

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук  
Кафедра информационных систем

*Приложение «BoardGameMeet»*

*Курсовой проект*

*по дисциплине*

*Технологии программирования*

09.03.21 Информационные системы и технологии

Информационные системы и сетевые технологии

6 семестр 2022/2023 учебного года

Зав. Кафедрой	_____ д. т. н., профессор А. А. Сирота
Обучающийся	_____ Д. А. Карташов, 3 курс, д/о
Обучающийся	_____ И. М. Корольков, 3 курс, д/о
Обучающийся	_____ В. С. Егоров, 3 курс, д/о
Обучающийся	_____ Д. П. Корчагин, 3 курс, д/о
Руководитель	_____ В.С. Тарасов, ст. преподаватель. __.__.20__.

Воронеж 2023

Введение .....	4
1 Постановка задачи .....	5
2 Анализ предметной области.....	6
2.1 Используемые определения .....	6
2.2 Цель создания приложения.....	8
2.3 Сфера применения.....	8
2.4 Технический обзор .....	8
2.5 Обзор аналогов .....	9
2.5.1 GoBoards .....	9
2.5.2 Tabletop.Events .....	10
2.5.3 Meetup .....	12
2.5.4 Invitor.....	16
2.6 Требования к функциональности .....	17
2.7 Требования, не касающиеся функциональной части .....	17
2.8 Пользователи системы .....	18
2.9 Графическое описание работы системы .....	20
2.9.1 Диаграмма IDEF0 .....	20
2.9.2 Use Case диаграмма.....	22
2.9.3 Диаграмма классов .....	24
2.9.4 Диаграмма объектов.....	25
2.9.5 Диаграмма состояний.....	26
2.9.6 Диаграмма активности .....	27
3 Реализация приложения .....	30
3.1 Обоснование выбранных средств реализации .....	30

3.1.1 Backend.....	30
3.1.2 Frontend .....	31
3.2 Навигация по приложению .....	31
Заключение .....	51
Список используемых источников .....	52

## **Введение**

Настольные игры - это очень интересный и захватывающий способ скоротать досуг. За игрой можно потерять счет времени и надолго погрузиться в выдуманный мир. Но самая главная особенность в том, что игра может познакомить несколько совершенно, на первый взгляд, разных людей.

Единственная проблема в том, что не всегда так просто найти компанию для игры. Часто случается так, что у человека либо мало друзей, либо никто из них не любит играть в настольные игры. Из-за таких ситуаций и создаются приложения для организаций встреч по интересам, будь то танцы, обучение языкам или занятия спортом.

И BoardGameMeet не исключение. Это приложение создано с целью помочь людям, которые хотят провести свое свободное время за настольной игрой, найти соответствующую компанию. А если проблем с поиском людей, желающих поиграть не возникает, то приложение поможет организовать эту встречу, позволит договориться о дате и месте проведения и о количестве человек, необходимом для комфортной игры.

## 1 Постановка задачи

Задачей данной курсовой работы является разработка приложения для организации встреч людей, желающих сыграть в настольные игры.

Приложение должно отвечать следующим требованиям:

1. Приложение должно обладать следующими функциями:
  - 1.1. Создание мероприятий для организации встреч по настольным играм, возможность их редактировать и удалять;
  - 1.2. Возможность общения участников одного мероприятия в чате;
  - 1.3. Возможность присоединения к любому мероприятию, если оно активно и в нем есть свободные места;
  - 1.4. Возможность нахождения мероприятия по его названию;
  - 1.5. Возможность указывания места встречи;
  - 1.6. Возможность отмечания предметов, которые было бы хорошо взять с собой на встречу;
  - 1.7. Просмотр активных мероприятий для каждого пользователя;
  - 1.8. Просмотр пользователем своего личного профиля и профилей других пользователей, возможность редактировать свой.
2. Для реализации серверной части приложения должен использоваться язык программирования Kotlin;
3. Для реализации клиентской части приложения должен использоваться язык программирования Dart и его фреймворк Flutter.

## 2 Анализ предметной области

### 2.1 Используемые определения

**Фреймворк** - программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

**Язык программирования** - формальный язык, предназначенный для записи компьютерных программ, который определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (обычно — ЭВМ) под её управлением.

**Чат** - средство обмена сообщениями по компьютерной сети в режиме реального времени, а также программное обеспечение, позволяющее организовывать такое общение.

**Интерфейс** - это графическая оболочка приложения, с помощью которой пользователь взаимодействует с визуальными элементами программы, используя указатель, клавиатуру или сенсорный экран.

**Серверная часть** — это программно-аппаратная часть сервиса, которая хранится на сервере, обрабатывает полученные данные и отправляет ответ обратно.

**Kotlin** — статически типизированный, объектно-ориентированный язык программирования, работающий поверх Java Virtual Machine и разрабатываемый компанией JetBrains[1].

**Dart** — язык программирования, созданный Google. Dart позиционируется в качестве замены/альтернативы JavaScript[5, 6].

**Flutter** — комплект средств разработки и фреймворк с открытым исходным кодом для создания мобильных приложений под Android и iOS, веб-приложений, а также настольных приложений под Windows, macOS и Linux с

использованием языка программирования Dart, разработанный и развиваемый корпорацией Google[7].

**Docker** — язык программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений[8].

**Swagger** — это инструмент, использующий спецификацию OpenAPI.

**Смартфон** — мобильный телефон (современный — как правило, с сенсорным экраном), дополненный функциональностью карманного персонального компьютера.

**Ярлык** — файл, служащий указателем на объект (например, файл, который требуется определённым образом обработать), программу или команду и содержащий дополнительную информацию.

**Геолокация** — реальное географическое местоположение электронного устройства, например, радиопередатчика, сотового телефона или компьютера, подключенного к Интернету.

**Регистрация** — процесс или акт создания учётной записи пользователя.

**Виджет** — примитив графического интерфейса пользователя, имеющий стандартный внешний вид и выполняющий стандартные действия.

**Социальная сеть** — онлайн-платформа, которая используется для общения, знакомств, создания социальных отношений между людьми, которые имеют схожие интересы или офлайн-связи, а также для развлечения (музыка, фильмы) и работы.

**Виртуальное мероприятие** — это мероприятие любого типа, проводимое в режиме онлайн, а не в реальном помещении.

**Онлайн** — деятельность, служба, которая доступна исключительно через Интернет.

**Фильтр поиска** — это сервис на сайте, который облегчает пользователю поиск необходимого товара или информации.

**Гость** — неавторизованный пользователь.

**Скриншот** — это снимок экрана компьютера или мобильного устройства, сделанный с помощью стандартных средств или специальной программы.

**Тег** — это метка, которая классифицирует данные, чтобы облегчить интернет-пользователям процесс поиска.

**Главное меню** — это один из элементов командного интерфейса программы, расположенный в главной панели основного окна программы.

## **2.2 Цель создания приложения**

Цель создания приложения - помощь в организации встреч участников настольных игр в реальной жизни.

## **2.3 Сфера применения**

Приложение для организации встреч участников настольных игр может быть использовано в сфере организации различных мероприятий по настольным играм. Оно позволяет создавать и настраивать свои мероприятия, к которым могут присоединяться участники и общаться в чате мероприятия.

## **2.4 Технический обзор**

- Создание мероприятий и участие в них. Пользователь имеет возможность создавать мероприятия или присоединяться к уже существующим;
- Общение в чате. Участники мероприятия могут общаться в чате этого мероприятия;
- Нужные предметы. Создатель мероприятия может добавлять в специальный список предметы, которые нужно взять с собой на мероприятие.



## 2.5 Обзор аналогов

### 2.5.1 GoBoards

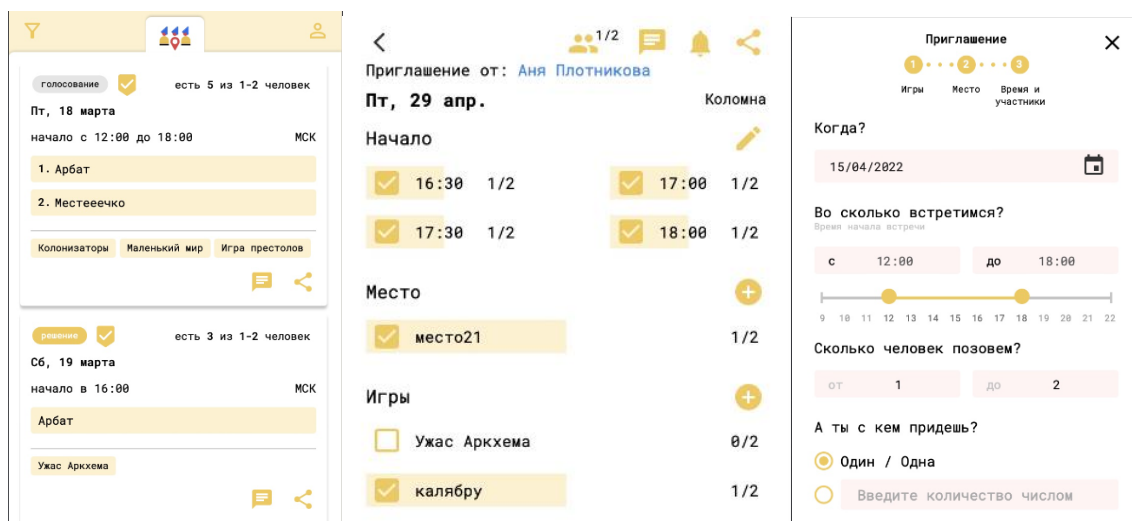


Рисунок 1 - GoBoards

Почти единственное в своем роде приложение для организации встреч, заточенных именно под настольные игры. Создано командой разработчиков из Москвы. Стоит отметить, что изначально был создан одноименный сайт с таким же функционалом, а после создана версия под смартфон, которая, по сути является ярлыком вышеуказанного сайта. По информации от разработчиков, вначале своего существования им удалось вступить в сотрудничество с некоторыми антикафе, которые продвигали их приложение. Но после непродолжительного времени им пришлось прекратить его поддержку, из-за невозможности монетизации. На данный момент приложение не работает.

Преимущества:

- Возможность голосовать за время, игру и место встречи внутри каждого мероприятия;
- Возможность отметить, что на встречу вы придете не одни;

- Местом встречи можно отметить антикафе из предложенного списка и указать стоимость нахождения в нем;
- Интуитивно понятный, не перегруженный интерфейс.
- Недостатки:
  - Невозможность задать место сбора с помощью геолокации;
  - Отсутствие чата внутри мероприятия;
  - Отсутствие возрастного ограничения для мероприятий.

### 2.5.2 Tabletop.Events

Зарубежное приложение для организации мероприятий и конференций по настольным играм. Создано в первую очередь для помощи в управлении крупными событиями. Дает возможность очень гибкой настройки мероприятий. Вплоть до аренды помещений, регистрации участников, планирования событий на несколько дней и многое другое. Также включает в себе возможность гибкой настройки дизайна конференции (различные шрифты, фон, добавление видеороликов).

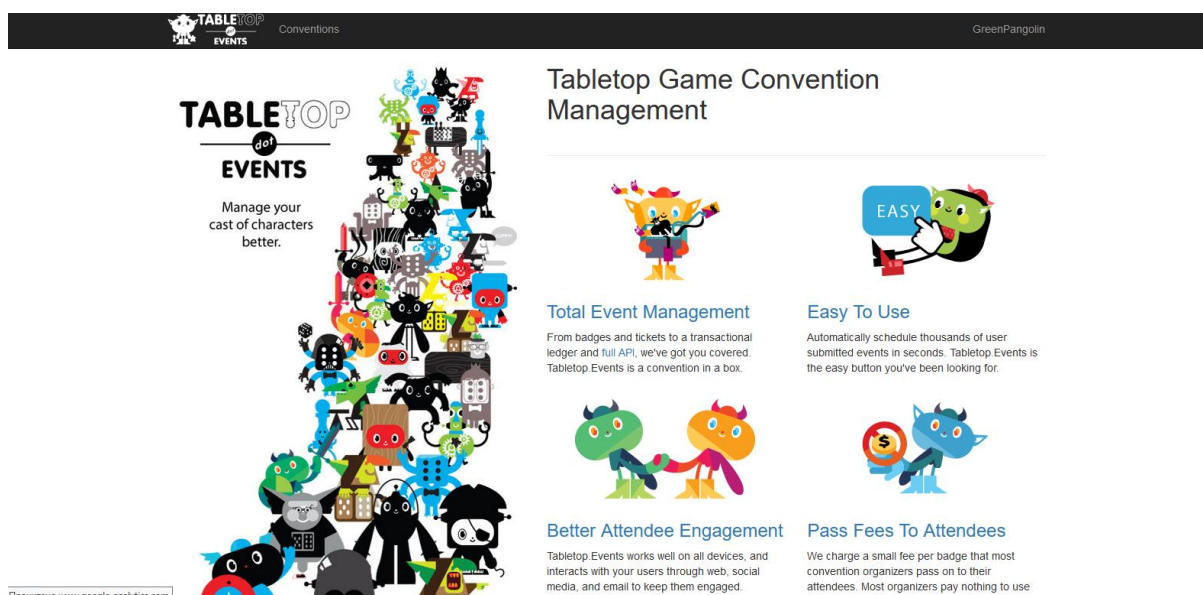


Рисунок 2 - Начальная страница

Данное приложение имеет ряд недостатков:

- Огромное количество настроек. Для создания мероприятия требуется указать большое количество разнообразных параметров. Приложение переполнено различными возможностями, поэтому создать мероприятие быстро вряд ли получится;
- Отсутствие русского языка. В купе с предыдущим минусом, освоение данного приложения многократно усложняется;
- Большая часть мероприятий платная. Приложение рассчитано в основном на управление крупными событиями, в которых предусмотрена платная регистрация.

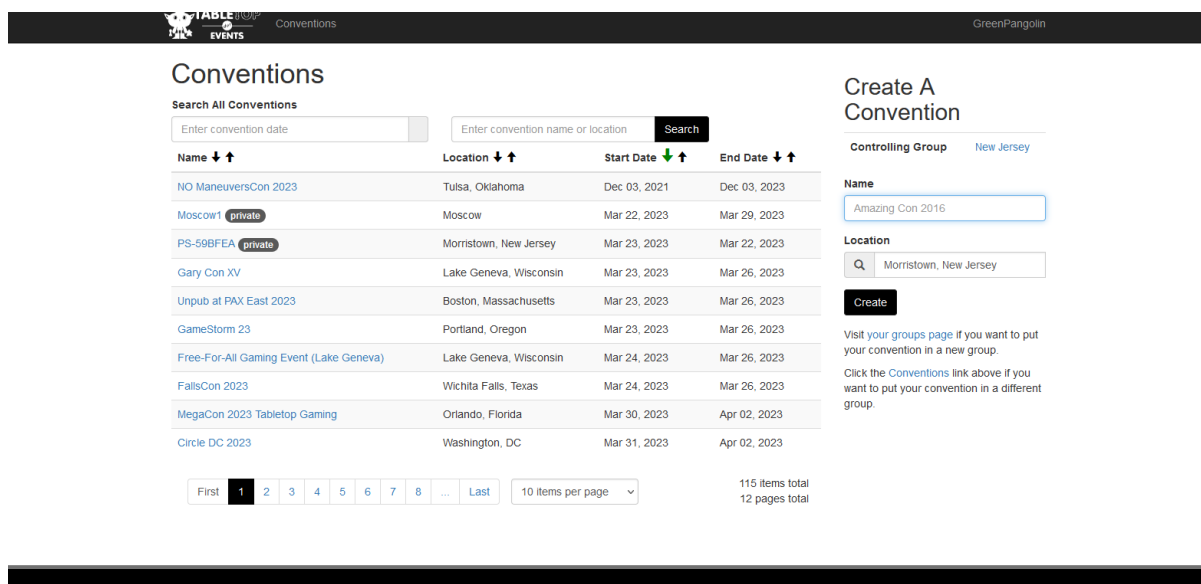


Рисунок 3 - Страница со списком мероприятий

Приложение обладает следующими преимуществами:

- Гибкая настройка и различные ограничения. Например, ограничения по возрасту. Это позволяет лучше планировать крупные события;

- Настройка страницы мероприятия. Создатель мероприятия может всячески кастомизировать страницу с представлением события, добавляя туда разнообразные виджеты;
- Показ места проведения мероприятия на карте;
- Помощь в регистрации участников (продажа билетов и т.п.).



Рисунок 4 - Страница мероприятия

Данное приложение является отличным инструментом для организации крупных мероприятий по настольным играм, но плохо подходит для дружеских посиделок или поиска компании.

### 2.5.3 Meetup

Представляет собой платформу социальных сетей для размещения и организации личных и виртуальных мероприятий, собраний и мероприятий для людей и сообществ со схожими интересами, хобби и профессиями.

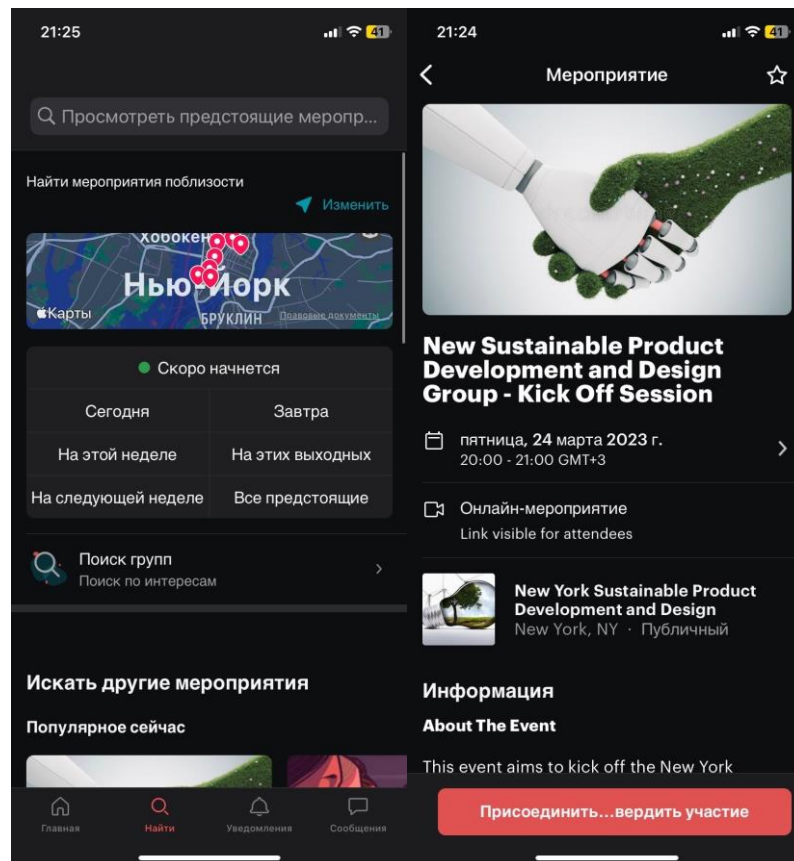


Рисунок 5 - Meetup

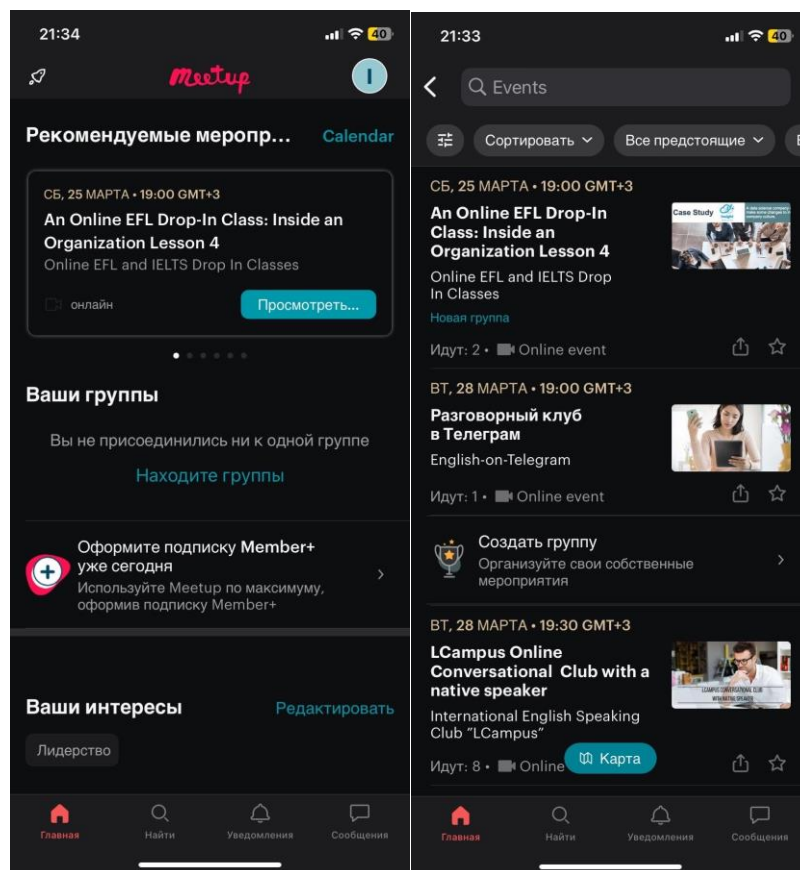


Рисунок 6 - Meetup

В приложении можно отметить такие недостатки как ограниченный функционал для пользователей не имеющих платную подписку и некоторые моменты, связанные с неаккуратность интерфейса. Однако можно выделить такие плюсы, как множество фильтров при поиске мероприятий и наличие большого количества опций для участников.

Недостатки:

- При входе в приложения вам сразу же предлагается войти или зарегистрироваться, хотя можно войти как гость;
- Геолокация по умолчанию определяется в Нью-Йорке США;
- Конкретно информационный блок мер безопасности мероприятия сокращен и нет возможности просмотреть текст полностью;

- Без платной подписки ограничено число проведенных мероприятий и невозможно создать группу.

#### Преимущества:

- К местоположению прилагается скриншот спутниковой карты с обозначением места проведения мероприятия;
- Мероприятия могут создавать как отдельные люди, так и группы; Присутствует возможность просмотра мероприятий отдельной группы;
- При просмотре мероприятий показываются похожие события, проходящие поблизости;
- Существует возможность добавить мероприятие в избранное;
- Возможность сортировки мероприятий по разным параметрам (дата, интересы и т.д.);
- Возможность написать личное сообщение участнику или организатору мероприятия;
- Проведение мероприятий онлайн.



## 2.5.4 Invitor

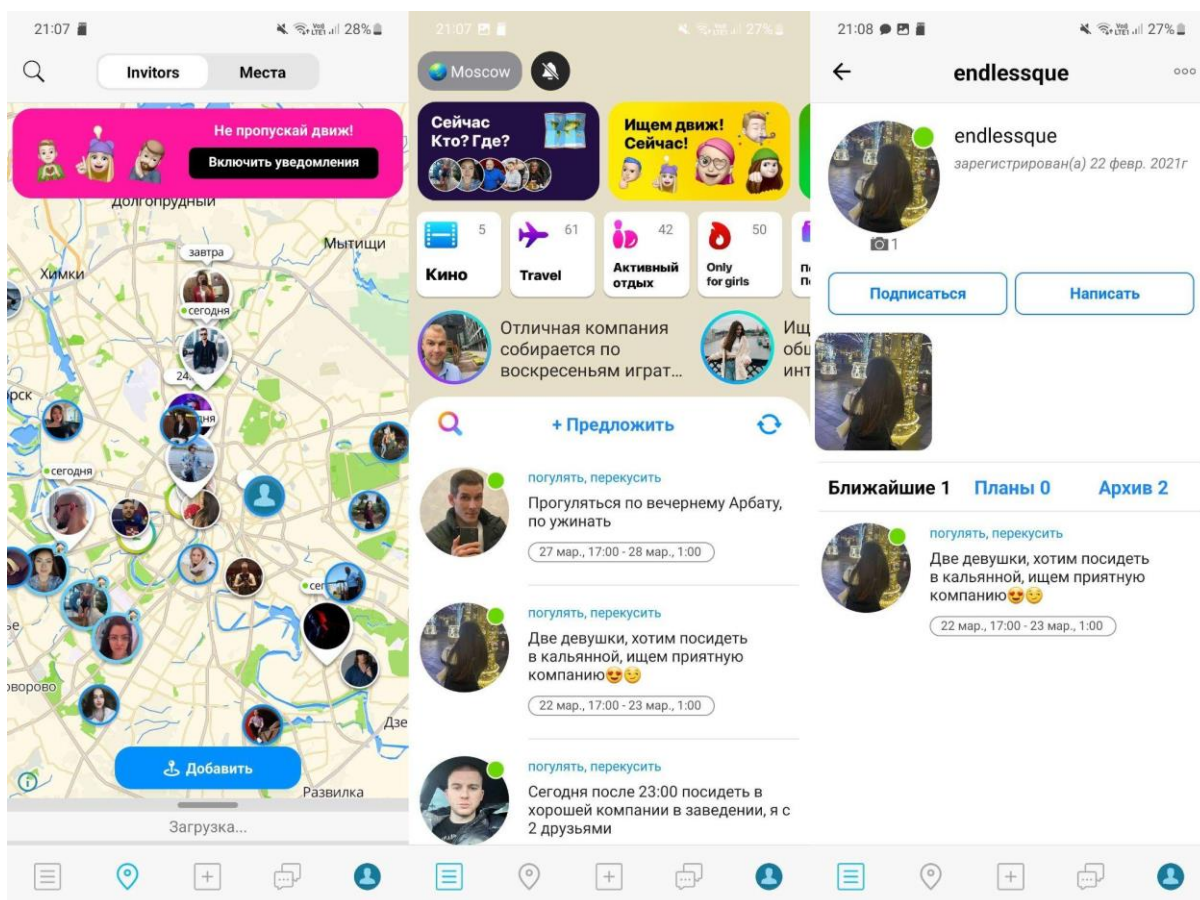


Рисунок 7 - Invitor

Приложение сделано для поиска компании для различных мероприятий, нет функций для организации. Можно создать комнату с различной видимостью в которой есть только описание и чат, комнаты можно находить по списку и на карте. Также можно искать по тегу, но только через список. Есть отдельная вкладка с чатами.

Недостатки:

- В России аудитория есть только в Москве, в др. странах тоже только в крупных городах;
- Есть тег only girls, но нет only boys (тегов всего 10);



- Возрастной диапазон ограничен только снизу;
- Чат внутри мероприятия виден всем.

Преимущества:

- Удобный поиск мероприятий по карте;
- Можно смотреть события в других городах (удобное переключение между городами);
- Можно выключить уведомления через главное меню.

## **2.6 Требования к функциональности**

- Создание и редактирование мероприятий;
- Присоединение к мероприятию и выход из него;
- Редактирование списка необходимых предметов;
- Просмотр участников мероприятия;
- Общение в чате;
- Возможность создателя мероприятия удалять созданное мероприятие;
- Возможность создателя мероприятия удалять участников из мероприятия.

## **2.7 Требования, не касающиеся функциональной части**

Технологические требования:

- Приложение должно устанавливаться и работать на мобильных устройствах версий Android 10 и выше и имеющих доступ к сети Интернет;
- Приложение должно иметь архитектуру, соответствующую шаблону Business Logic Component (BLoC);
- Серверная часть приложения должна быть реализована на языке Kotlin версии 1.8.0;

- Клиентская часть приложения должна быть реализована на языке Dart версии 2.19, используя фреймворк Flutter версии 3.7.9;
- В роли СУБД должна быть использована PostgreSQL 15 версии;
- Для документирования API должен использоваться Swagger;
- Для развертывания приложения должен использоваться Docker;
- Для определения местоположения пользователя плагин Yandex Maps (yandex\_mapkit) версии 3.2.0.

Дизайн приложения должен быть выполнен в темных тонах.

Основные цвета:

- Оттенок черного (#292929) - цвет фона приложения;
- Оттенок зеленого (#50bc55) - цвет большинства кнопок;
- Белый (ffffff) - цвет текста, фон мероприятия и чата;
- Оттенок синего (#4b84f1) - цвет кнопки восстановления пароля и некоторых небольших элементов.

Вторичные цвета:

- Оттенок черного (#171717) - цвет поисковой строки;
- Красный (ff0000) - цвет стрелок и крестов.

В ходе разработки допускается добавление новых цветов или удаление старых, если это не меняет общий стиль приложения.

## 2.8 Пользователи системы

В приложении существуют такие группы пользователей, как авторизованные пользователи, неавторизованные пользователи, создатели мероприятий и администраторы.

Возможности неавторизованных пользователей:

- Просмотр списка мероприятий;
- Просмотр информации о мероприятии;
- Регистрация;

— Авторизация.

Возможности авторизованного пользователя:

- Просмотр списка мероприятий;
- Просмотр информации о мероприятии;
- Вступление в мероприятие;
- Редактирование данных своего профиля;
- Выход из профиля;
- Общение в чате мероприятия;
- Просмотр профилей других участников;
- Выход из мероприятия;
- Просмотр собственных мероприятий.

Возможности создателя мероприятий:

- Просмотр списка мероприятий;
- Просмотр информации о мероприятии;
- Вступление в мероприятие;
- Редактирование данных своего профиля;
- Выход из профиля;
- Общение в чате мероприятия;
- Просмотр профилей других участников;
- Выход из мероприятия;
- Просмотр собственных мероприятий;
- Исключение игроков из своих мероприятий;
- Редактирование параметров своих мероприятий;
- Удаление своих мероприятий.

Возможности администратора:

- Просмотр списка мероприятий;
- Просмотр информации о мероприятии;
- Вступление в мероприятие;

- Редактирование данных своего профиля;
- Выход из профиля;
- Общение в чате мероприятия;
- Просмотр профилей других участников;
- Выход из мероприятия;
- Просмотр собственных мероприятий;
- Исключение игроков из своих мероприятий;
- Редактирование параметров своих мероприятий;
- Удаление своих мероприятий.

## **2.9 Графическое описание работы системы**

### **2.9.1 Диаграмма IDEF0**

Рассмотрим работу приложения на примере диаграммы IDEF0.

Функциональный блок первого уровня диаграммы IDEF0 представлен в виде Участия в мероприятии пользователем (рисунок 8).

Управление осуществляет список существующих мероприятий.

Входным параметром является пользователь, желающий принять участие в мероприятии.

Механизмом осуществления работы данного процесса являются:

- Приложение, которое осуществляет хранение мероприятия в базе данных;
- Администратор, который проверяет информацию о данном мероприятии;
- Создатель мероприятия, который является организатором данного мероприятия.

Результатом работы системы на данном уровне является участие пользователя в выбранном мероприятии или блокировка пользователя.

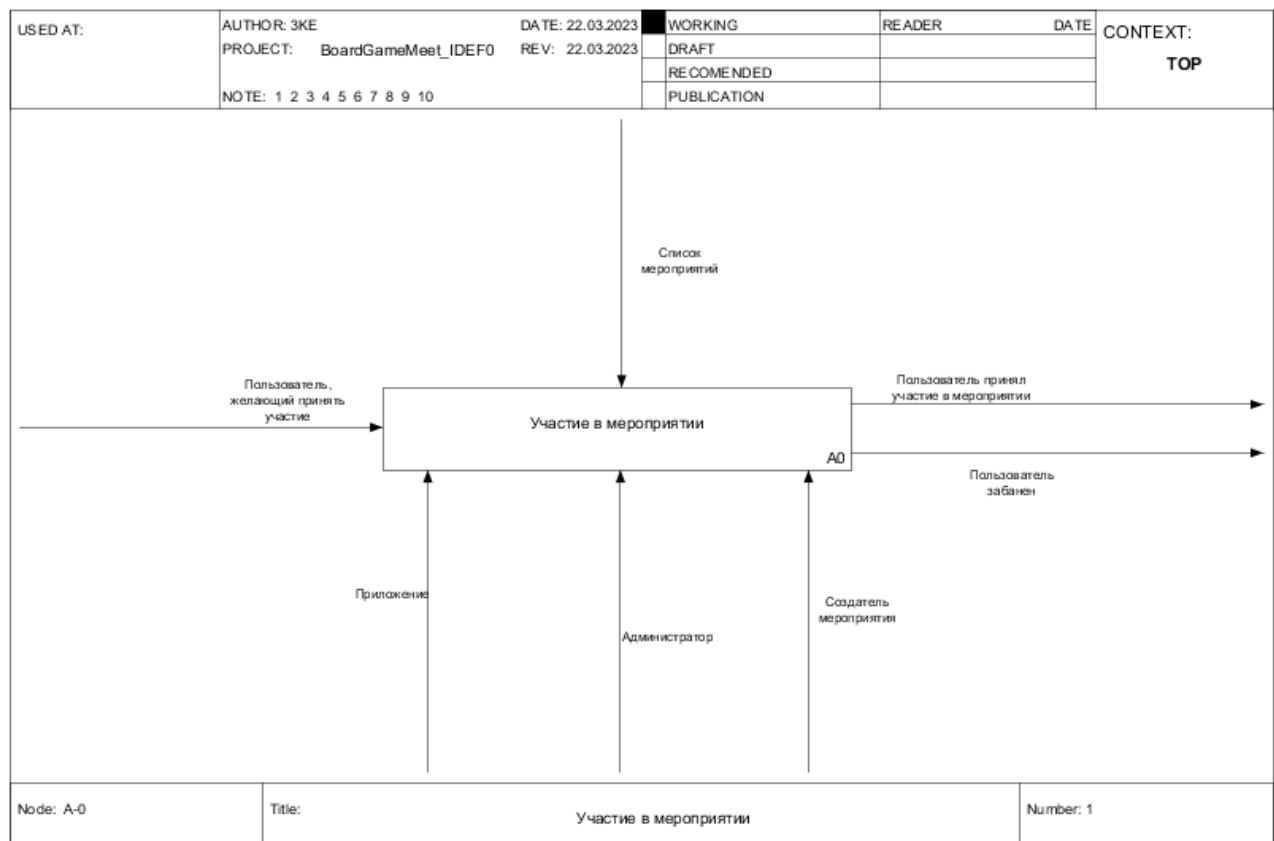


Рисунок 8 - Первый уровень диаграммы IDEF0

Далее представлен второй уровень диаграммы IDEF0, на котором отображена более подробная схема работы системы (рисунок 9).

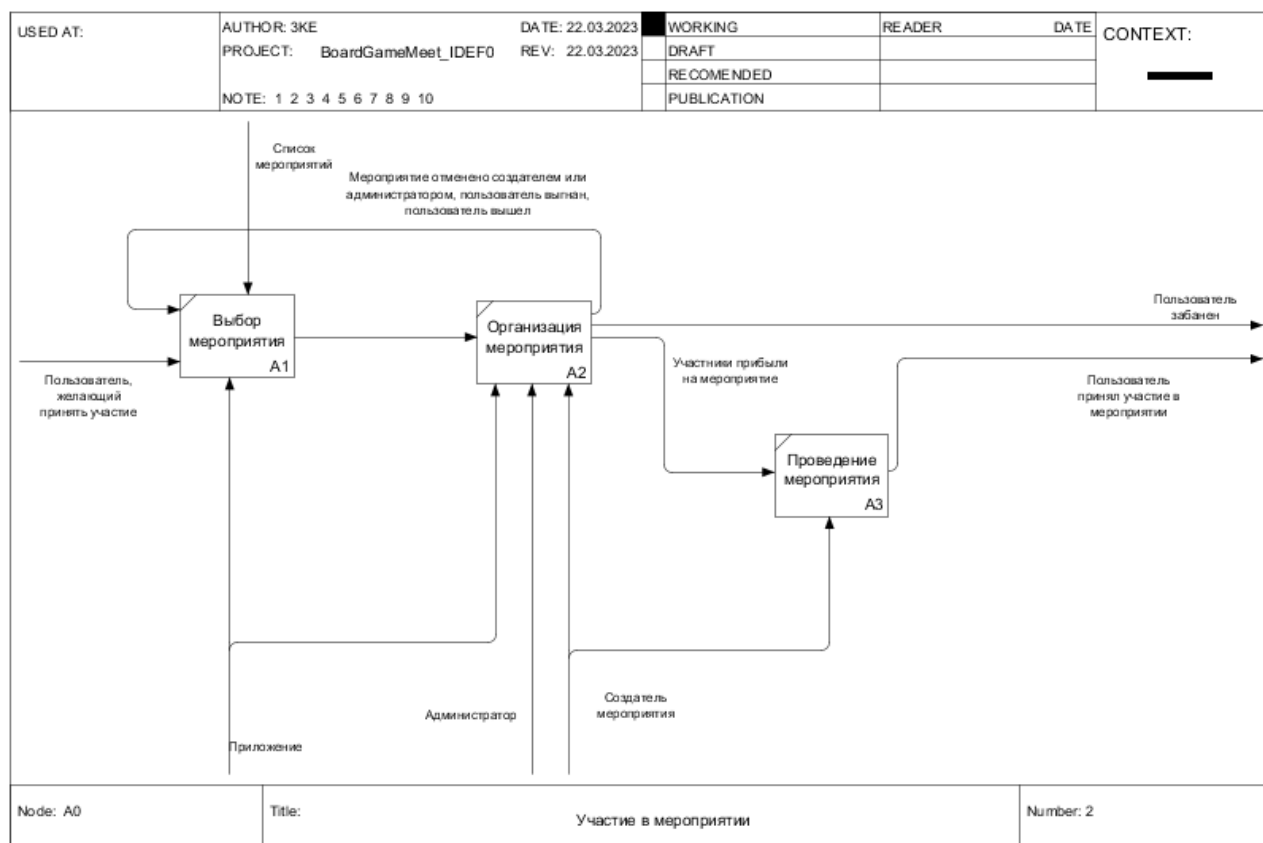


Рисунок 9 - Второй уровень диаграммы IDEF0

### 2.9.2 Use Case диаграмма

Функциональность и возможности программы описывает Use Case диаграмма (рисунок 10).

На диаграмме мы можем увидеть несколько актеров: гость, пользователь, создатель мероприятия и администратор.

Далее рассмотрим возможности каждого актера.

Гость может:

- Зарегистрироваться в приложении;
- Просматривать список мероприятий;
- Просматривать информацию о конкретных мероприятиях.

Пользователь может:

- Просматривать список мероприятий;
- Просматривать информацию о конкретных мероприятиях;
- Редактировать данные своего профиля;
- Выйти из профиля;
- Просматривать активные мероприятия;
- Вступить в активные мероприятия;
- Просматривать профили других участников мероприятия;
- Выйти из мероприятия.

Создатель мероприятия может:

- Создать мероприятие;
- Составить список предметов, требующихся для мероприятия;
- Исключить пользователей из мероприятия;
- Редактировать данные активных мероприятий ;
- Удалить мероприятие.

Администратор может:

- Блокировать пользователей;
- Просматривать информацию о мероприятиях;
- При необходимости удалять мероприятия.

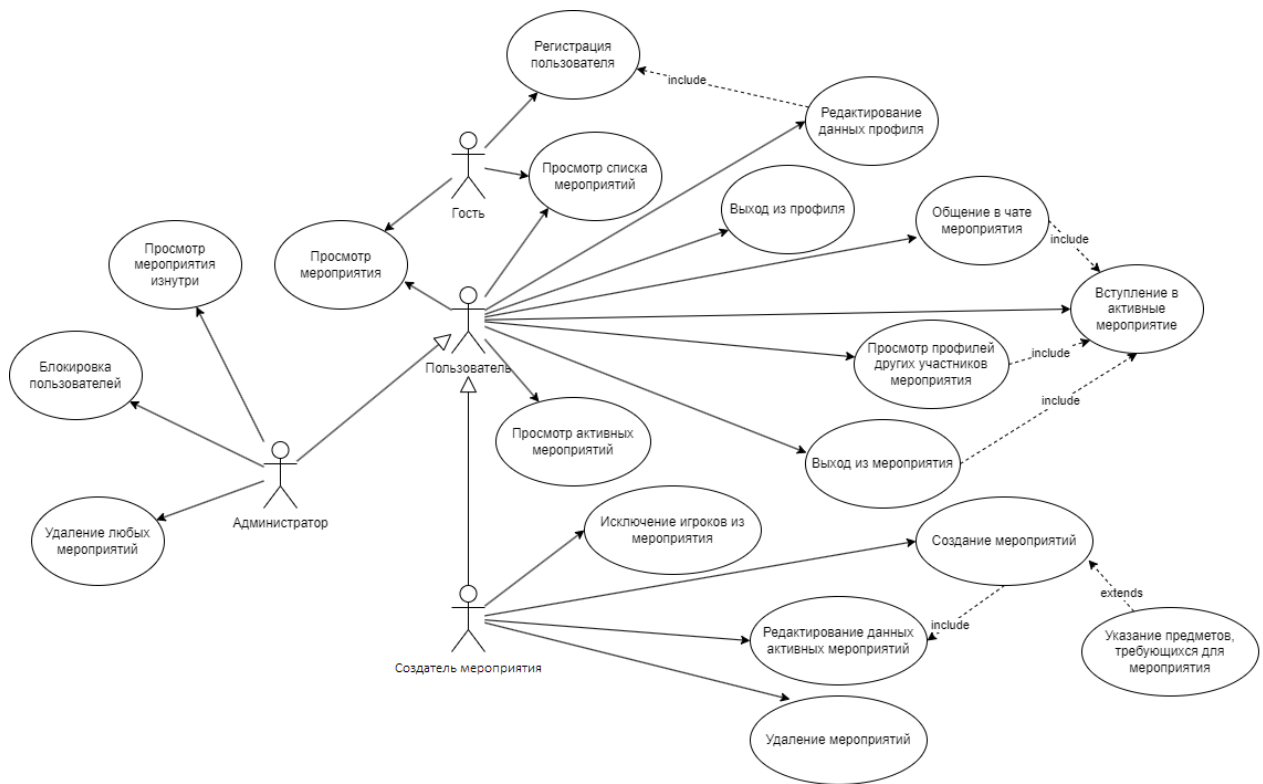


Рисунок 10 - Use Case диаграмма

### 2.9.3 Диаграмма классов

Представление внутренней структуры программы отражено в диаграмме классов (рисунок 11).

В данной системе диаграмма представлена шестью классами:

- Person описывает пользователей приложения
- Event описывает мероприятия
- Message описывает сообщения и используется для чатов мероприятий
- Item описывает предметы для списка требуемых предметов
- Role описывает роли пользователей и используется для предоставления определенных возможностей



— Token представляет набор данных и используется для хранения информации о пользователях в базе данных и осуществления проверок авторизации пользователей.



Рисунок 11 - Диаграмма классов

## 2.9.4 Диаграмма объектов

Диаграмма объектов является экземпляром диаграммы классов и предназначена для демонстрации совокупности моделируемых объектов и связей между ними в фиксированный момент времени (рисунок 12).

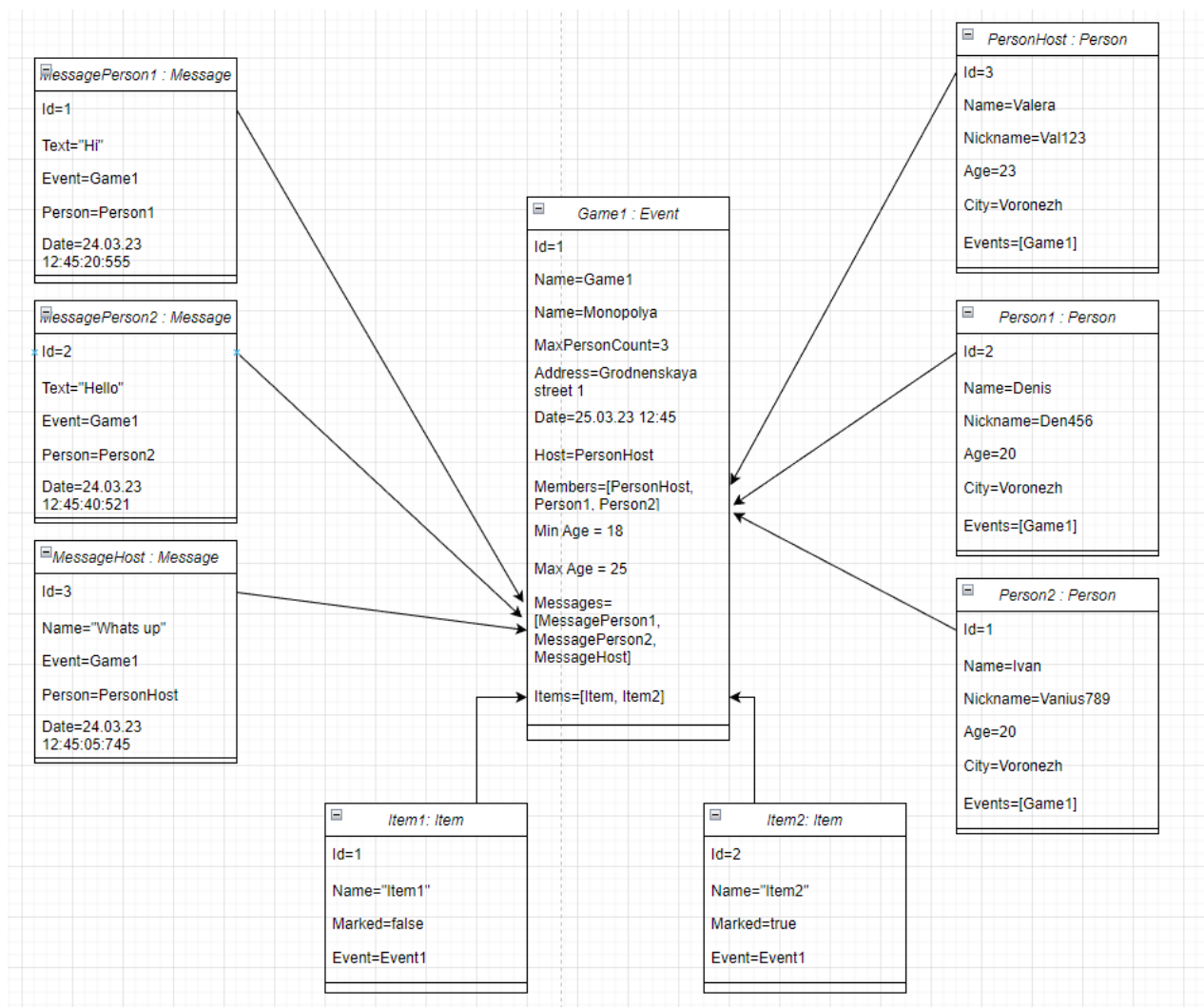


Рисунок 12 - Диаграмма объектов

### 2.9.5 Диаграмма состояний

Диаграмма состояний определяет, как объект переходит из одного состояния в другое. Рассмотрим диаграмму состояний мероприятия (рисунок 13).

Во время работы приложения мероприятие проходит пять состояний:

- Мероприятие создано, после добавлении мероприятия создателем;
- Мероприятие организовывается, во время того, как пользователи присоединяются к данному мероприятию;

- Мероприятие готово к проведению, после того как время на присоединение прошло;
- Мероприятие проведено, после того как создатель провел данное мероприятие;
- Мероприятие завершено, после того как создатель завершает данное мероприятие.



Рисунок 13 - Диаграмма состояний

### 2.9.6 Диаграмма активности

Диаграмма активности для мероприятия отображает действия, состояния которых описаны в диаграмме состояний, и помогает описать логику поведения системы (рисунок 14).

Активности на диаграмме расположены в трёх колонках, каждая из которых соответствует поведению одного из трех объектов — Пользователя, Системы и Создателя мероприятия. Благодаря этому легко определить, каким из объектов выполняется каждая из деятельности.

Сценарий работы приложения, описанный в диаграмме активностей мероприятия, включает в себя следующие пункты:

- Создатель мероприятия создает мероприятие;

- Система добавляет новое мероприятие в базу данных;
- Система обновляет список мероприятий;
- Пользователь выбирает мероприятие из списка;
- Если мероприятие не подходит пользователю, то он возвращается к выбору мероприятия;
- Если мероприятие подходит пользователю, то он осуществляет попытку вступления в данное мероприятие;
- Если пользователь не авторизован, то система перекидывает его на страницу авторизации. После этого пользователь возвращается к выбору мероприятия;
- Если пользователь авторизован, то система добавляет пользователя в данное мероприятие;
- Создатель проводит мероприятие;
- Создатель завершает мероприятие ;
- Система удаляет мероприятие.

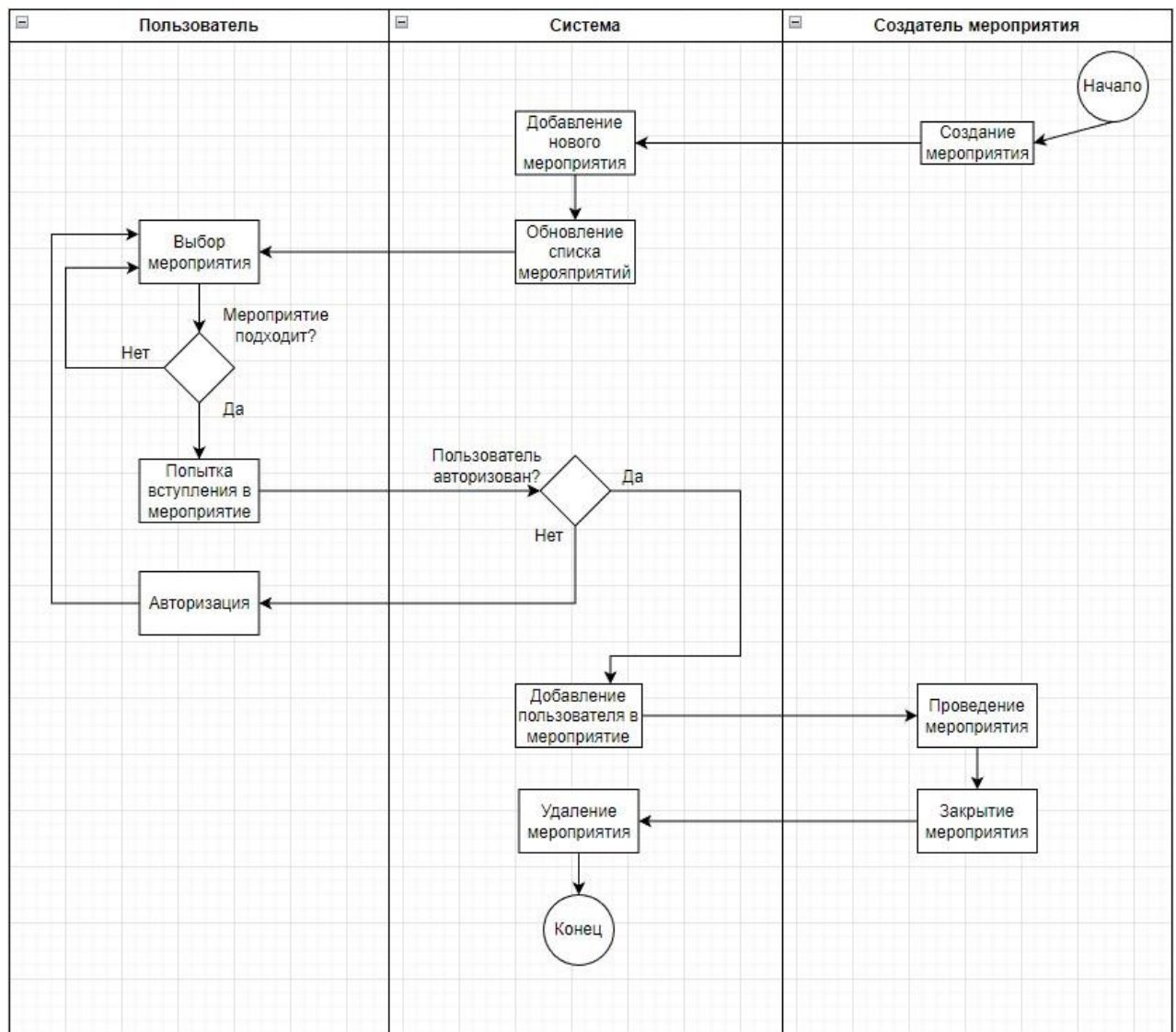


Рисунок 14 - Диаграмма активности

### **3 Реализация приложения**

При разработке приложения использовалась методология Водопадная модель. Она предусматривает последовательное выполнение работ, сначала анализ, затем проектирование, разработка, тестирования.

#### **3.1 Обоснование выбранных средств реализации**

##### **3.1.1 Backend**

Язык программирования Kotlin полностью совместим с языком программирования Java. Это означает, что в процессе разработки серверной части, можно будет пользоваться всеми фреймворками, которые совместимы с Java. Помимо всего прочего, Kotlin во многом лаконичнее и удобнее, чем Java, имеет в себе такие полезные особенности, как, например, более понятные лямбда-выражения, аргументы по умолчанию, именованные аргументы, ключевое слово `data`, которое сильно облегчает создание DTO сущностей, автоматически генерируя геттеры и сеттеры для всех полей класса.

Основным фреймворком для серверной части приложения выбран Spring, который является очень мощным инструментом разработки, который упрощает организацию общения клиента и сервера посредством REST API и не только, и имеет такие полезные модули:

- Spring Boot, отвечающий за управление зависимостями и автоматическую конфигурацию приложения;
- Spring Data JPA, облегчающий работу с базой данных для разработчика;
- Spring Security, который был использован для создания авторизации и регистрации пользователей;
- Spring Websocket, который использовался для организации работы чата.

В качестве базы данных выбрана Postgresql, потому что она легка в использовании и предоставляет исчерпывающее количество инструментов и типов данных для использования в разработке.

Также для создания образов серверной части приложения и базы данных был использован Docker, который является удобным инструментом упаковки приложения со всем его окружением и зависимостями в контейнер.

В качестве облачного сервиса, на котором будет развернута серверная часть приложения был выбран Okteto. Этот сервис позволяет полноценно развернуть серверную часть приложения и не затрудняет ее общение с клиентской частью.

### **3.1.2 Frontend**

Клиентская часть приложения написана на языке Dart с использованием фреймворка Flutter. Это выбор был сделан из-за того, что Flutter автоматизирует многие задачи разработки, например, горячую перезагрузку кода. Также преимущество разработки с использованием Flutter это использование единственного языка программирования, который достаточно прост в освоении. Flutter представляет собой современный инструмент для разработки красивых и высокопроизводительных кроссплатформенных приложений. Фреймворк предоставляет большее количество готовых виджетов и библиотек, которые значительно упрощают разработку.

Клиентская часть приложения написана с использованием архитектуры BLoC. Данная архитектура используется для построения приложений, которые должны быть быстрыми и реактивными, отлично подходит для Flutter приложений.

## **3.2 Навигация по приложению**

При первом входе в приложение пользователь попадает на приветственные экраны, на которых кратко объясняется то, как пользоваться приложением (рисунок 15).

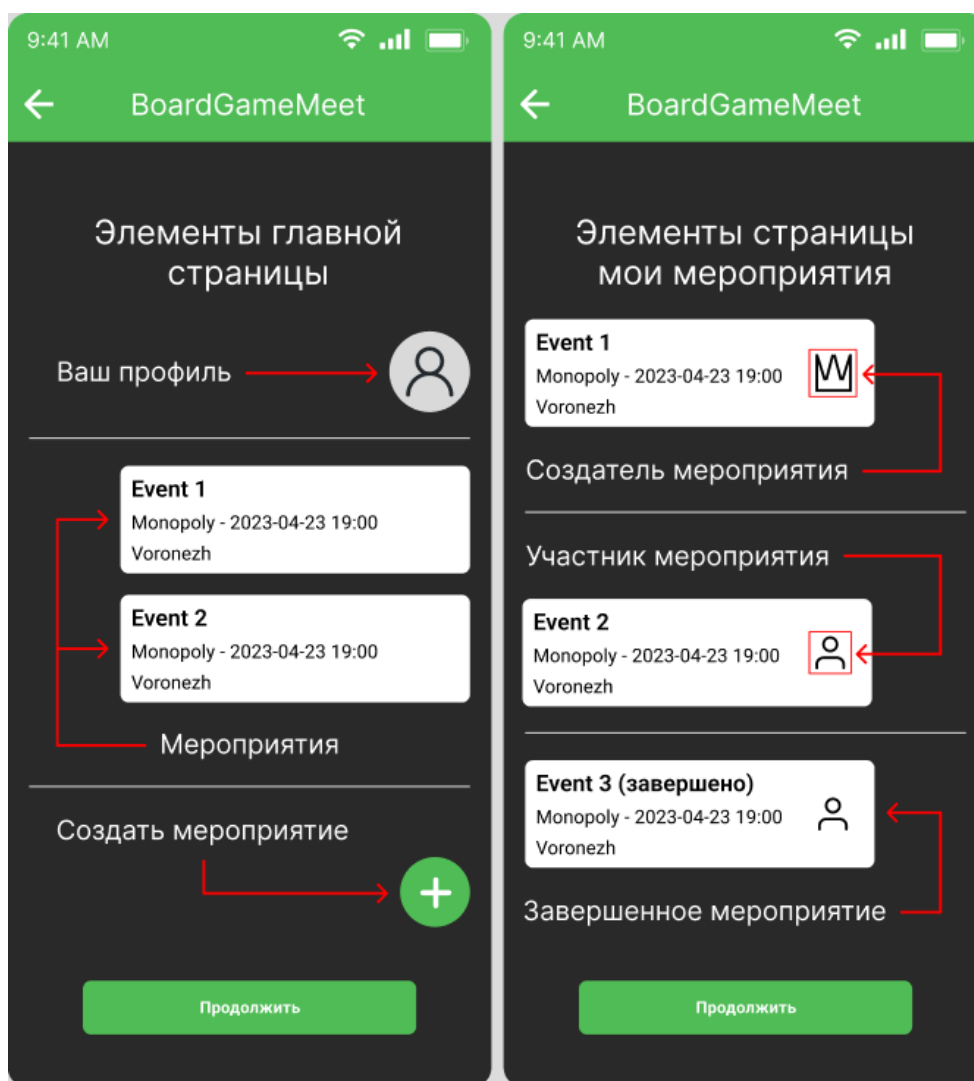


Рисунок 15 - Приветственные экраны

Далее незарегистрированный пользователь попадает на экран с мероприятиями, в которые может вступить (рисунок 16). С него он может перейти на карточку каждого из доступных мероприятий и посмотреть информацию о нем или авторизоваться. При попытке перейти на экран «Мои



мероприятия» или «Создание мероприятия» пользователь будет перенесен на экран авторизации.

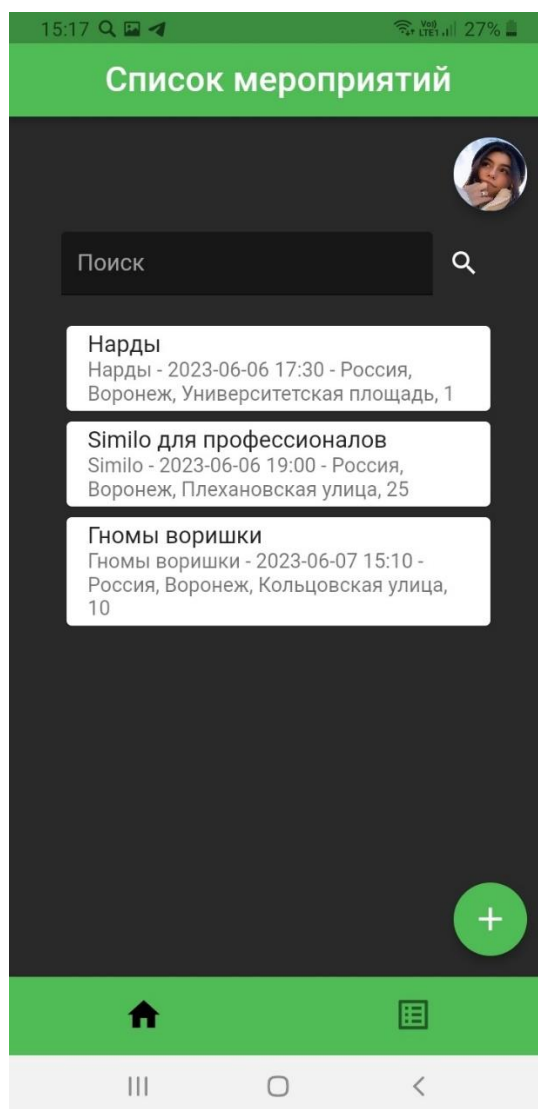


Рисунок 16 - Экран списка мероприятий

Находясь на экране авторизации, пользователь может войти в свой аккаунт, зарегистрироваться или восстановить пароль (рисунок 17).

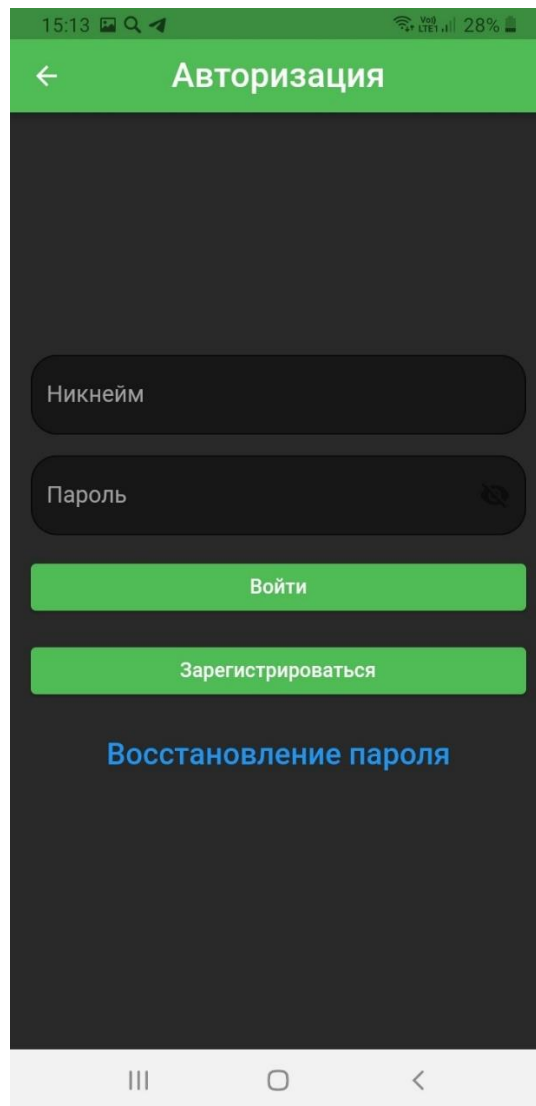


Рисунок 17 - Экран авторизации

При регистрации пользователь заполняет обязательные поля, помеченные звездочкой, а также, при желании, необязательные поля (возраст и пол), которые он сможет заполнить в любое время в своем профиле (рисунок 18).

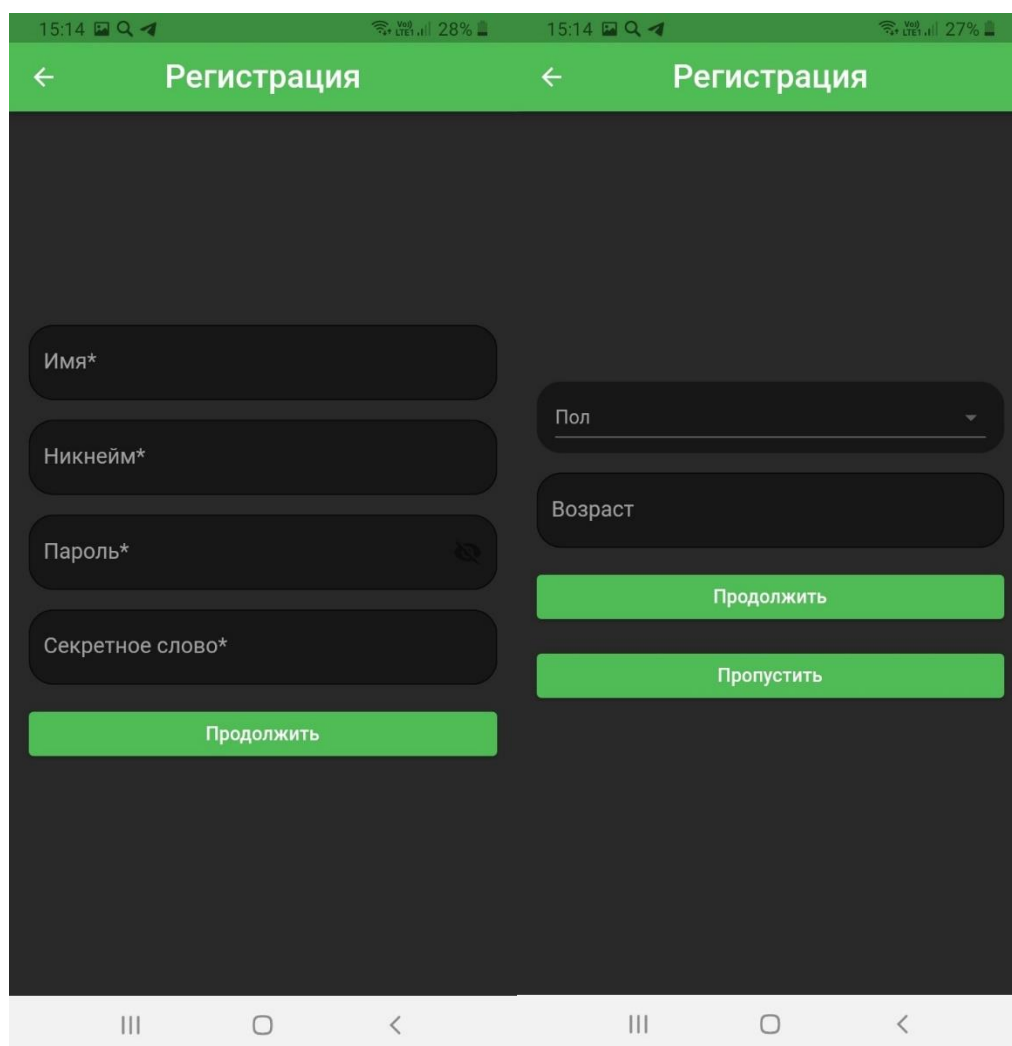


Рисунок 18 - Экраны регистрации

Если пользователь забыл свой пароль, он может воспользоваться функцией восстановления пароля (рисунок 19). Пользователь вводит свой никнейм и секретное слово для идентификации личности и, после успешной проверки, выбирает новый пароль.

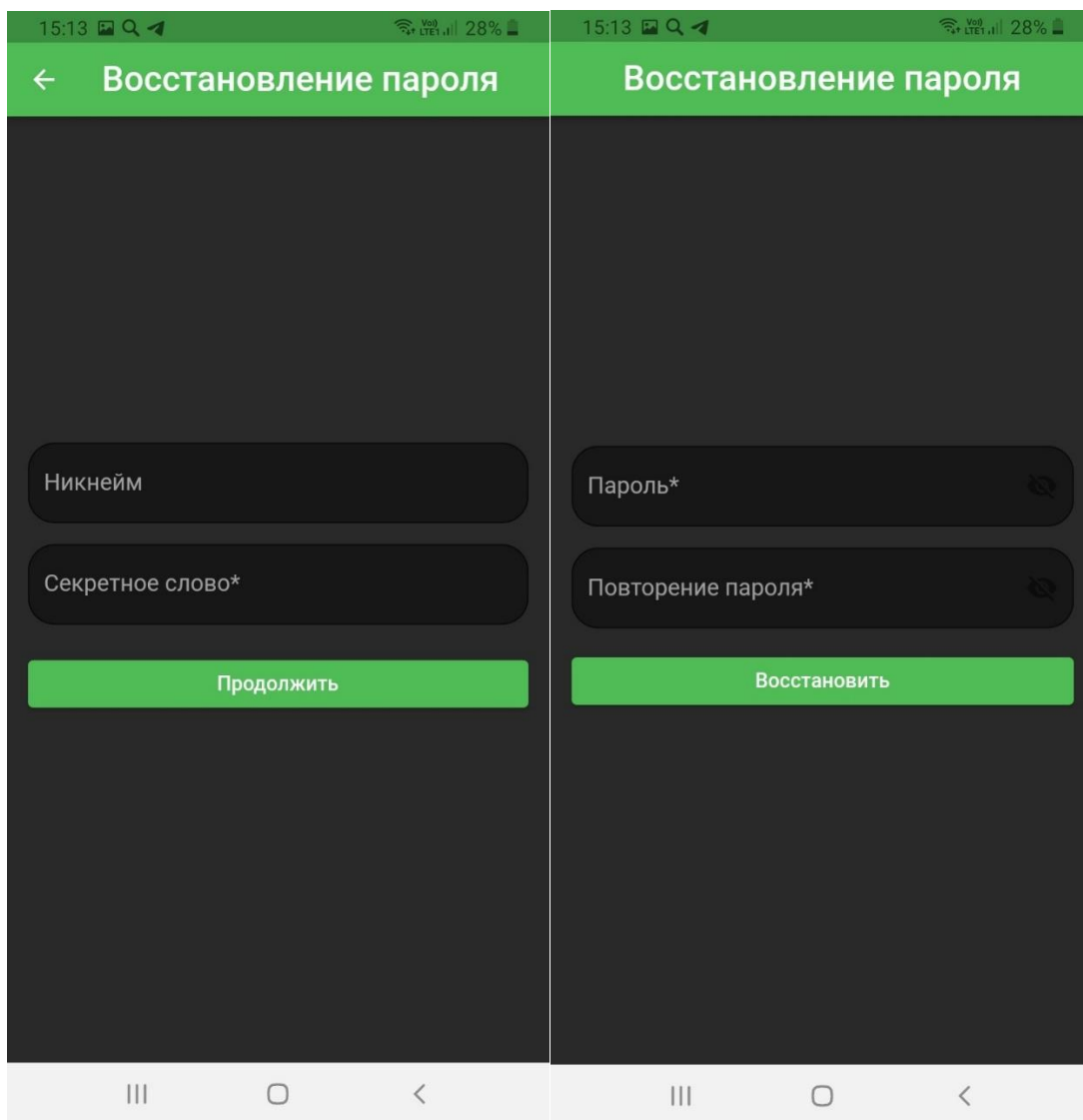


Рисунок 19 - Экраны восстановления пароля

Экран профиля другого пользователя содержит краткую информацию о нем: имя, никнейм, город. Также, при заполнении пользователем, на экране может отображаться его возраст и пол (рисунок 20).

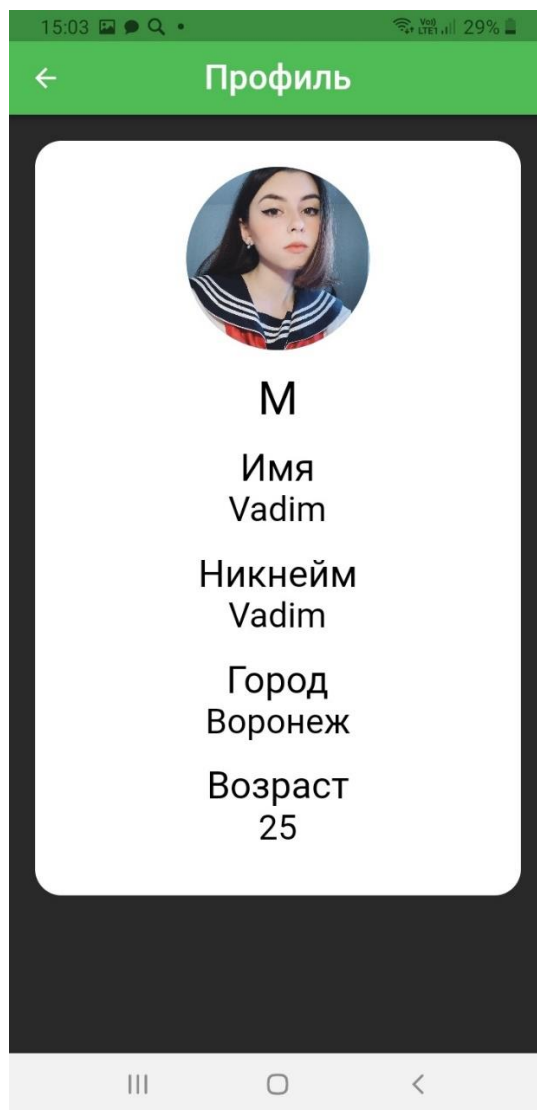


Рисунок 20 - Экран профиля другого пользователя

Экран собственного профиля пользователя также содержит краткую информацию о нем (рисунок 21). Дополнениями являются функция редактирования своего профиля и возможность выхода из него.

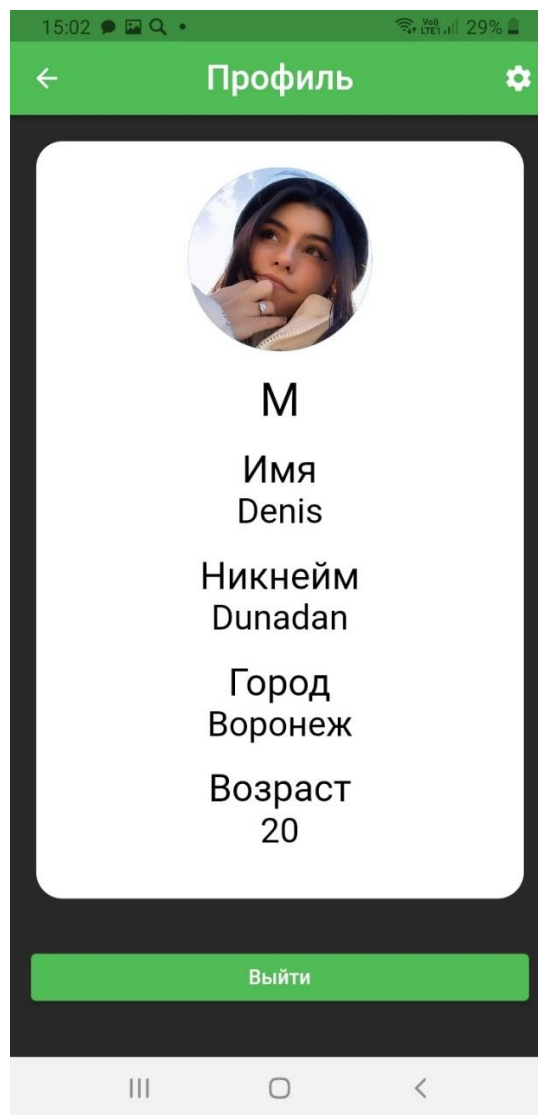


Рисунок 21 - Экран собственного профиля

Экран редактирования профиля представляет возможность пользователю редактировать информацию о себе (рисунок 22).

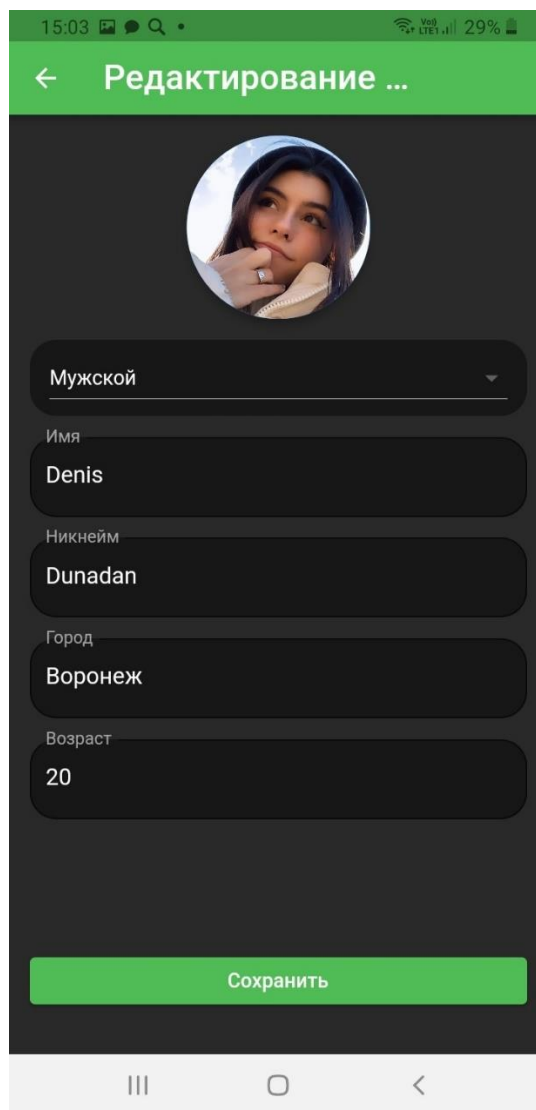


Рисунок 22 - Экран редактирования профиля

Экран редактирования аватара предоставляет пользователю список возможных фотографий для использования в качестве аватара профиля (рисунок 23).

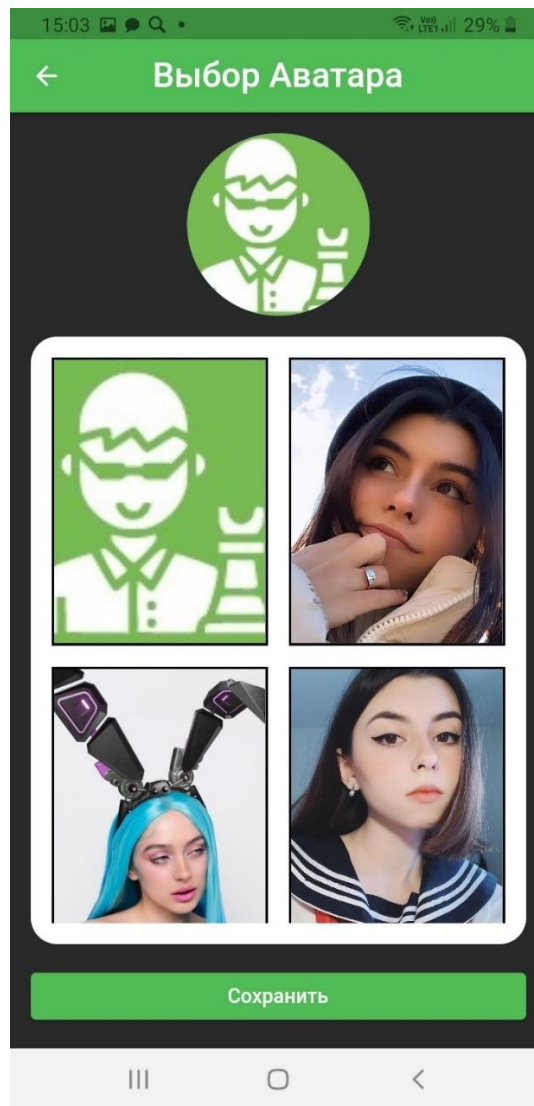


Рисунок 23 - Экран редактирования аватара

Одним из главных экранов приложения является экран моих мероприятий (рисунок 24). На нем отображается список мероприятий, в которых пользователь принимает участие или является создателем. На экране также присутствует кнопка создания нового мероприятия.



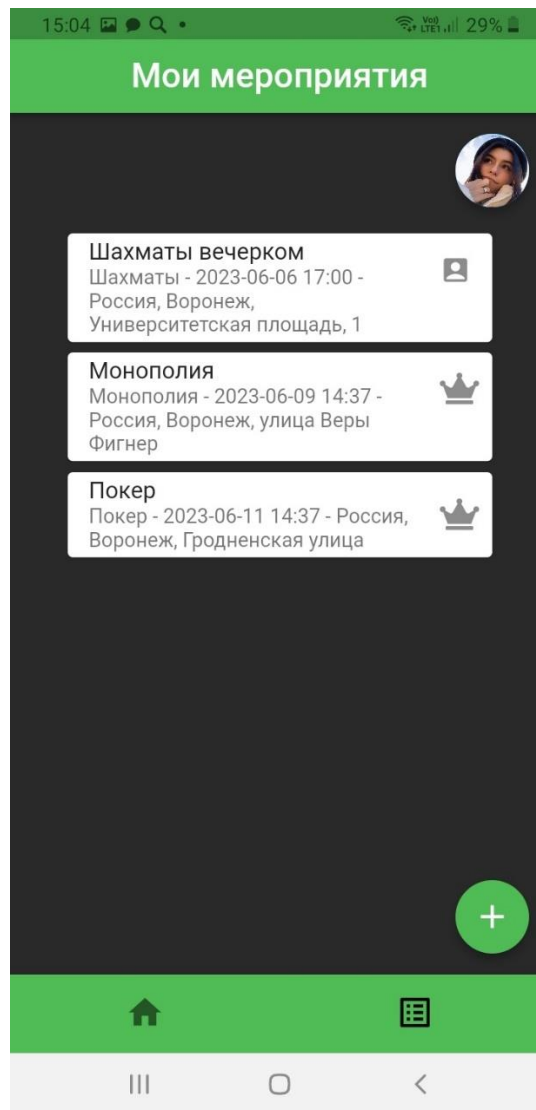


Рисунок 24 - Экран моих мероприятий

Экран создания мероприятия содержит необходимые для заполнения поля, помеченные звездочкой, а также необязательные поля, которые пользователь, являющийся создателем, впоследствии сможет заполнить (рисунок 25).

Рисунок 25 - Экран создания мероприятия

Экран просмотра мероприятия пользователем, не являющимся участником данного мероприятия, содержит краткую информацию о нем (рисунок 26).

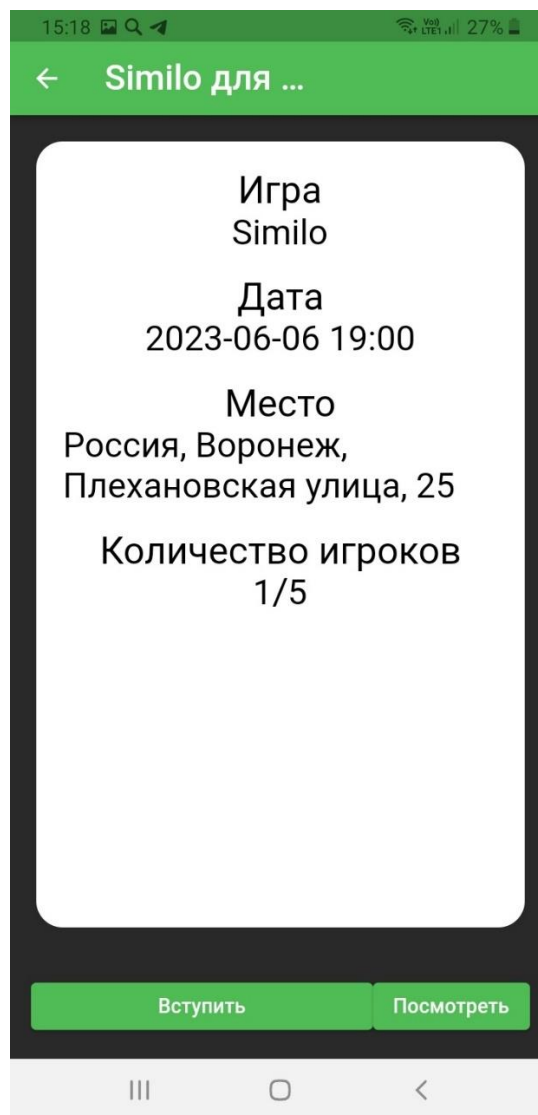


Рисунок 26 - Экран просмотра мероприятия

Экран мероприятия, которое просматривает пользователь, являющийся его участником, содержит полную информацию о нем (рисунок 27). Также имеется возможность покинуть данное мероприятие или перейти в его чат.

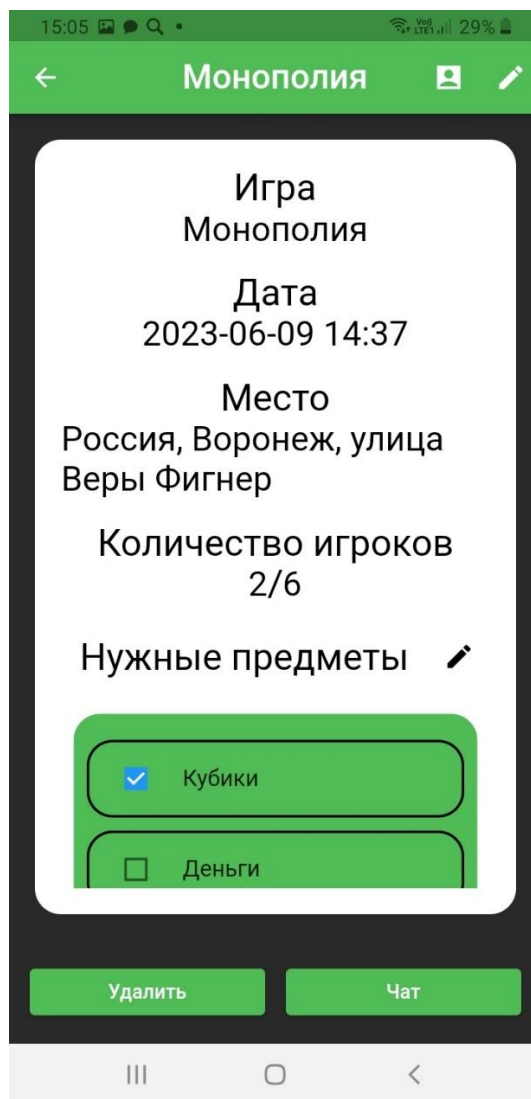


Рисунок 27 - Экран мероприятия

Экран просмотра мероприятия пользователем, являющимся создателем, также содержит полную информацию о нем (рисунок 28). Дополнением являются возможность удаления данного мероприятия, редактирование списка требуемых предметов и самого мероприятия.

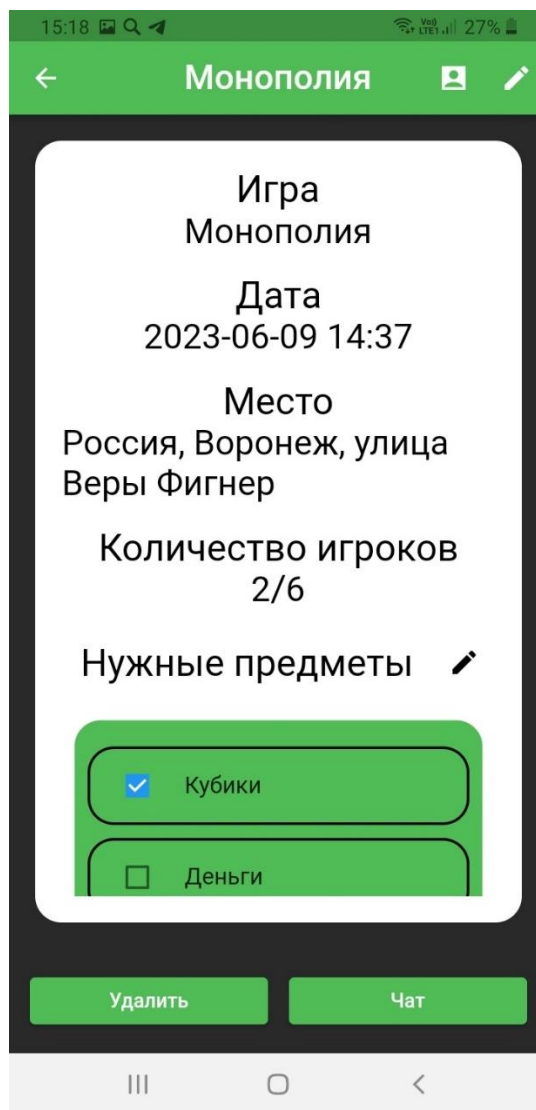


Рисунок 28 - Экран мероприятия (создатель)

Экран редактирования мероприятия отображает создателю данного мероприятия поля, доступные для изменения (рисунок 29).

Рисунок 29 - Экран редактирования мероприятия

Экран редактирования требуемых предметов позволяет создателю мероприятия удалить существующие предметы или добавить новые (рисунок 30).



Рисунок 30 - Экран редактирования нужных предметов

Пользователь, являющийся участником мероприятия, может посмотреть список его участников (рисунок 31). Данный экран содержит список никнеймов пользователей, принимающих участие в данном мероприятии, и его создателя, отмеченным значком короны.

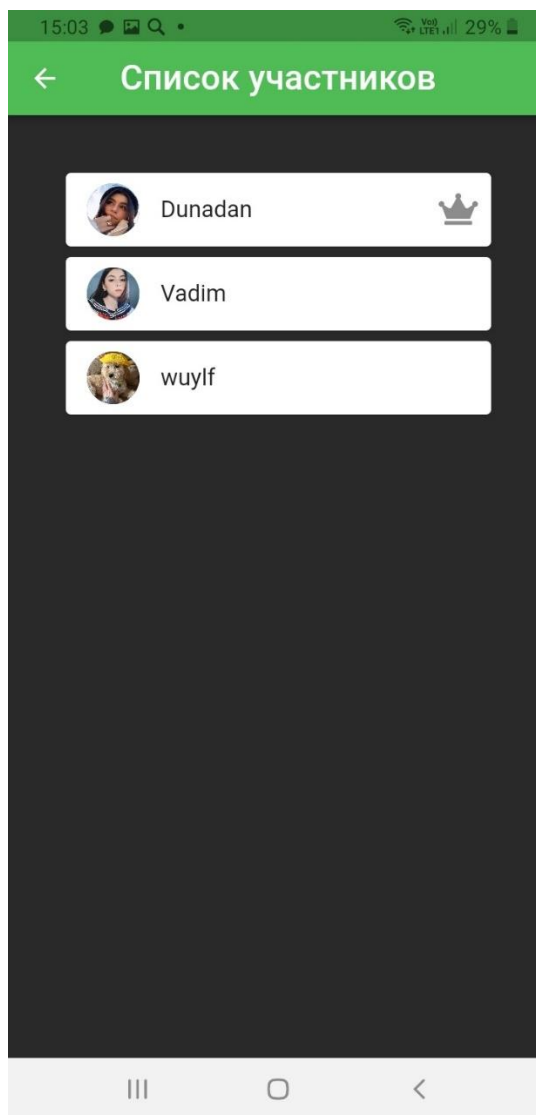


Рисунок 31 - Экран списка участников (участник)

Экран просмотра списка участников мероприятия пользователем, являющимся его создателем, также содержит список никнеймов пользователей, принимающих в нем участие, и создателя данного мероприятия. Дополнением является возможность исключения участников из мероприятия создателем (рисунок 32).



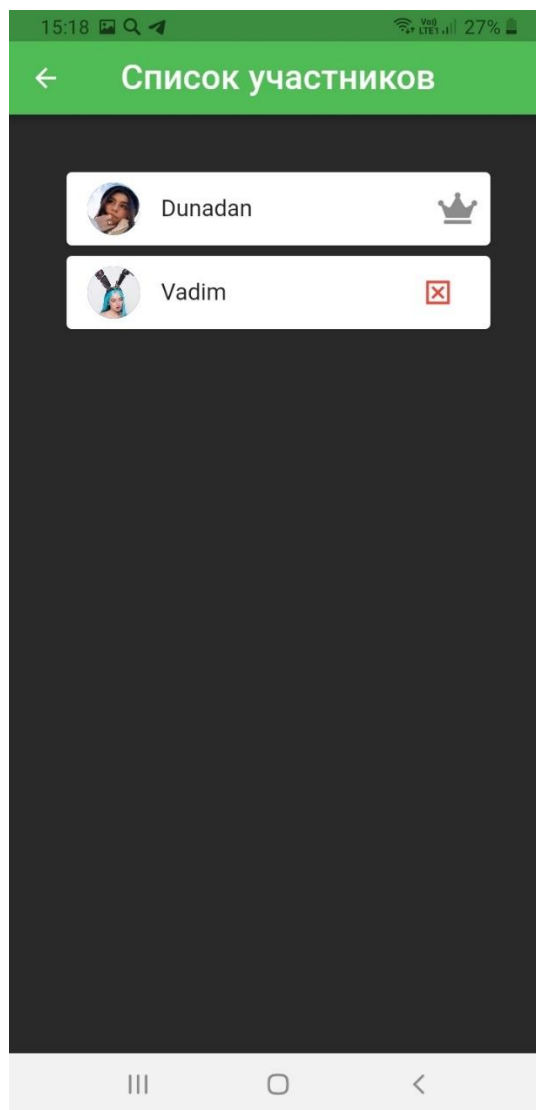


Рисунок 32 - Экран списка участников (создатель)

Пользователи имеют возможность пользоваться чатом мероприятия, в котором принимают участие. Экран чата содержит список предыдущих сообщений, отправленных участниками данного мероприятия, а также поле для написания текста сообщения и кнопку для его отправки (рисунок 33).

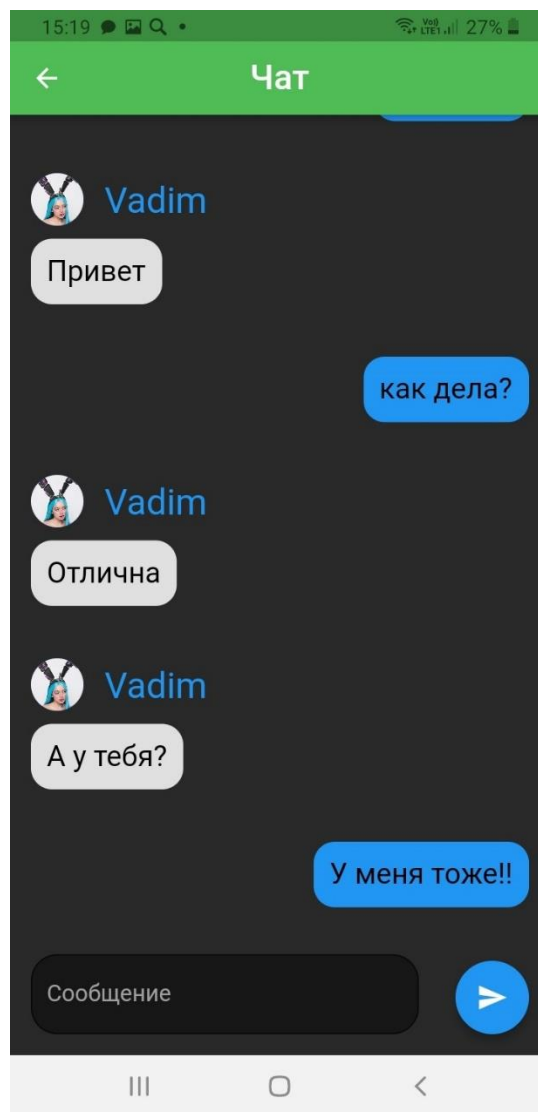


Рисунок 33 - Экран чата мероприятия

## Заключение

В результате выполнения курсовой работы было разработано мобильное приложение, под названием BoardGameMeet, для организации встреч людей, желающих сыграть в настольные игры.

Данное приложение отвечает всем требованиям, указанным в постановке задачи курсовой работы:

Приложение обладает следующими функциями:

- Реализована возможность создания мероприятий для организации встреч по настольным играм, их редактирования и удаления;
- Реализована возможность общения участников одного мероприятия в чате как во время организации мероприятия, так и после его завершения;
- Реализована возможность присоединения к любому мероприятию, если оно активно, в нем есть свободные места и пользователь подходит по возрастным требованиям данного мероприятия;
- Реализована возможность нахождения мероприятия по его названию;
- Реализована возможность указания адреса проведения мероприятия;
- Реализована возможность отмечания требуемых предметов, которые нужно взять с собой на проведение мероприятия;
- Реализована возможность просматривания активных мероприятий как для авторизованного пользователя, так и для неавторизованного;
- Реализована возможность просмотра пользователем профилей других пользователей и своего личного профиля, а также возможность редактирования своего.

Серверная часть приложения реализована с помощью использования языка программирования Kotlin.

Клиентская часть приложения реализована с помощью использования языка программирования Dart и его фреймворка Flutter.

Приложение BoardGameMeet имеет следующие перспективы развития:

- Возможность отображения адреса проведения мероприятия на карте, для того чтобы пользователи могли посмотреть точное место проведения прямо в приложении, не используя альтернативные карты.
- Возможность отправления приглашений для вступления в мероприятие другим людям посредством ссылок, что упростит процесс нахождения интересующего их мероприятия и даст возможность пользователям организовывать встречи с друзьями и знакомыми.
- Возможность добавления пользователей в друзья. Это позволит в любой момент просматривать профиль интересующего вас человека. Дополнительно можно реализовать личные чаты для пользователей, являющихся друзьями, и просмотр мероприятий, в которых он принимает участие.
- Возможность добавления ссылок на социальные сети пользователя, что упростит знакомство пользователей и улучшит их коммуникацию, а также поможет получить более подробную информацию о другом пользователе.
- Возможность загрузки изображения из галереи телефона для использования в качестве аватара пользователя.
- Возможность пользователей получать напоминания о предстоящих мероприятиях в день их проведения или в установленное пользователем время.

### **Список используемых источников**

1. Moskala M., Wojda I. Android Development with Kotlin. – Packt Publishing Ltd, 2017.

2. Späth P. Pro Android with Kotlin: Developing Modern Mobile Apps. – Apress, 2018.
3. Karanpuria R., Roy A. S. Kotlin Programming Cookbook: Explore more than 100 recipes that show how to build robust mobile and web applications with Kotlin, Spring Boot, and Android. – Packt Publishing Ltd, 2018.
4. Cosmina I., Cosmina I. Spring and Kotlin //Pivotal Certified Professional Core Spring 5 Developer Exam: A Study Guide Using Spring Framework 5. – 2020. – C. 819-853.
5. Buckett C. Dart in action. – Simon and Schuster, 2013.
6. Biessek A. Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2. – Packt Publishing Ltd, 2019.
7. Google Flutter Mobile Development Quick Start Guide by Prajyot Mainkar Salvatore Giordano – 2019
8. Nickoloff J., Kuenzli S. Docker in action. – Simon and Schuster, 2019.