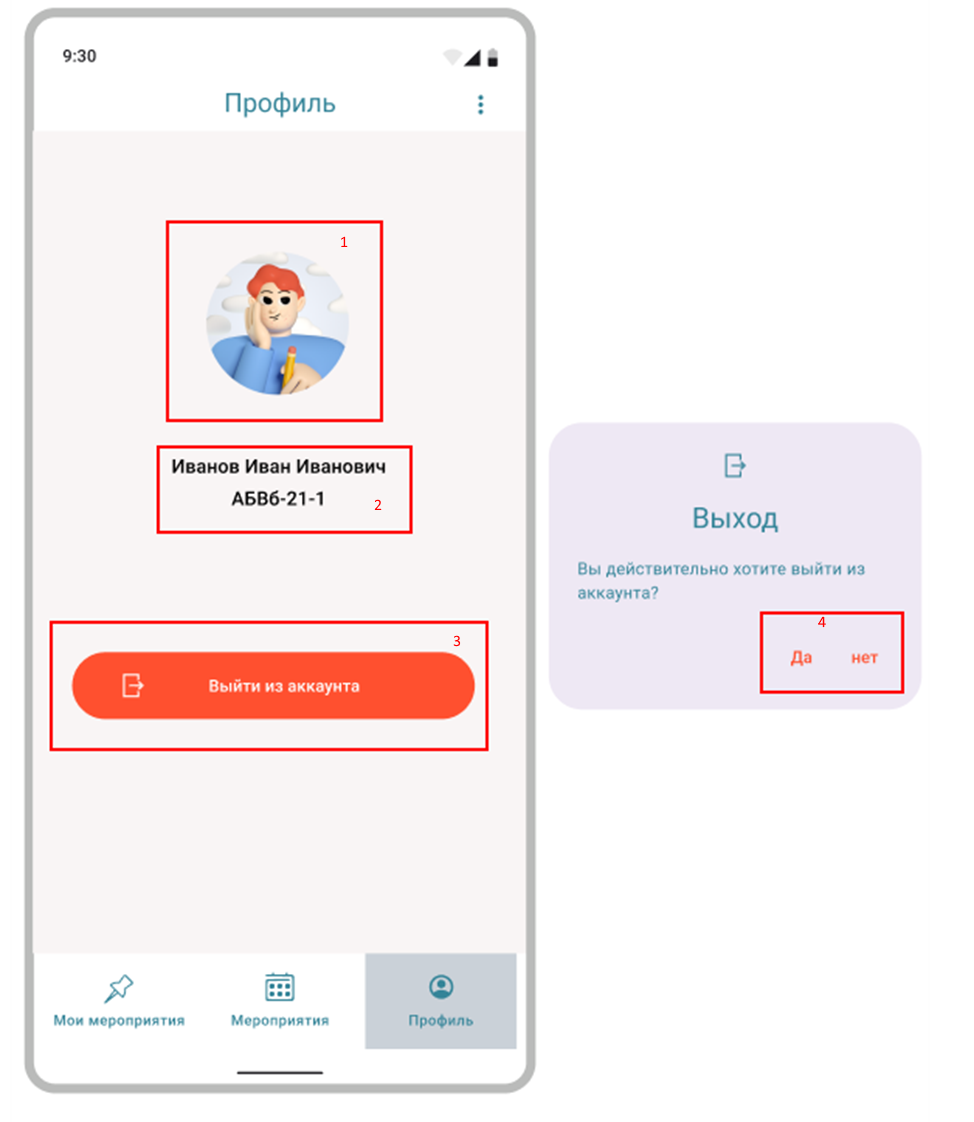


Рисунок 2.20 – Макет экрана «Профиль» (авторизованный)

**2.5.3.3 Описание экрана «Настройки»**

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

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

Макет данного экрана представлен на рисунке 2.21.

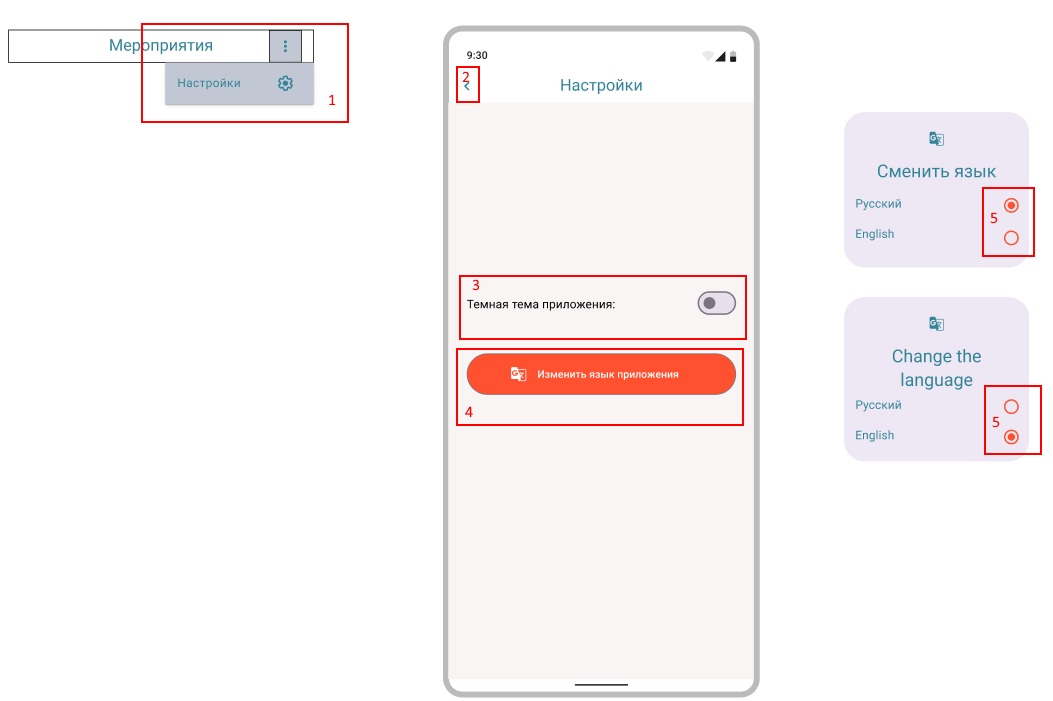


Рисунок 2.21 – Макет экрана «Настройки»

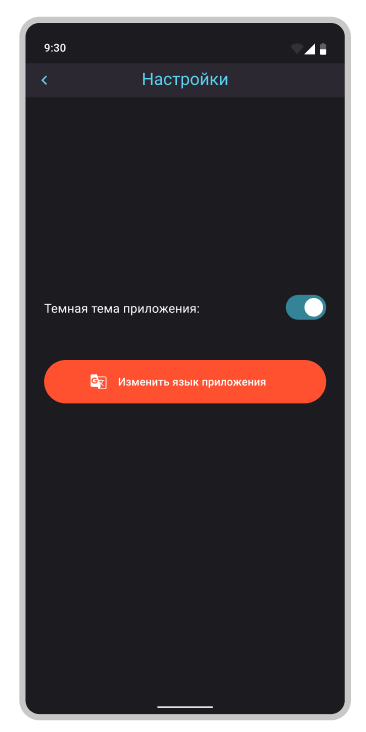


Рисунок 2.22 – Макет экрана «Настройки» (темная тема)