МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ЭВМ
_____ Д. В. Топольский
«___»_____ 2022 г.

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ОРГАНИЗАЦИИ МЕРОПРИЯТИЙ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУРГУ-090301.2022.206 ПЗ ВКР

Руководитель работы,
к.пед.н., доцент каф. ЭВМ
_____ Ю. Г. Плаксина
«___»_____ 2022 г.

Автор работы,
студент группы КЭ-405
_____ А. С. Бредихин
«___»_____ 2022 г.

Нормоконтролёр,
к.пед.н., доцент каф. ЭВМ
_____ М. А. Алтухова
«___»_____ 2022 г.

Челябинск-2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ
_____ Д. В. Топольский
«___»_____ 2022 г.

ЗАДАНИЕ

на выпускную квалификационную работу бакалавра

студенту группы КЭ-405
Бредихину Александру Сергеевичу,
обучающемуся по направлению
09.03.01 «Информатика и вычислительная техника»

1) Тема работы: «Разработка мобильного приложения для организации мероприятий» утверждена приказом по университету от «12» декабря 2021 г. № 308/141.

2) Срок сдачи студентом законченной работы: «1» июня 2022 г.

3) Исходные данные к работе:

– разработка приложения должна осуществляться для ОС Android;

– разрабатываемый продукт должен быть написан на языке Kotlin;

– возможность регистрироваться в приложение по номеру телефона;

– возможность авторизоваться как пользователь;

– возможность осуществлять поиск мероприятий по фильтрам и картам;

– возможность создавать мероприятие;

– возможность общения между пользователями внутри приложения.

4) Перечень подлежащих разработке вопросов:

– анализ предметной области на наличие аналогичных технических решений и готовых продуктов;

– анализ и выбор программных средств для реализации проекта;

– определение требований и основного функционала проекта;

– реализация проекта;

– провести тесты работоспособности и соответствия требованиям.

5) Дата выдачи задания: «1» декабря 2022 г.

Руководитель работы _____ / *Ю. Г. Плаксина* /

Студент _____ / *А. С. Бредихин* /

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	07.02.2022	
Разработка модели, проектирование:	07.03.2022	
– анализ предметной области на наличие аналогичных приложений;	18.02.2022	
– определение требований к разрабатываемому приложению.	04.03.2022	
– выбор среды разработки; – выбор языка программирования.	04.04.2022	
– проектирование пользовательского интерфейса; – написание бек-энд составляющей приложения.	29.03.2022	
Тестирование, отладка	10.05.2022	
Компоновка текста работы и сдача на нормоконтроль	16.05.2022	
Подготовка презентации и доклада	24.05.2022	

Руководитель работы _____/ *Ю. Г. Плаксина* /

Студент _____/ *А. С. Бредихин* /

АННОТАЦИЯ

А. С. Бредихин. Разработка мобильного приложения для организации мероприятий. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭКН; 2022, 69 с., 34 ил., библиогр. список – 17 наим.

В рамках выпускной квалификационной работы выполнена разработка мобильного приложения для организации мероприятий.

В данной работе был проведен сравнительный анализ существующих проектов для организации мероприятий под операционной системы Android. Проведено сравнение и выбор основных технологических решений с целью выбора подходящей среды для разработки. Проанализированы наиболее подходящие среды разработки, а также подходящий язык программирования. Проведен анализ и сравнение баз данных, подходящих для решения поставленных задач в ходе разработки. Разработан пользовательский интерфейс и произведено проектирование, разработка и тестирование готового продукта.

Результатом выполненной работы является полноценно функционирующее мобильное приложение.

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

ОС – операционная система

ПК – персональный компьютер

БД – база данных

СУБД – система управления базами данных

ID (от англ. identifier) **–** уникальный идентификатор

iOS (от англ. iPhone Operating System) **–** операционная система для продуктов компании Apple

IDE (от англ. Integrated Development Environment) **–** система программных средств, используемая программистами для разработки программного обеспечения

API (от англ. Application Programming Interface) **–** специальный протокол для взаимодействия, с помощью которого одна компьютерная программа может взаимодействовать с другой программой

Use-case диаграмма – диаграмма вариантов использования, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне

ВВЕДЕНИЕ

В современном мире люди заняты все также, как двадцать или сорок лет назад, но досуг людей кардинально изменился. Если раньше люди проводили досуг посещая театры и оперы, то сегодня они могут выйти из дома и вблизи них с высоким шансом будет развлекательные общественные места: парки, развлекательные центры, антикафе и прочие.

Но возникает проблема, что не всегда ваши друзья или знакомые могут посетить эти места и составить компанию на мероприятие. По причинам сложности состыковки графиков рабочего времени или же отсутствие человека в городе.

В связи с тем появляется потребность в приложениях, позволяющих быстро и без труда найти компанию для проведения досуга. Благодаря этому можно расширить свой круг общения, заиметь полезные знакомства, найти людей со схожими интересами или снискать новых друзей.

Поэтому для удобного поиска компаньонов и создания своих собственных мероприятий необходим сервис, специализирующийся конкретно на этой задаче. Так как поиск сервисов и программ с подобным функционалом вызвал затруднения ввиду наличия только агрегатора или приложений неподходящий по функционалу, то было решено создать собственное приложение. Оно должно быть представлено в виде мобильного приложения, на данный момент подавляющее большинство людей пользуются смартфонами в связи с этим поднимается выбор более популярной платформы. По статистике сайта statcount.com самой популярной мобильной операционной системой является операционная система (ОС) Android [1].

Целью выпускной квалификационной работы является разработка мобильного приложения для устройств с ОС Android для организации и поиска мероприятий.

Для достижения поставленной цели необходимо решить следующие задачи:

1) анализ существующих приложений, обладающих схожим функционалом;

2) определение функционала приложения на основе анализа существующих систем;

3) изучение и выбор средств для осуществления разработки мобильного приложения для ОС Android;

4) проектирование разработки приложения;

5) разработка приложения;

6) тестирование разработанного приложения.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Для того, чтобы провести анализ существующих аналогов, необходимо выделить критерии на основе функционала приложений, по которым будут сравниваться уже существующие проекты. Выделим функции, которые должны присутствовать в таких приложениях:

1) возможность создавать мероприятие. Данная функция позволяет пользователям создавать мероприятие, к которому могут присоединиться другие пользователи;

2) возможность осуществлять поиск мероприятий, с помощью ленты. Функция приложения, с помощью которой пользователь может найти интересное для себя событие, прокручивая ленту с уже созданными мероприятиями;

3) возможность связи между участником и организатором мероприятия. Функция, с помощью которой пользователи могут общаться внутри приложения, чтобы узнать подробнее о предстоящем мероприятии;

4) возможность устанавливать фильтры для поиска: по городу; по дате; по времени; по количеству требуемых людей. Функция, необходимая для сужения поиска под предпочтения пользователя;

5) возможность производить поиск мероприятий по карте. Данная функция позволяет пользователю выбрать предстоящие мероприятия, проходящее в определенном месте, на карте.

Сравним существующие приложения по тем критериям, которые основаны на функциональных требованиях, выделенные ранее. Приложения будем отбирать в магазине мобильных приложений Google Play, в котором представлены приложения для ОС Android. Проанализируем функции представленные в таких приложения.

1.1 Обзор существующих аналогов

В магазине для мобильных телефонах Google play в основном представлены как агрегаторы, так и приложения для каких-то конкретных мероприятий, будь то концерт или конференция. Проектов, которые предоставляли бы возможность одновременно создавать и организовывать поиск нашлось не так много. В данном обзоре были представлены популярные приложения, которые наиболее близки на разрабатываемый продукт и соответствуют предметной области работы.

Приложения, которые отобраны для сравнительного обзора:

1) Event.Rocks;

2) Eventee;

3) Meetup;

4) Eventzilla;

5) InVitor.

В **Event.Rocks** [2] отсутствует лента мероприятий, вместо этого пользователю предлагается указать заранее известный ID мероприятия в поисковике на экране. Следовательно, поиск по каким-либо фильтрам тоже отсутствует. Однако, данное приложение является хорошим проектом для проведения больших конференций, разделенных на несколько этапов, так как включает в себя расписание таких этапов с указанием времени их проведения.

Преимущества:

– расписание этапов, проводимого мероприятия;

– наличие чата внутри приложения;

– наличие карты местности, здания, где проходит мероприятие;

– бесплатное распространение в магазине для мобильных устройств.

Недостатки:

– необходимость знать ID мероприятия, чтобы присоединиться к нему;

– отсутствие ленты мероприятий;

– отсутствие поиска мероприятий по карте;

– отсутствие поиска по каким-либо фильтрам;

– отсутствие возможности создать свое мероприятие.

Проект **Eventee** [3] разработан для проведения больших конференций, с участием специалистов. Данное приложение оснащено прокручивающийся лентой мероприятий, где есть возможность найти что-то интересное. Также как и Event.Rocks, присутствует расписание каждого из этапов, проводимых на конкретной конференции. Присутствует возможность оставить отзыв о прошедшем мероприятии, в котором пользователь принимал участие. Однако в приложение напрочь отсутствует возможность создать свое событие, для этого пользователю нужно написать разработчикам данного приложения, чтобы они добавили их мероприятие в ленту.

Преимущества:

– бесплатное распространение в магазине для мобильных устройств;

– наличие чата внутри приложения;

– возможность оставить отзыв, о прошедшем мероприятии;

– наличие расписаний этапов мероприятия.

Недостатки:

– отсутствие поиска мероприятий по карте;

– отсутствие поиска по каким-либо фильтрам;

– отсутствие возможности создать свое мероприятие.

Приложение **Meetpup** [4] оснащено и поиском, и возможностью создать свое мероприятие. Присутствует сортировка по дате проведения и популярности. Оснащено категориями мероприятий, при выборе которых, пользователю будут показаны события относящихся только к заданной категории. Присутствует чат, однако выполнен по принципу писем через почтовые сервисы, пользователь вписывает логин другого пользователя, после чего прописывает текст сообщения. Данное решение не удобно, так как ошибившись в одной букве логина, письмо получит другой пользователь.

Преимущества:

– возможность создать свое мероприятие;

– наличие чата внутри приложения;

– бесплатное распространение в магазине для мобильных устройств;

– наличие расписаний этапов мероприятия.

Недостатки:

– отсутствие поиска мероприятий по карте;

– неудобное исполнение чата;

– отсутствие поиска по фильтрам.

Приложение **Eventzilla** [5] разработано сразу в двух вариантах – для организации мероприятий и для их поиска. Приложение для поиска предоставляет возможность искать мероприятии по средством прокручивания ленты. Присутствует расписание по отдельным этапам мероприятия. Приложение для организации позволяет создать событие с возможностью подробного описания предстоящей программы. Однако использование данного сервиса не удобно тем, что для использования полного функционала проекта, требуется скачивать два отдельных приложения на мобильное устройство.

Преимущества:

– возможность создать свое мероприятие;

– бесплатное распространение в магазине для мобильных устройств;

– наличие статистики по мероприятию;

– наличие расписаний этапов мероприятия.

Недостатки:

– отсутствие поиска мероприятий по карте;

– отсутствие поиска по фильтрам;

– отсутствие чата.

Проект **InVitor** [6] наиболее походящий по функционалу на разрабатываемое приложение. Присутствует карта, по которой можно осуществлять поиск уже созданных событий. Также присутствует ленты мероприятий, в которой указаны категории для удобной сортировки данных событий. Удобный и привычный, для мессенджеров, чат. Присутствует возможность создания своего личного мероприятия, однако создание

происходит по уже заготовленным категориям, добавить или изменить которые нельзя. Однако наиболее значимым минусом приложения является невозможность создать мероприятие за пределами Москвы. При выборе другого города, приложение автоматически выставляют Москву.

Преимущества:

– наличие возможности создать свое мероприятие;

– наличие поиска по картам;

– наличие удобного чата;

– наличие категорий мероприятий;

– наличие ленты мероприятий;

– бесплатное распространение в магазине для мобильных устройств.

Недостатки:

– отсутствие поиска по фильтрам;

– отсутствие возможности добавить категорию, кроме имеющихся;

– отсутствие создания мероприятий за пределами Москвы.

Выделим наиболее важные функции приложений, по которым будет происходить сравнение:

1) возможность создания мероприятий;

2) наличие ленты мероприятий;

3) наличие чата;

4) поиск по фильтрам:

– по городу;

– по дате;

– по времени;

– по количеству требуемых участников;

5) наличие сортировки;

6) наличие поиска по карте.

Результаты сравнения описанных приложений по выделенным критериям представлены в таблице 1.

Таблица 1 – Результаты сравнения

		Event.Rocks	**Eventee**	**Meetup**	**Eventzilla**	**InVitor**
Возможность создания мероприятий		—	—	+	+/—	+
Наличие ленты мероприятий		—	+	+	+	+
Наличие чата		+	+	+/—	—	+
Поиск по фильтрам	По городу	—	—	—	—	—
	По дате	—	—	—	—	—
	По времени	—	—	—	—	—
	По количеству участников	—	—	—	—	—
Наличие сортировки		—	+	—	+	+
Наличие поиска по карте		—	—	—	—	+

По результатам данной таблицы следует, что разработка приложения с перечисленным выше функционалом является актуальным.

1.2 Анализ основных технологических решений

Каждый день в мире планируется много разных мероприятий. Многие из них являются массовыми, например, игра в футбол или поход в горы, а в других требуется не более 3 участников, например игра в настольные игры или поездка на машине загород.

Сервисы, которые позволяют организовать свое мероприятие, можно разделить на две категории:

– десктопные приложения;

– мобильные приложения.

Рассмотрим подробнее каждую из перечисленных категорий.

1.2.1 Десктопные приложения

Данные приложения предназначены для использования на персональном компьютере (ПК). Удобство заключается в том, что такие приложения не требуют постоянного подключения к интернету и находятся на вашем компьютере, а не на сайте. Такие приложения автоматически синхронизируют весь список мероприятий, как только персональный компьютер подключается к глобальной сети, поэтому при отсутствии доступа к интернету пользователю не составит труда получить загруженную информацию о предстоящем событии.

Однако стоит отметить, что такая независимость также может вызвать дискомфорт - в случае сбоя компьютера или вдали от рабочего места пользователь не сможет воспользоваться данным приложением.

Большим преимуществом данного приложения является то, что основной функционал программы доступен бесплатно. Существует также Pro-версия для пользователей, желающих больший функционал недоступный в бесплатных версиях.

Достоинства:

– аппаратные ресурсы работы, предоставляемые компьютером, выше чем у мобильных устройств;

– возможность реализации более масштабных приложение под данную платформу;

– возможность повысить произвольность по средством заменой аппаратных частей устройства.

Недостатки:

– мобильность персональных компьютеров уступает мобильным телефонам;

– зависимость в постоянном электропотреблении;

– необходимость в дополнительной периферии: клавиатура; микрофон; компьютерная мышь;

1.2.2 Мобильные приложения

Мобильные устройства имеют ряд преимуществ перед десктопными версиями приложений: мобильность; постоянный доступ к требуемому приложению (если данное устройство не разрядится); удобство коммуникации; не требуется дополнительного оборудования для использования аудио или видео звонков.

Достоинства:

– мобильность мобильных устройств, что позволяет обеспечить постоянный доступ к приложению;

– дешевизна приобретения таких гаджетов, в сравнении с ПК;

– наличие встроенных динамика и микрофона, а также мобильной клавиатуры;

– доступ к мобильной сети интернет.

Недостатки:

– ограниченная производительность, без возможности модификации (кроме памяти устройства).

Исходя из перечисленных достоинств, разрабатываемое приложение должно разработаться на мобильные устройства. Также иметь доступ к сети интернет, чтобы пользователь имел возможность отслеживать интересные для него мероприятия, а также получать актуальную информацию по предстоящему мероприятию.

1.3 Выбор операционной системы

Выделим существующие на данный момент операционные системы:

– Apple iOS;

– Android;

– Symbian;

– Blackberry OS.

На сегодняшний день по статистике сайта statcounter.com [1] наиболее востребованным являются ОС Andoid и iOS.

1.3.1 Операционная система Apple iOS

Система iOS от Apple является очень популярной на сегодняшний день.

IOS - операционная система, которая разработана для различных устройств от компании Apple. Данная ОС впервые появилась в 2007 году и была разработана для iPhone и iPod, а с 2014 года система работает на iPad. Главная особенность, которая отличает iOS от Android, заключается в том, что она не поддерживается на устройствах других производителей.

Следует отметить, что данная система установлена только на устройства компании Apple. Однако эта система и устройства, которые ее используют, очень популярны и востребованы, и, согласно некоторым исследованиям, операционная система iOS является самой популярной операционной системой на сегодняшний день [7].

Достоинства:

– автоматическая синхронизация с периферийными устройствами Apple, как в ручном, так и в автоматическом режиме;

– поддержка обновлений устройств с внесением доработок и улучшений в работу гаджетов;

– собственный магазин приложений Apple Store с наличием огромного количества программ;

– закрытость операционной системы. Закрытая система означает, что в неё вносят доработки только специалисты Apple, не позволяя обычным пользователем делать изменения в системе самостоятельно.

Недостатки:

– сложности при передачи данных с устройства на ОС iOS на операционную систему смартфона или ноутбука;

– невозможность расширить память. В гаджетах от Apple слот для карты памяти отсутствует, а цена самого аппарата зависит от объёма встроенной памяти;

– необходимость оплачивать учетную запись разработчика в размере 99 долларов США ежегодно.

1.3.2 Операционная система Android

Популярность системы Android обусловлена тем, что ее используют большинство устройств на мировом рынке. Кроме того, эти устройства являются более бюджетными по сравнению с продуктами от Apple на ОС iOS.

Система Android предоставляет разработчикам огромное количество API-интерфейсов, что значительно помогает в разработке мобильных приложений для устройств, использующих эту операционную систему.

Достоинства:

– большой выбор телефонов на ОС Android позволяет подбирать телефоны с разными возможностями, например поддержка 2 сим-карт или слотами для карт памяти;

– возможность расширять встроенную память по средством карты памяти;

– собственный магазин приложений Play Маркет с наличием всевозможных программ;

– Android является открытой операционной системой, любой пользователь может доработать управление устройством под себя, внося изменения в программную часть;

Недостатки:

– открытость системы является и минусом, так как приходится обращать внимание какие данные и из каких источников скачивать во избежание последующей некорректной работы аппарата или появления вируса;

– необходимость однократно оплатить учетную запись разработчика в размере 25 долларов США.

Операционная система Android является основным конкурентом iOS и, согласно некоторым исследованиям, является самой популярной системой. Согласно данным Mail.ru Group, топ мобильных платформ устройств, с которых пользователи посещают Интернет, выглядит таким образом:

– Android (55% пользователей);

– iOS (26% пользователей);

– другие операционные системы (18% пользователей) [8].

Операционная система Android является наиболее оптимальным решением для разрабатываемого приложения.

1.4 Выбор среды разработки

Существуют различные среды разработки, позволяющие разрабатывать приложения под устройства, которые используют операционную систему Android. Однако, самыми популярными являются следующие IDE:

– Microsoft Xamarin;

– Eclipse;

– Android Studio.

1.4.1 Среда разработки Microsoft Xamarin

Среда разработки Microsoft Xamarin - это платформа с открытым исходным кодом, позволяющая создавать приложения для устройств с лидирующими на рынке операционными системами – iOS, Android. Microsoft Xamarin является бесплатной и доступна в составе Microsoft Visual Studio.

Разработчики, используя Xamarin, могут создавать кроссплатформенные приложения под различные платформы – iOS, Android, Windows и Mac. Разработка приложений происходит на языке C# с использованием .NET.

Данная среда позволяет создать единый пользовательский интерфейс на все платформы, используя инструмент Xamarin.Forms [9].

Достоинства:

– единый стек технологий для разработки приложений на любые платформы;

– производительность приложений близка к нативной;

– полная поддержка оборудования, то есть устраняются все проблемы совместимости оборудования, используя плагины и различные API;

– Xamarin упрощает поддержку и обновление программного обеспечения.

Недостатки:

– приложения написанные через Xamarin весят обычно больше, чем нативные, иногда даже в два раза;

– задержка в обновлении поддержки платформы, требуется некоторое время на внедрение поддержки для новой версии Android или iOS.

1.4.2 Среда разработки Eclipse

Eclipse является интегрированной средой разработки, которая позволяет подключать дополнения, используя которые можно значительно расширить функционал данной программы. IDE является кроссплатформенной – она применяется для таких операционных системам как Linux, Windows, Mac OS X и Solaris.

Данная среда разработки является бесплатной и очень популярной. Большое количество разработчиков предпочитают использовать IDE Eclipse для создания различных приложений на языках Java, C, C++, PHP, Perl, Python и других [10].

Однако, данная среда разработки не предназначена для создание именно мобильных приложений под систему Android, что является как плюсом, так и минусом.

С одной стороны, то, что IDE Eclipse позволяет очень удобно создавать разнообразные продукты.

С другой стороны, при разработке приложений под систему Android из-за гибкости Eclipse возникают проблемы, в сравнении со средой разработки, которая ориентированная именно на создание данных приложений. Одной из таких проблем является необходимость самостоятельно дополнительно устанавливать эмулятор устройства с ОС Android, который необходим для тестирования разрабатываемого приложения.

Достоинства:

– простата в установке и использовании;

– бесплатное программное обеспечение с открытым исходным кодом;

– наличие функций, утилит и автодополнения, что облегчают написание кода;

– является доступной для любых платформ, так как написана на Java;

– присутствует поддержка всевозможных языков программирования;

– имеет подключение к разнообразным базам данных.

Недостатки:

– использует больше системных ресурсов в сравнение с другими IDE;

– запуск среды разработки медленный, а временами задействует больше памяти в сравнение с другими IDE;

– необходимость дополнительно устанавливать эмулятор с ОС Android для тестирования разрабатываемых приложений;

– имеет большой размер при установке на внутреннюю память ПК.

1.4.3 Среда разработки Android Studio

Среда разработки Android Studio - это интегрированная среда разработки предназначенная для работы с ОС Android, анонсированная 16 мая 2013 года на

конференции Google I/O [11]. Этот продукт был создан компанией Google специально для разработки мобильных приложений на языке Kotlin или Java, на усмотрение разработчика, для операционной системы Android. Данная среда является кроссплатформенной и разработана на основе IntelliJ IDEA.

IDE Android Studio, разработана специально для разработки приложений под операционную систему Android, ориентирована исключительно на эту систему и поэтому довольно удобна, понятна и функциональна.

Среда разработки Android Studio имеет ряд преимуществ:

– предварительный просмотр изменений при редактировании xml;

– возможность создания шаблонных активностей;

– улучшенный интеллектуальный анализ кода;

– сборка приложений, основанная на Gradle;

– различные виды сборок и генерация нескольких .apk файлов;

– рефакторинг кода;

– статический анализатор кода (Lint), позволяющий находить проблемы производительности, несовместимости версий и другое;

– шаблоны основных макетов и компонентов Android;

– полноценный режим отладки и улучшенное добавление точек остановки.

Учитывая указанные преимущества и то, что разрабатываемое приложение предназначено для мобильных устройств с ОС Android, было решено использовать именно данную IDE.

1.5 Выбор языка программирования

В среде разработки Android Studio интегрированы такие языки программирования как:

– Java;

– Kotlin.

1.5.1 Язык программирования Java

Java является языком объектно-ориентированного программирования, разрабатываемый компанией Sun Microsystems с 1991 года и официально выпущенный 23 мая 1995 года. Программы для Java переводятся в байт-код, затем происходит обработка этого кода виртуальной машиной Java, а инструкция передается оборудованию в качестве интерпретатора [12].

При описанном способе байт-код программа выполняется полностью независимо от операционной системы и оборудования, что позволяет запускать Java-приложения на любом устройстве.

Еще одной важной особенностью является гибкость системы безопасности. Это достигается за счет того, что выполнение программы полностью контролируется виртуальной машиной. Любые действия, превышающие установленные полномочия (например, попытка несанкционированного доступа к данным или подключение к другому компьютеру, могут быть отнесены к таким действиям), приводят к немедленному прерыванию.

Основными достоинствами языка Java является:

– объектно-ориентированный подход;

– безопасность, используются классы, имеющие цифровую подпись;

– надёжность. Компилятор способен выявить ошибки на ранних стадиях, до выполнения кода;

– независимость от аппаратной части и ОС;

– динамичность и адаптируемость;

– удобные и эффективные сетевые возможности.

1.5.2 Язык программирования Kotlin

Kotlin представляет собой один из современных, статически типизированных и быстрорастущих языков программирования, созданный и

разработанный компанией JetBrains. Kotlin можно использовать для создания широкого спектра приложений. Это и приложения для таких операционных систем как Android, iOS. Кроме того, Kotlin позволяет написать кроссплатформенный код, который будет реализован на всех платформах. Это как и веб-приложения, работающие на серверном бэкэнде, так и клиентские приложения на базе браузера – фронтэнде [13].

Самым популярным направлением, в котором используется язык Kotlin, является разработка под операционную систему Android. Кроме того, настолько популярно, что компания Google объявила Kotlin одним из официальных языков для разработки под Android на конференции Google I/O в 2017 году, а инструменты, связанные с этим языком, были добавлены по умолчанию в функциональность среды разработки Android Studio.

Основное отличие от языка программирование Java - это то, что Kotlin требует меньше «шаблонного» кода, имеется ввиду, более простой для чтения и восприятия код. Данный язык также устраняет такие ошибки, как исключение нулевого указателя, а также освобождает разработчика от необходимости заканчивать каждую строку кода точкой с запятой.

Основные плюсы данного языка это:

1) чистота кода. В Kotlin нет лишних функций или избыточных модулей, код получается компактным и лаконичным. По сравнению с другими языками, код в Kotlin имеет меньшее количество строк, что снижает количество ошибок;

2) безопасность. На этапе компиляции происходит автоматическая проверка кода на null безопасность. В Kotlin предусмотрена функция Null Safety, при помощи которой можно избежать ошибок неопределённости в коде — самых распространённых и самых трудно выявляемых ошибок;

3) производительность. Благодаря схожей структуре байт-кода приложения написанные на языках Kotlin и Java выполняются с одинаковой скоростью, при условии одного и того же аппаратного обеспечения;

4) наличие корутин. Корутина - это функционал, который выполняет блоки кода асинхронно, по очереди. В нужный момент исполнение данного

блока приостанавливается с сохранением всех его свойств, чтобы запустить другой код.

Принимая во внимание перечисленные преимущества и тот факт, что язык программирования Kotlin имеет низкий порог входа, легко поддается изучению, было решено писать разрабатываемое приложение именно на данном языке.

1.6 Выбор базы данных

Для разрабатываемого приложения требуется база данных, чтобы хранить необходимую информация.

База данных представляет собой упорядоченный набор информации или данных, обычно хранящихся в электронном виде в компьютерной системе. База данных управляется системой управления базами данных (СУБД). Данные в наиболее распространенных современных типах баз данных обычно создаются в виде строк и столбцов в ряде таблиц, чтобы более эффективно обрабатывать и запрашивать данные. Затем можно легко получать доступ, проверять и редактировать данные.

СУБД действует как интерфейс между базой данных и пользователями или программами, предоставляя пользователям возможность получать, обновлять и оптимизировать информацию. СУБД также упрощает мониторинг и управление базами данных, позволяя выполнять различные административные операции, такие как мониторинг производительности, настройка и резервное копирование и восстановление [14].

Для мобильных приложений в основном используются следующие базы данных:

– SQLite;

– MySQL;

– PostgreSQL;

– Firebase.

Рассмотрим данные варианты подробнее и выберем наиболее подходящую базу данных для реализации разрабатываемого приложения.

1.6.1 База данных SQLite

SQLite – это автономная файловая открытая СУБД. Его основными характеристиками являются переносимость и высокая производительность. SQLite определяется как «бессерверная» база данных. Большинство механизмов реляционной базы данных реализуются как серверный процесс, при котором другие программы взаимодействуют с запросами. Однако в SQLite любой запрос, получивший доступ к базе данных, считывает и записывает данные непосредственно в файл базы данных. Это облегчает процесс установки SQLite, поскольку устраняет необходимость настройки сервера. SQLite - бесплатное программное обеспечение с открытым исходным кодом, использование которого не требует специальной лицензия [15].

Достоинства:

– библиотека SQLite является легковесной и полностью автономной, для нее не требуется никаких внешних зависимостей;

– для работы с базой данных не требуется настраивать и запускать серверный процесс, что упрощает интеграцию с приложением;

– вся база данных хранится в одном файле, что делает ее легко переносимой.

Недостатки:

– только один процесс может вносить изменения в базу данных в любой момент времени, что ограничивает параллелизм;

– в SQLite отсутствует управление пользователями, что делает его неподходящим для приложений, где требуется несколько пользователей с разными правами доступа;

– ядро базы данных, использующей сервер, обеспечивает более надежную защиту от ошибок, чем SQLite.

1.6.2 База данных MySQL

MySQL обеспечивает очень быстрый, многопоточный, многопользовательский и надежный сервер баз данных SQL (Structured Query Language). Программное обеспечение MySQL имеет двойную лицензию: пользователи могут выбрать использование программного обеспечения MySQL в качестве продукта с открытым исходным кодом на условиях стандартной общественной лицензии GNU или могут приобрести стандартную коммерческую лицензию у Oracle. MySQL имеет широкое распространение и поддерживает веб-сайты и приложения [16].

Достоинства:

– популярность MySQL означает, что имеется множество информации об установке и управлению базой данных, также существует множество инструментов, упрощающих работу с MySQL;

– простота в использовании. MySQL достаточно легко инсталлируется, а наличие множества плагинов и вспомогательных приложений упрощает работу с базами данных;

– безопасность. Система изначально создана таким образом, что множество встроенных функций безопасности в ней работают по умолчанию;

– скорость. Высокая производительность системы обеспечивается за счет упрощения некоторых используемых в ней стандартов.

Недостатки:

– MySQL был разработан для скорости и простоты использования, поэтому не соответствует полностью стандарту SQL, он имеет определенные функциональные ограничения;

– бесплатной версией сообщества могут пользоваться только проекты с открытым исходным кодом, а некоторые функции и плагины доступны только для проприетарных версий.

1.6.3 База данных PostgreSQL

PostgreSQL – свободно распространяемая объектно-реляционная СУБД. Это означает, что она также включает в себя такие функции, как наследование таблиц и перегрузка функций. PostgreSQL поддерживает обширный список типов данных, кроме стандартных типов данных в нем также присутствуют: uid, денежный, перечисляемый, геометрический, бинарный тип данных, сетевые адреса, битовые строки, xml, json, массивы.

Достоинства:

– PostgreSQL стремится к более строгому соблюдению стандартов SQL, чем другие СУБД. PostgreSQL поддерживает 160 из 179 функций, необходимых для полного соответствия требованиям SQL;

– PostgreSQL свободно распространяется и его использование не накладывает никаких ограничений на проект;

– поддержка пользовательских типов данных и их поведения делает данную СУБД более гибкой.

Недостатки:

– PostgresSQL используется не так широко, как MySQL, это означает, что для данной СУБД существует меньше сторонних инструментов, помогающих в управлении базой данных;

– для каждого нового клиентского соединения PostgreSQL разветвляется новый процесс, требующий примерно 10 МБ памяти, что довольно значительно для базы данных с большим количеством подключений.

1.6.4 База данных Firebase

Firebase - это облачная база данных, которая позволяет пользователям хранить и получать сохраненную информацию, а также имеет удобные средства и методы взаимодействия с ней.

Firebase хранит текстовые данные в JSON формате и предоставляет удобные методы для чтения, обновления и извлечения данных. Также, Firebase позволяет организовать регистрацию и авторизацию пользователей, хранить сессии (авторизованные пользователи), медиафайлов к которым с легкостью предоставляет доступ благодаря Cloud Storage [17].

Достоинства:

– удобная статистика, позволяющая получать данные о действиях пользователей и поддерживать обратную связь;

– высокая скорость работы приложения, так как база данных находится в облачном пространстве;

– простая масштабируемость - рост количества пользователей не потребует изменений в серверном коде;

– кроссплатформенность - приложение создано один раз и настроено для работы со всеми операционными системами;

– данная база данных обеспечивает оптимальную безопасность и доступность данных за счет осуществления регулярного резервного копирования;

– обладает большим функционалом, что в будущем немаловажно для усовершенствования разрабатываемого приложения.

Недостатки:

– в Firebase присутствует возможность использовать только нереляционный вид базы данных, что влечет за собой сложности в использовании сложных запросов;

– не все функции данного инструмента являются бесплатными.

Проанализировав достоинства и недостатки вышеперечисленных баз данных, было принято решение разрабатывать приложение с использованием Firebase. Данная база данных обладает нужными характеристиками в рамках разрабатываемого приложения.

2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

Общие требования для мобильного телефона:

– устройство на операционной системе Android;

– операционная система не ниже Android 5.0.

2.1 Функциональные требования

На этапе проектирования системы были выявлены следующие функциональные требования:

1) возможность регистрации пользователя внутри приложения;

2) возможность авторизации пользователя внутри приложения;

3) возможность заполнить профиль пользователя;

4) наличие чата;

5) возможность создать свое мероприятие;

6) наличие ленты мероприятий;

7) возможность производить поиск по фильтрам:

– по городу;

– по дате;

– по времени;

– по количеству требуемых участников.

2.2 Нефункциональные требования

На этапе проектирования были выявлены следующие нефункциональные требования:

– разработка системы на платформе Android Studio 2021.2.1;

– система должна быть написана на языке Kotlin;

– пользователь должен указать отображаемое имя и псевдоним после регистрации.

3 ПРОЕКТИРОВАНИЕ

3.1 Логическая структура

Диаграмма, которая представлена на рисунке 1, отображает какие действия может предпринять неавторизованный пользователь сразу после запуска приложения.

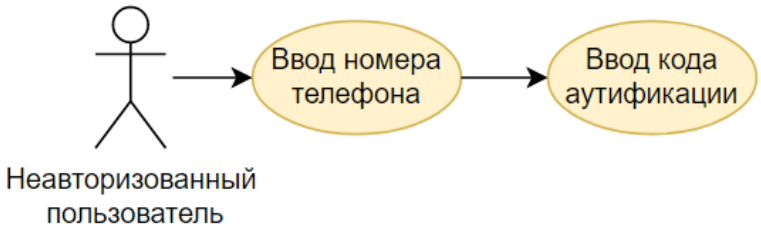

Рисунок 1 – Use-case диаграмма действий неавторизованного пользователя при входе в приложение

Далее приложение обращается к базе данных и сравнивает введенный номер телефона с теми, что уже присутствуют в базе данных. Если данный номер уже находится в базе данных, то пользователю приходит код для аунтификации, а если такого номера нет, то генерирует новый код и присылает неавторизированному пользователю, чтобы можно было войти в приложение.

При входе в приложения, пользователь может выдвинуть боковое меню, в котором отображены основные функции приложения. На рисунке 2 отображены все элементы бокового меню разрабатываемого приложения.

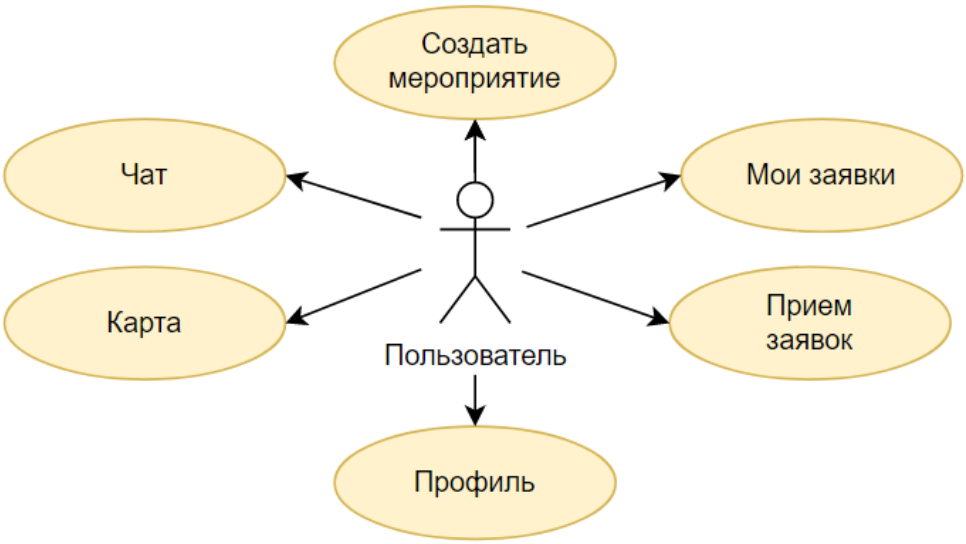

Рисунок 2 – Use-case диаграмма бокового меню

Пользователь может отредактировать данные в своем профиле. Для редактирования доступны параметры, указанные на рисунке 3.

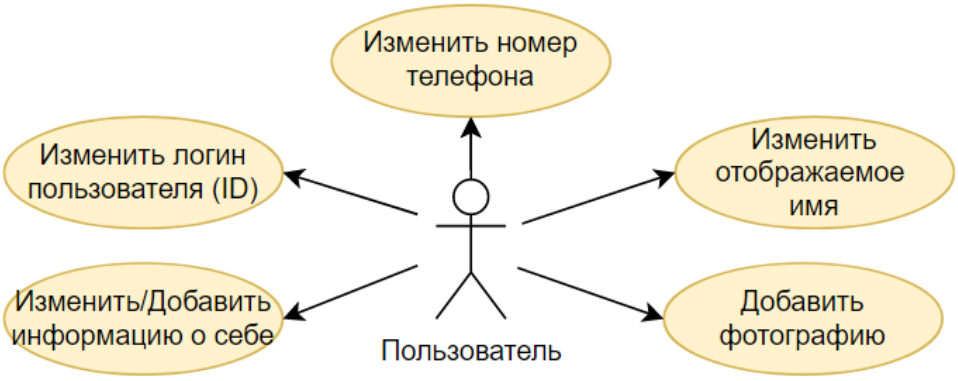

Рисунок 3 – Use-case диаграмма редактирования профиля

Главной страницей является агрегатор всех доступных предстоящих мероприятий, где пользователь может найти и выбрать интересное для него мероприятие.

Поиск осуществляется при помощи фильтров, которые устанавливает сам пользователь. Фильтрация происходит согласно use-case диаграмме, проиллюстрированной на рисунке 4.

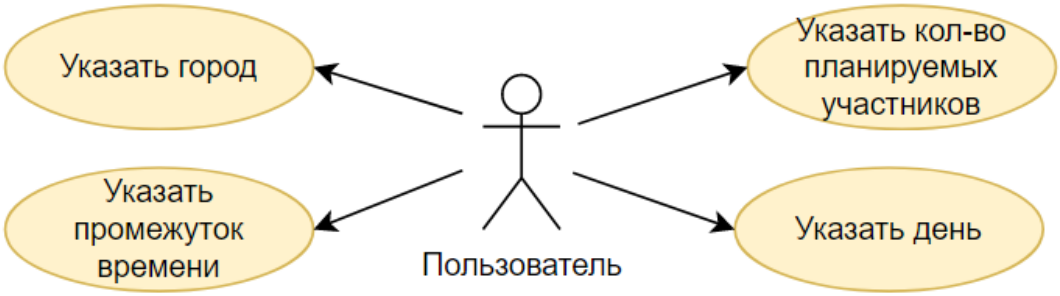

Рисунок 4 – Use-case диаграмма фильтрации мероприятий

Через боковое меню пользователь может создать свое мероприятие. Для этого необходимо указать следующие данные: краткое название создаваемого мероприятия; описание данного мероприятия; количество требуемых участников; дата и время, когда будет происходить мероприятие; указать место встречи на карте. Диаграмма создания мероприятий продемонстрирована на рисунке 5.

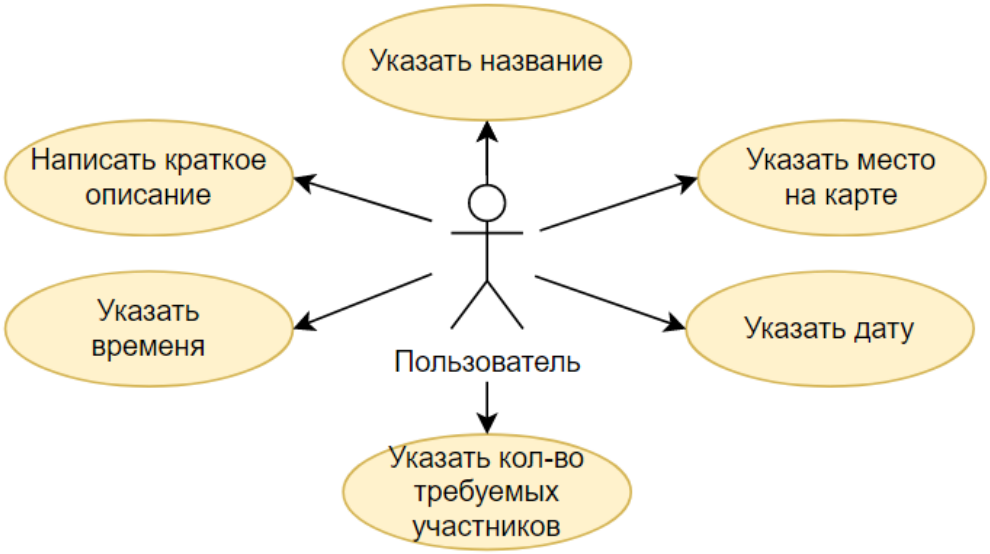

Рисунок 5 – Use-case диаграмма создания мероприятий

В пункте меню «Мои заявки» выводятся мероприятия, в которые пользователь подал заявку на участие. В том же пункте присутствуют возможности: отменить заявку или отредактировать сопроводительное сообщение, почему пользователь хочет присоединиться к данному событию, как продемонстрированно на рисунке 6.

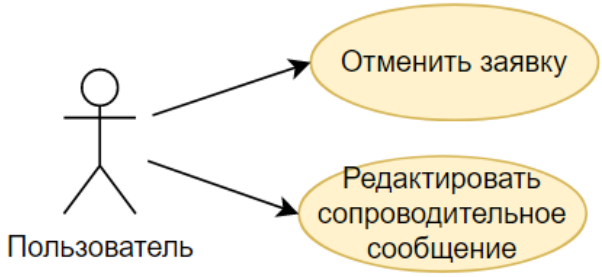

Рисунок 6 – Use-case диаграмма «Мои заявки»

В пункте меню «Прием заявок» выводятся запросы на участие в мероприятие, созданного пользователем-организатором, от других пользователей приложения. Создатель данного события имеет возможность как принять запрос на участие, так и отклонить. Имеется отдельная кнопка для связи, по средством чата, с пользователем, который отправил запрос на участие как продемонстрировано на рисунке 7.

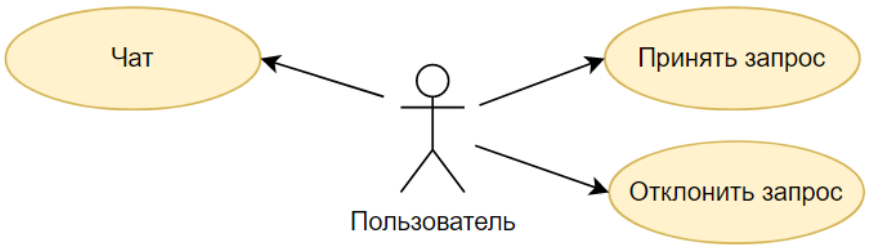

Рисунок 7 – Use-case диаграмма «Прием заявок»

3.2 Хранение данных

Для разрабатываемого приложения была выбрана база данных Firebase. Данная БД является нереляционной, а следовательно в отличие от большинства традиционных систем баз данных не используется табличная схема строк и столбцов. В этих базах данных применяется модель хранения, оптимизированная под конкретные требования типа хранимых данных. Например, данные могут храниться как простые пары "ключ - значение", документы JSON или граф, состоящий из ребер и вершин.

Рассмотрим базу данных для разрабатываемого приложения на рисунке 8.

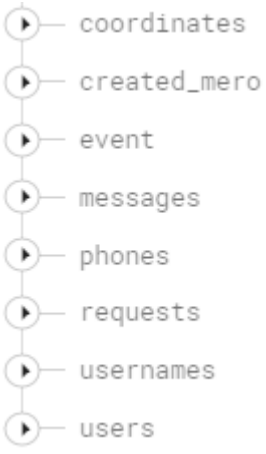

Рисунок 8 – Хранение данных в Firebase

Как можно заметить, никаких таблиц нет, данные хранятся в виде ключ-значение и находятся в выпадающих списках. Обратимся к списку «users», и рассмотрим данные изображенные на рисунке 9.

Рисунок 9 – Данные в списке «users»

Список «users» предназначен для хранения пользователей и связанной с ними информацией. Подробнее ознакомится с каждым элементом списка можно в таблице 2.

Таблица 2 – Описание данных в списке «users»

Название	Описание
bio	Описание профиля пользователя
fullname	Настоящие имя и фамилия пользователя
id	Идентификатор пользователя
phone	Номер телефона пользователя
photoUrl	Ссылка на фотографию, хранящуюся в облачном хранилище
state	Статус пользователя
username	Псевдоним пользователя

Список «usernames» хранит в себе псевдоним пользователя с указанием его идентификатора. На рисунке 10 представлены два пользователя.

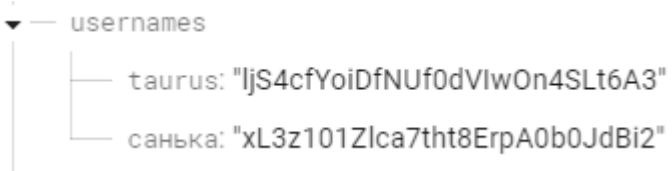

Рисунок 10 – Данные в списке «usernames»

На рисунке 11 продемонстрированы данные, хранящиеся в списке «phones» для хранения номера телефона пользователя.

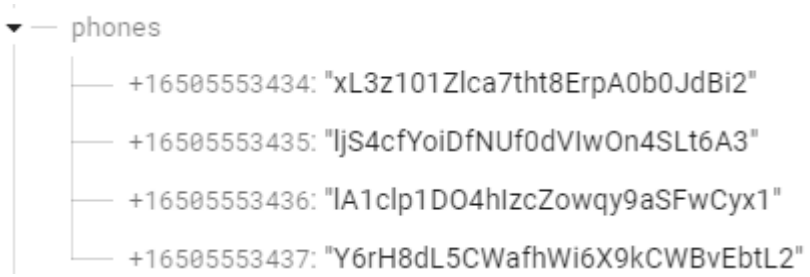

Рисунок 11 – Данные в списке «usernames»

Список «messages», изображенный на рисунке 12, хранит сообщения пользователей.

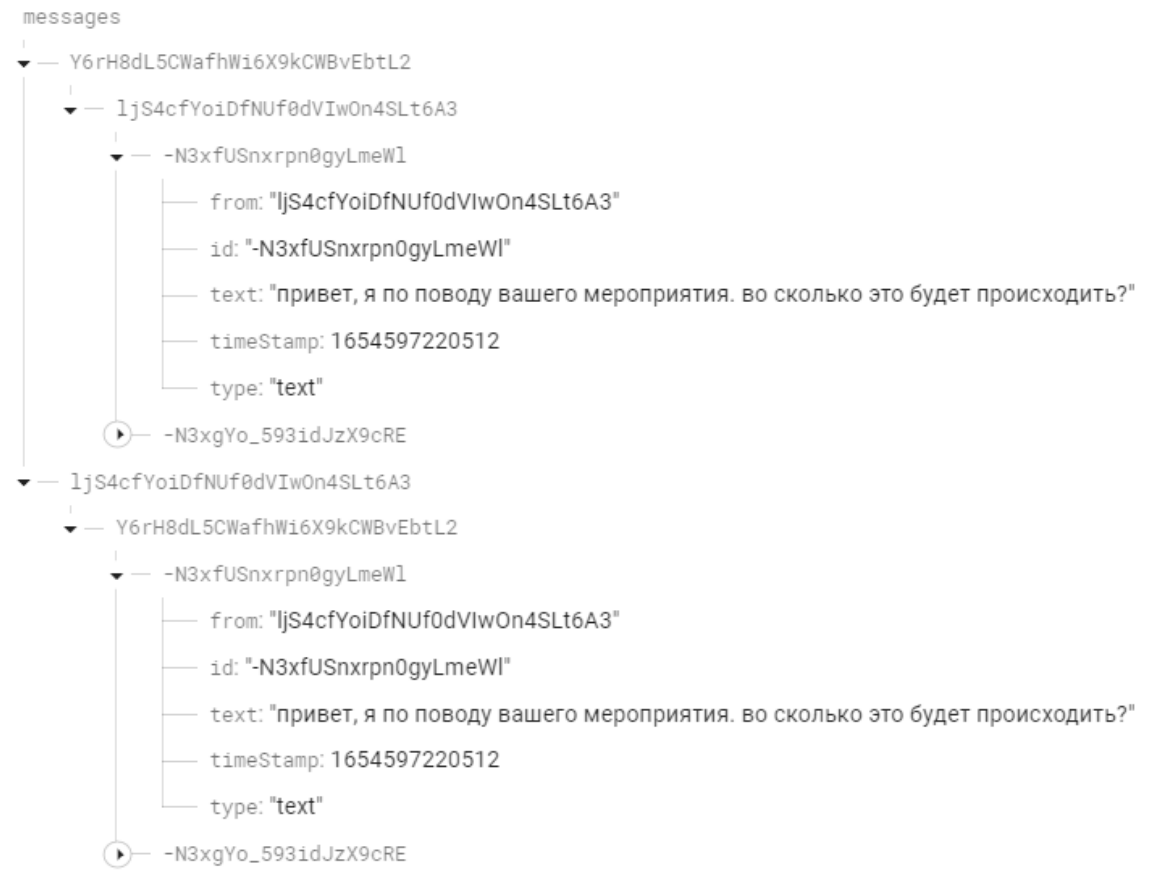

Рисунок 12 – Данные в списке «phones»

Подробное описание хранящихся данных указаны в таблице 3.

Таблица 3 – Описание данных в списке «phones»

Название	Описание
from	Идентификатор пользователя, от которого поступило сообщение
id	Идентификатор сообщения

Продолжение таблицы 3

text	Текст сообщения
timeStamp	Код времени
type	Тип данных сообщения

Список «event», рисунок 13, находятся созданные пользователями мероприятия.

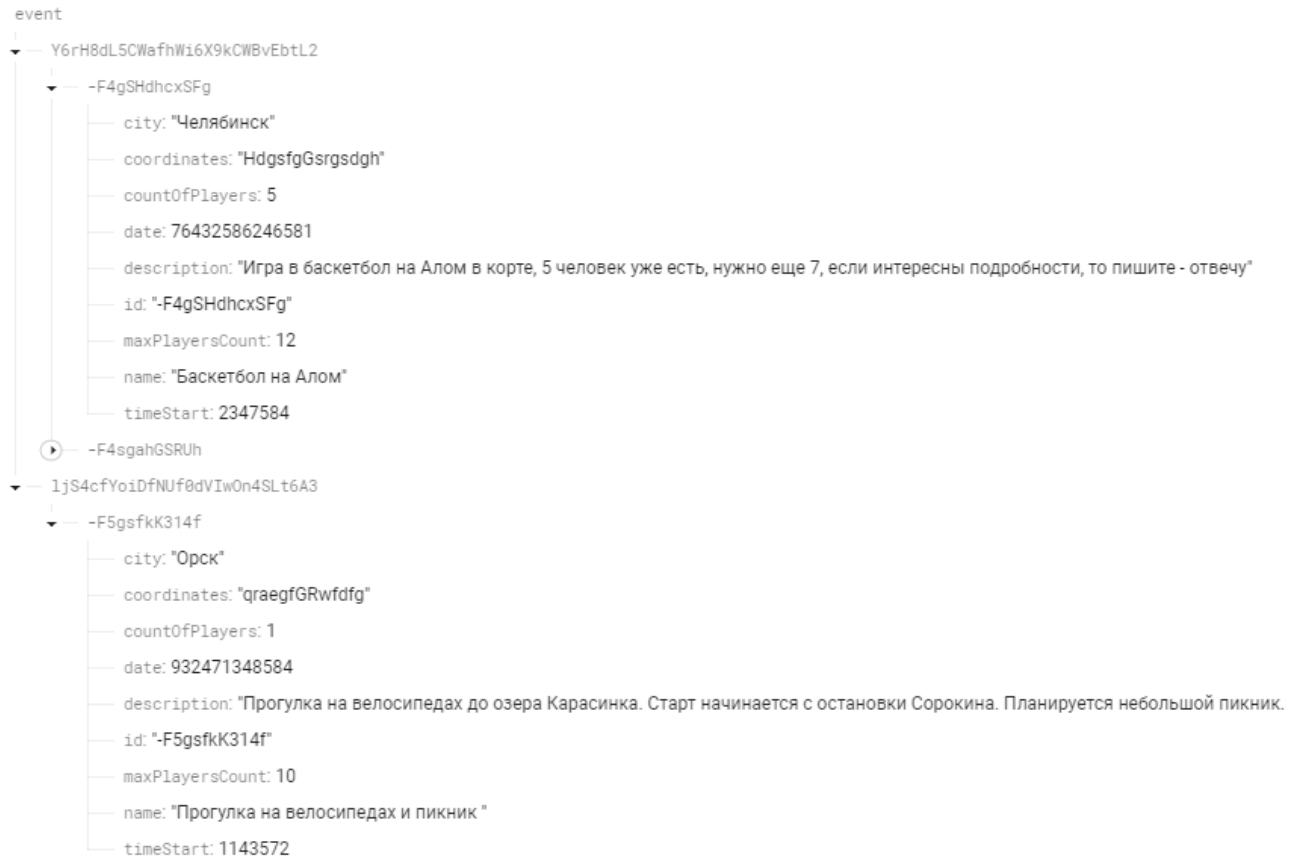

Рисунок 13 – Данные в списке «event»

Как продемонстрированно на рисунке, один пользователь может создавать сразу несколько мероприятий. В таблице 4 описаны данные находящимися в данном списке.

Таблица 4 – Описание данных в списке «users»

Название	**Описание**
city	Город, в котором будет происходить мероприятие
coordinates	Идентификатор координат места встречи

Продолжение таблицы 4

countOfPlayers	Имеющееся количество участников
date	Код даты проведения
description	Описание данного мероприятия
id	Идентификатор мероприятия
maxPlayersCount	Требуемое количество участников
name	Название мероприятия
timeStart	Код начала времени проведения

Список «coordinates» хранит в себе идентификатор координат, их долготу и широту. Развернутый список продемонстрирован на рисунке 14.

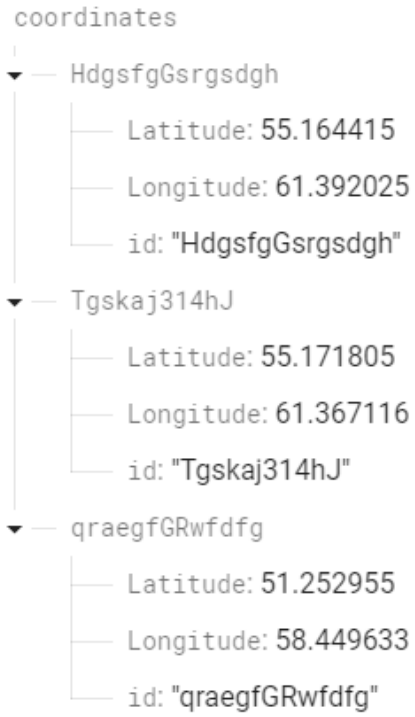

Рисунок 14 – Данные в списке «coordinates»

На рисунке 15 продемонстрирован список «created_mero», который хранит в себе данные, связывающие идентификатор пользователя и идентификатор созданного им мероприятия.

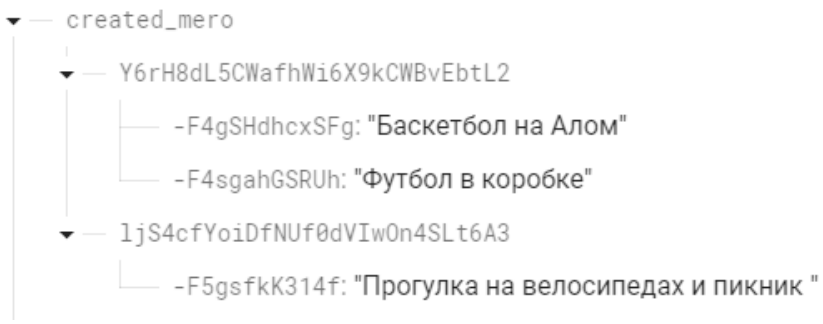

Рисунок 15 – Данные в списке «created_mero»

Список «requests» хранит в себе запросы пользователей на участие в мероприятиях других пользователей. Вложенный список «chain» хранит связь пользователя, отправившего запрос, с мероприятием, в котором планируется участие. На рисунке 16 продемонстрирован данный список.

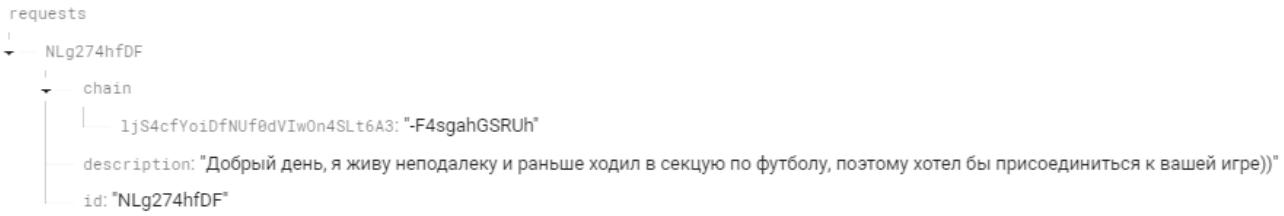

Рисунок 16 – Данные в списке «requests»

В таблице 5 находится описание хранящихся данных в списке «requests».

Таблица 5 – Описание данных в списке «users»

Название	**Описание**
description	Сопроводительное описание пользователя к запросу на участие
id	Идентификатор запроса на участие

4 РЕАЛИЗАЦИЯ

Для реализации разрабатываемого приложения был разработан пользовательский интерфейс.

При первом запуске приложения, неавторизованный пользователь попадает на форму введения номера телефона, как продемонстрировано на рисунке 17.

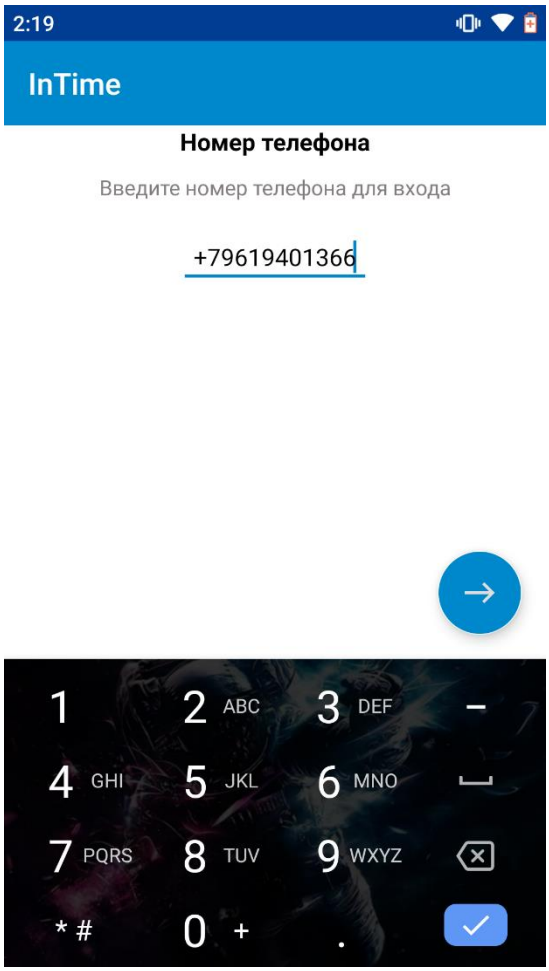

Рисунок 17 – Ввод номера телефона

После ввода номера телефона, пользователь переходит на форму для заполнения кода аунтификации, как показано на рисунке 18. Данный код высылается автоматически.

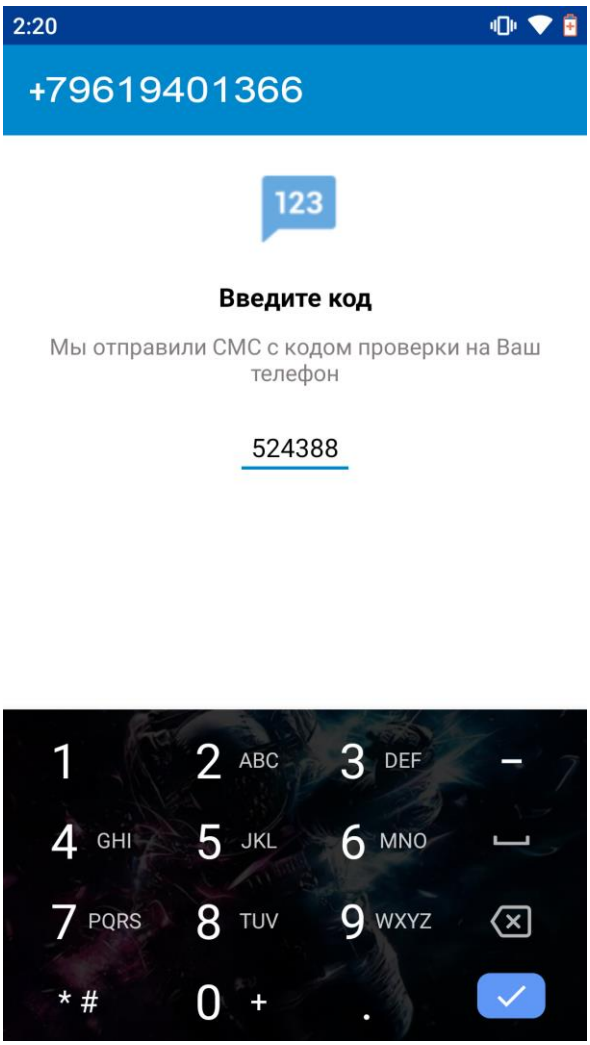

Рисунок 18 – Ввод кода проверки

В меню настроек пользователь может заполнить данные о себе:

– изменить отображаемое имя;

– добавить фотографию;

– изменить номер телефона;

– изменить идентификационное имя;

– заполнить или изменить информацию о себе.

Все введенные пользователем данные будут отображаться в меню настроек, как показано на рисунке 19.

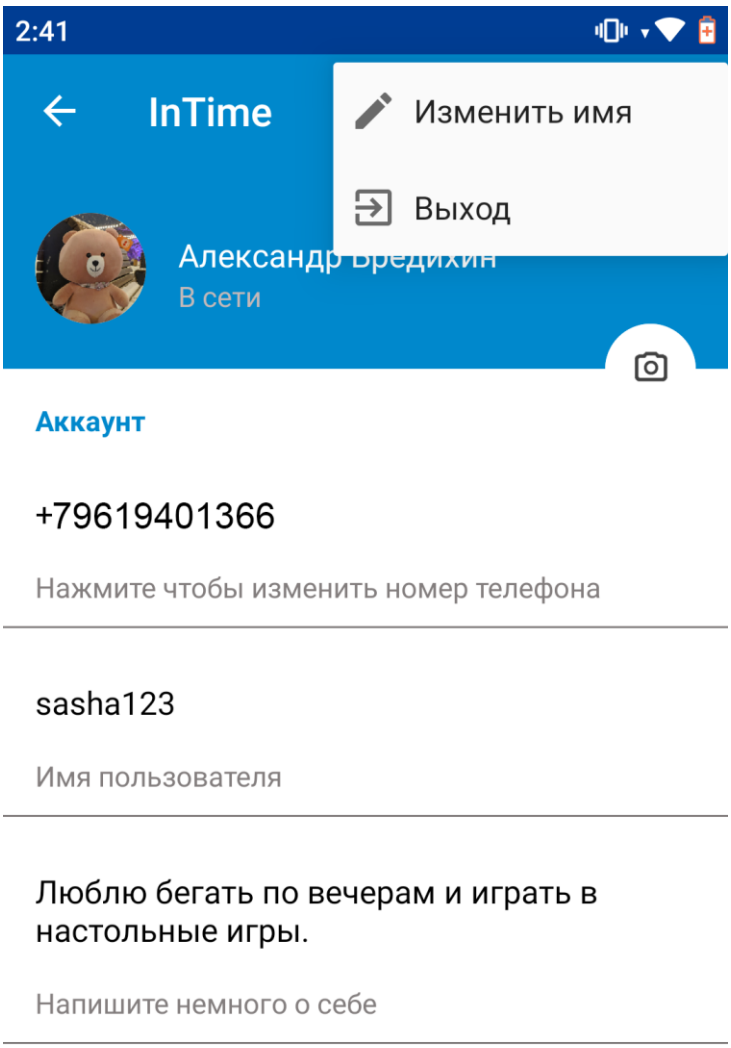

Рисунок 19 – Меню настроек

Главным окном разрабатываемого приложения является «Мероприятия». В данном окне выведены все существующие мероприятия на данный момент, этот список можно сузить, указав фильтры, что находятся выше. На рисунке 20 продемонстрирована главная страница приложения.

Рисунок 20 – Главная страница приложения

При нажатии на название города, открывается карта с меткой, где будет происходить мероприятие, как продемонстрированно на рисунке 21.

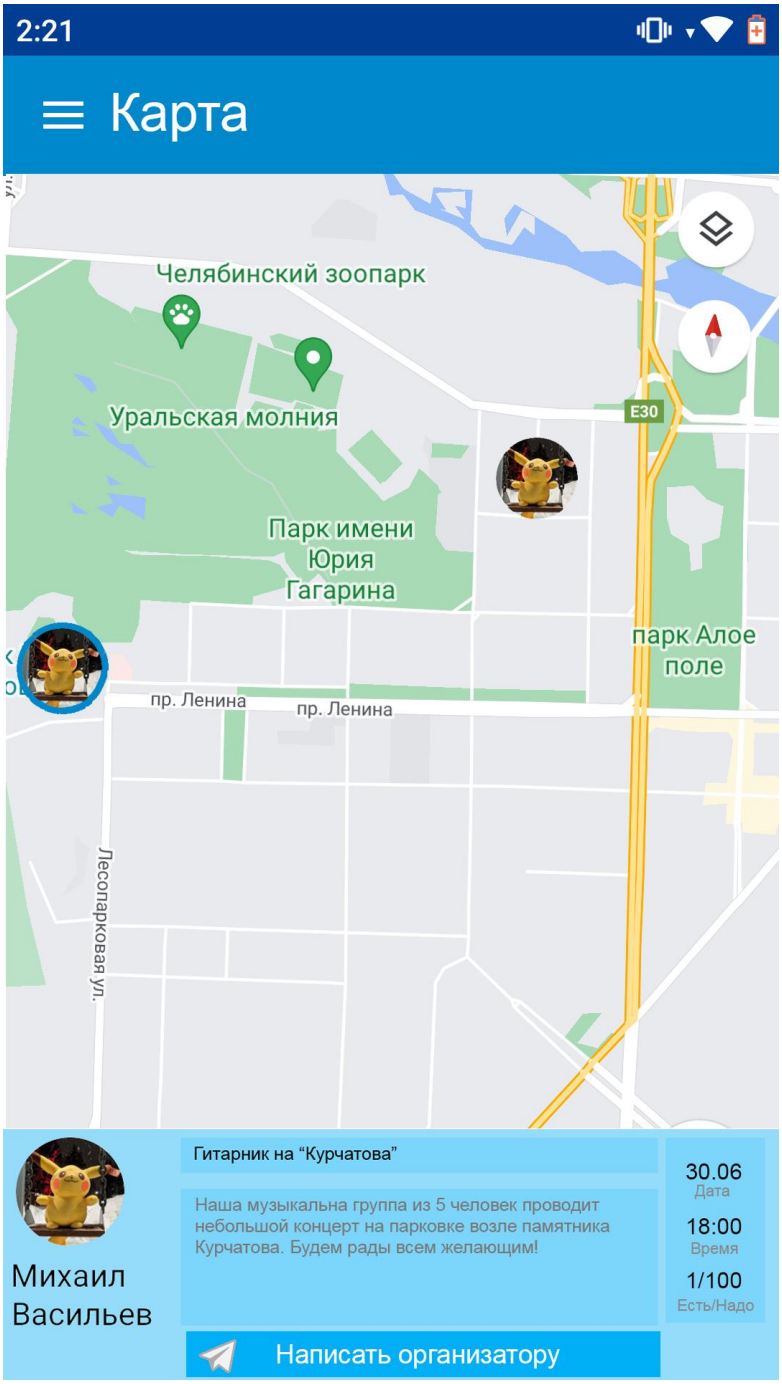

Рисунок 21 – Поиск мероприятий по карте

На данной странице приложения, пользователь видит метку, где будет происходит мероприятие, а также осуществлять поиск других мероприятий по карте.

Авторизованные пользователи могут создавать свои собственные мероприятие, форма для создания мероприятий показана на рисунке 22.

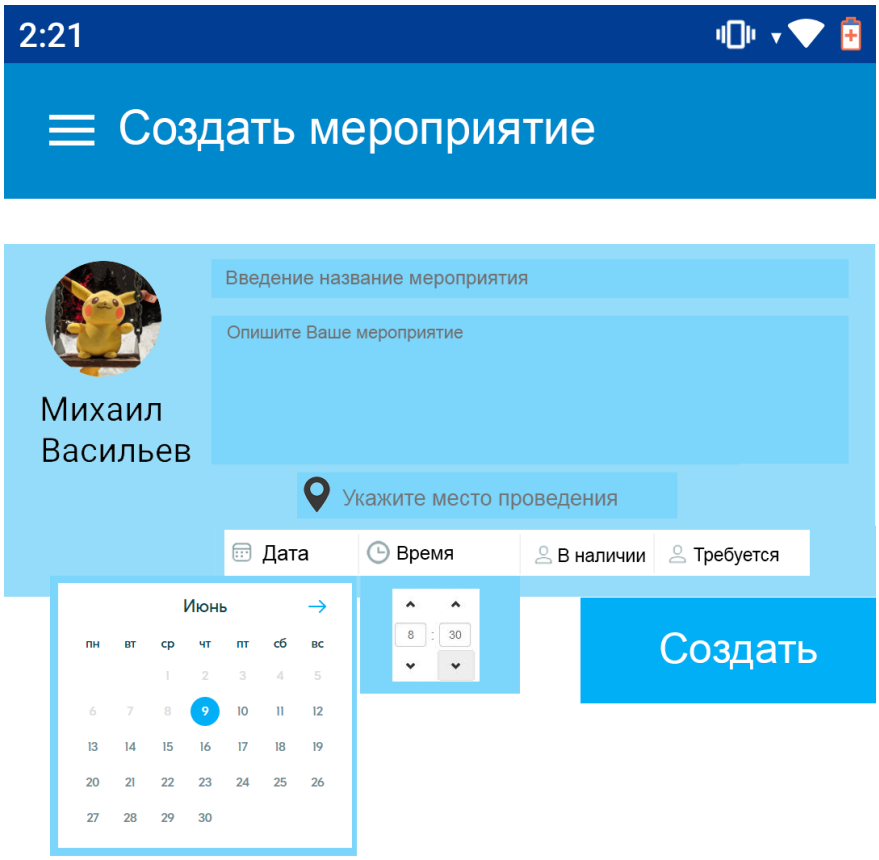

Рисунок 22 – Страница «Создать мероприятие»

На данной странице пользователю необходимо указать:

– название мероприятия;

– описать мероприятие;

– указать место проведения данного мероприятия;

– выбрать дату проведения в открывшейся форме «календарь»;

– указать время;

– указать требуемое количество участников и имеющихся на данный момент.

После введения данных, пользователь нажимает кнопку «Создать», чтобы данное мероприятие отобразилось в списке всех мероприятий, а также появилась метка на карте с данным событием.

На рисунке 23 представлена страница «Контакты».

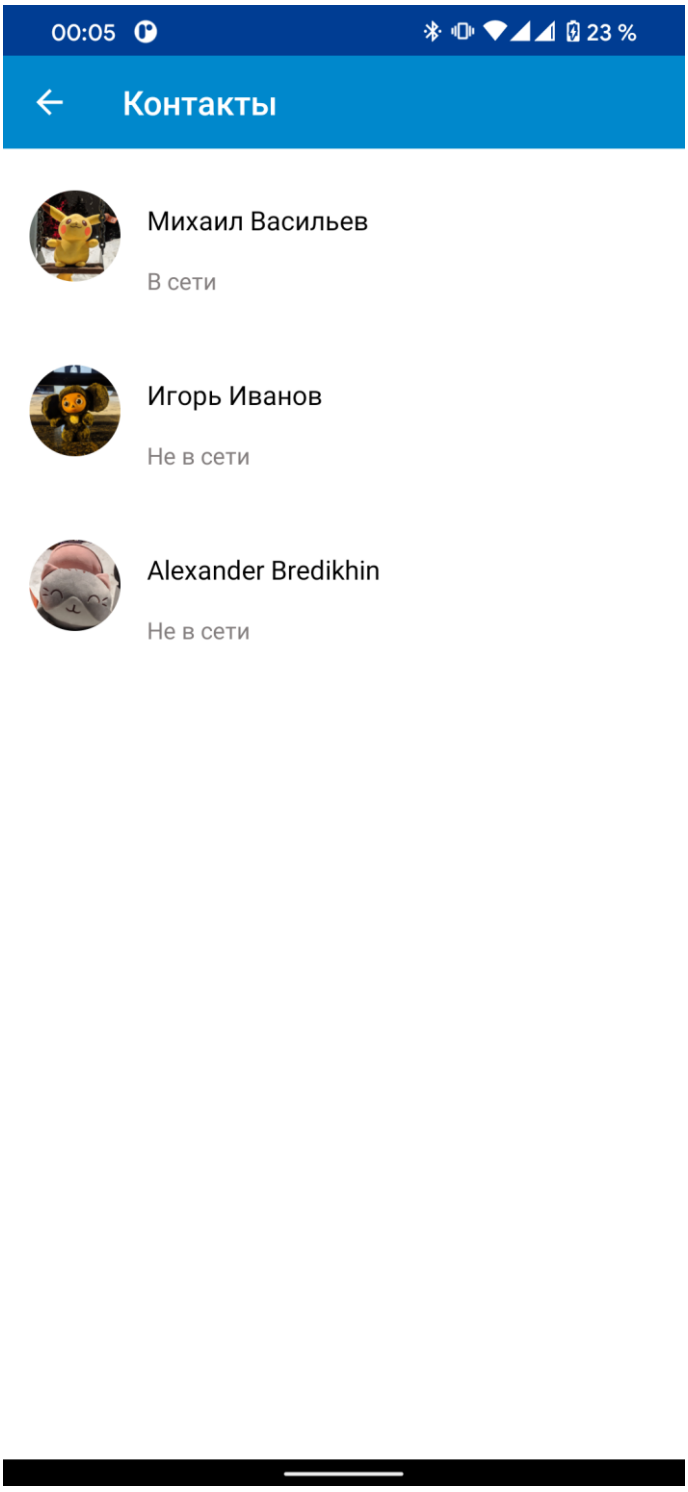

Рисунок 23 – Страница «Контакты»

На данной странице загружается список людей из телефонной книги пользователя, которые пользуются данным приложением. Также на странице «Контакты» отображаются все пользователи, с которыми происходило общение внутри приложения по средством чата.

Прием заявок происходит на отдельной странице «Прием заявок». Пользователю поступает запрос на участие в его мероприятие вместе с

сопроводительным письмом, где другой пользователь пишет почему хочет присоединиться. Данная страница показана на рисунке 24.

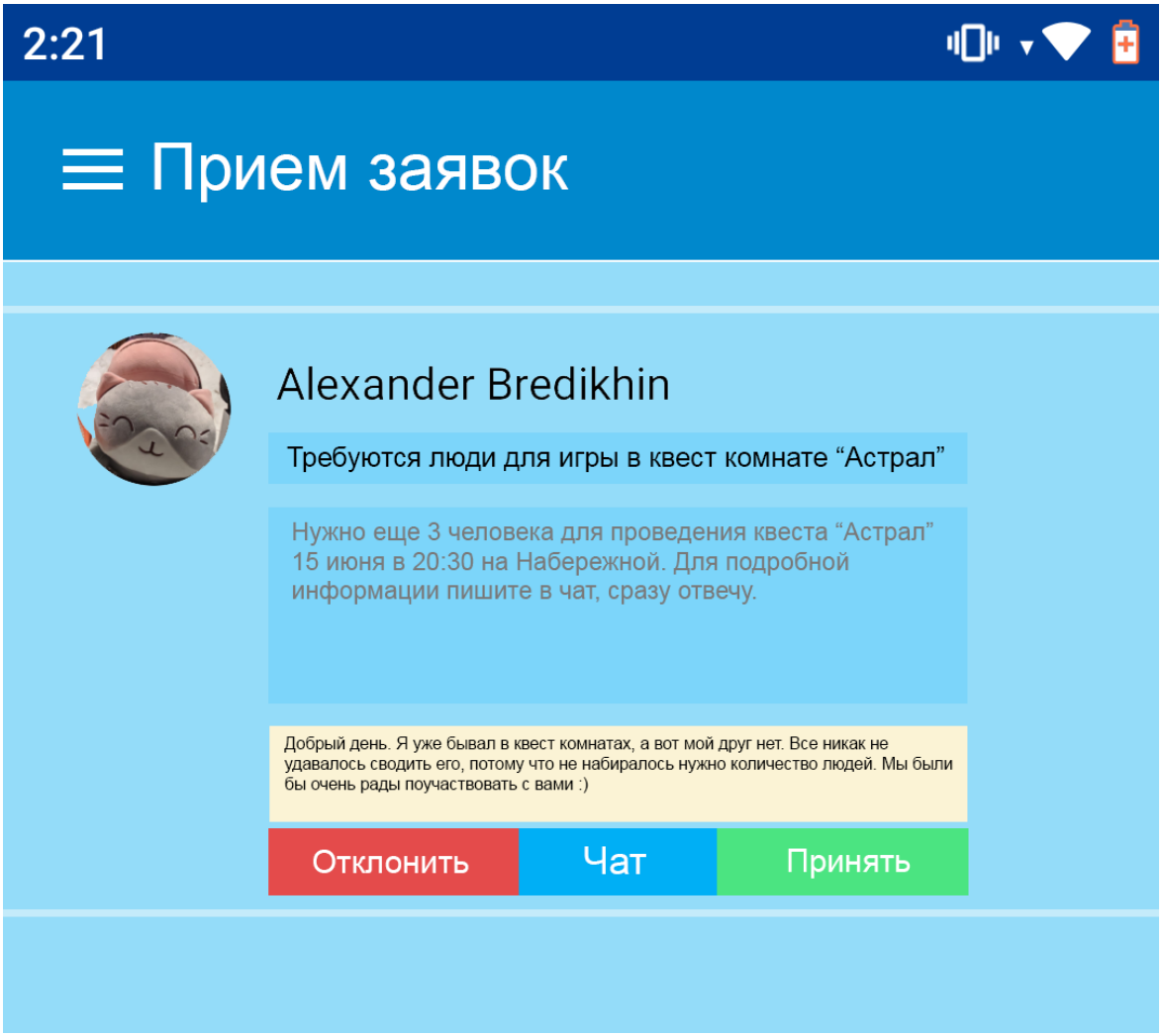

Рисунок 24 – Страница «Контакты»

Пользователь, который организовывает мероприятие может как принять данную заявку, так и отклонить. Имеется возможность перейти в чат с пользователь, который подал заявку на участие.

5 ТЕСТИРОВАНИЕ

В рамках дипломной работы было проведено функциональное тестирование мобильного приложения для организации мероприятий.

5.1 Тестирование формы авторизации

Форма авторизации не позволяет ввести пользователю латинские или арабские буквы. Проведем тестирование разработанного приложения на корректность введённых данных.

Не заполним форму ввода номера телефона и проверим что будет, результат выполнения продемонстрирован на рисунке 25.

Рисунок 25 – Валидация формы заполнения с пустыми данными

На рисунке 26 продемонстрировано тестирование на корректность введённых данных. Введем беспорядочный набор цифр и символов и проверим что будет.

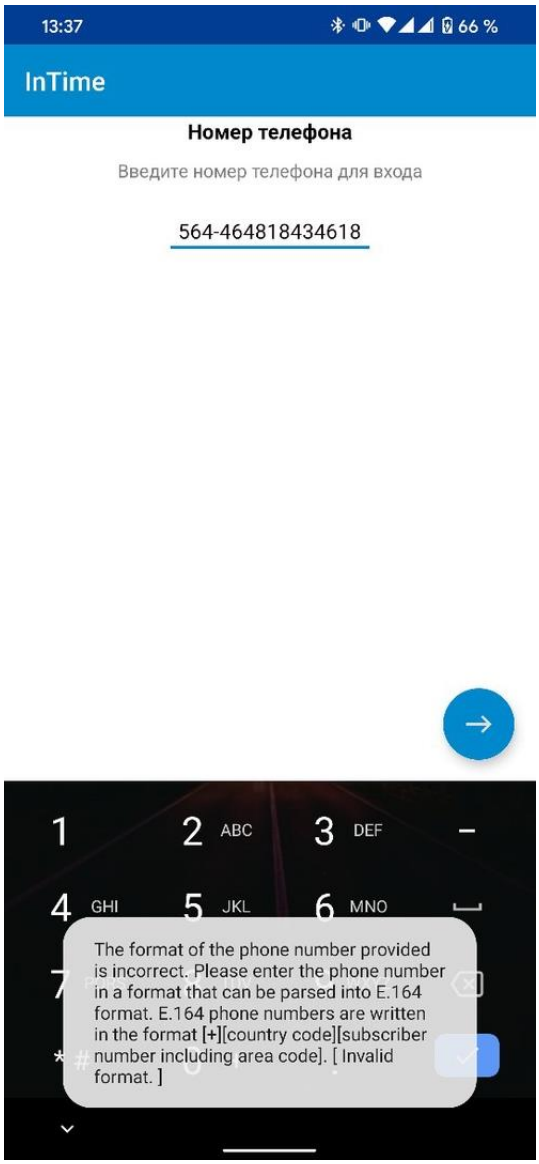

Рисунок 26 – Корректность заполнения данных

Проведем тестирование формы вывода кода аунтификации. Результат тестирования продемонстрирован на рисунке 27. Пользователю все еще запрещено вводить арабские или латинские символы.

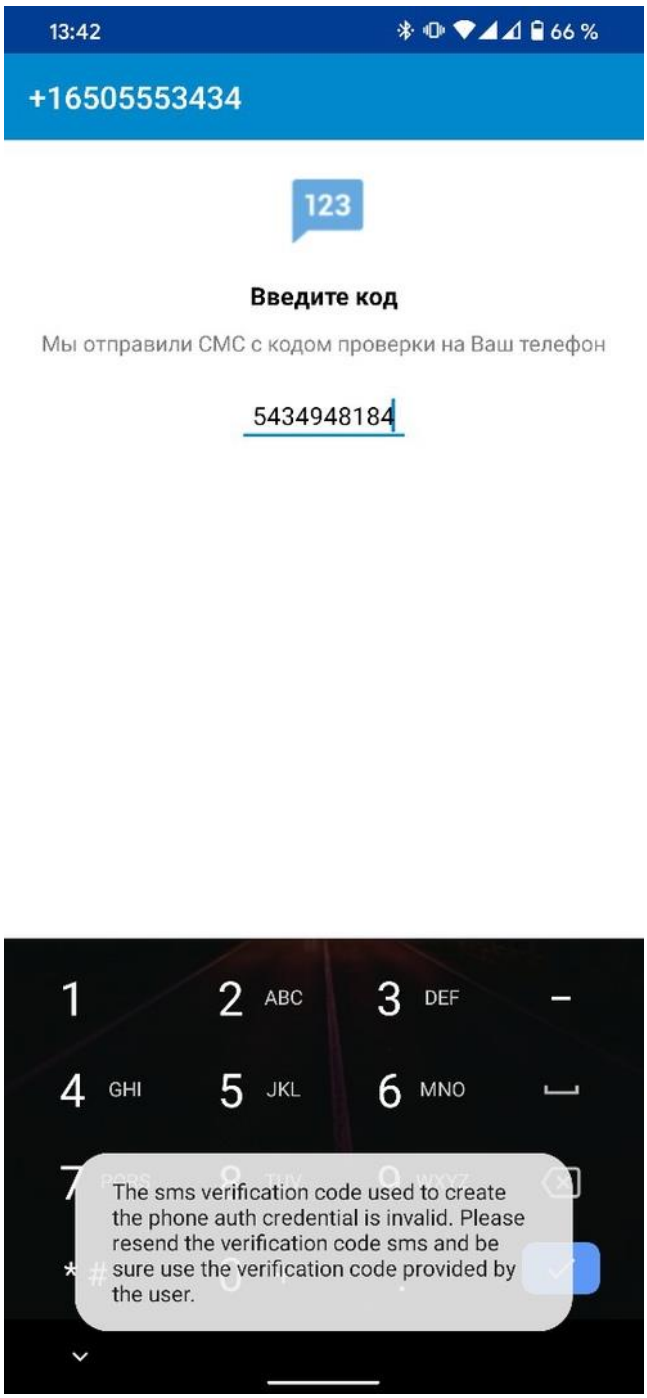

Рисунок 27 – Корректность заполнения кода аутнтификации

5.2 Тестирование формы «Создать мероприятие»

Проверим форму «Создать мероприятие» на отсутствие данных. Результат тестирования показан на рисунке 28.

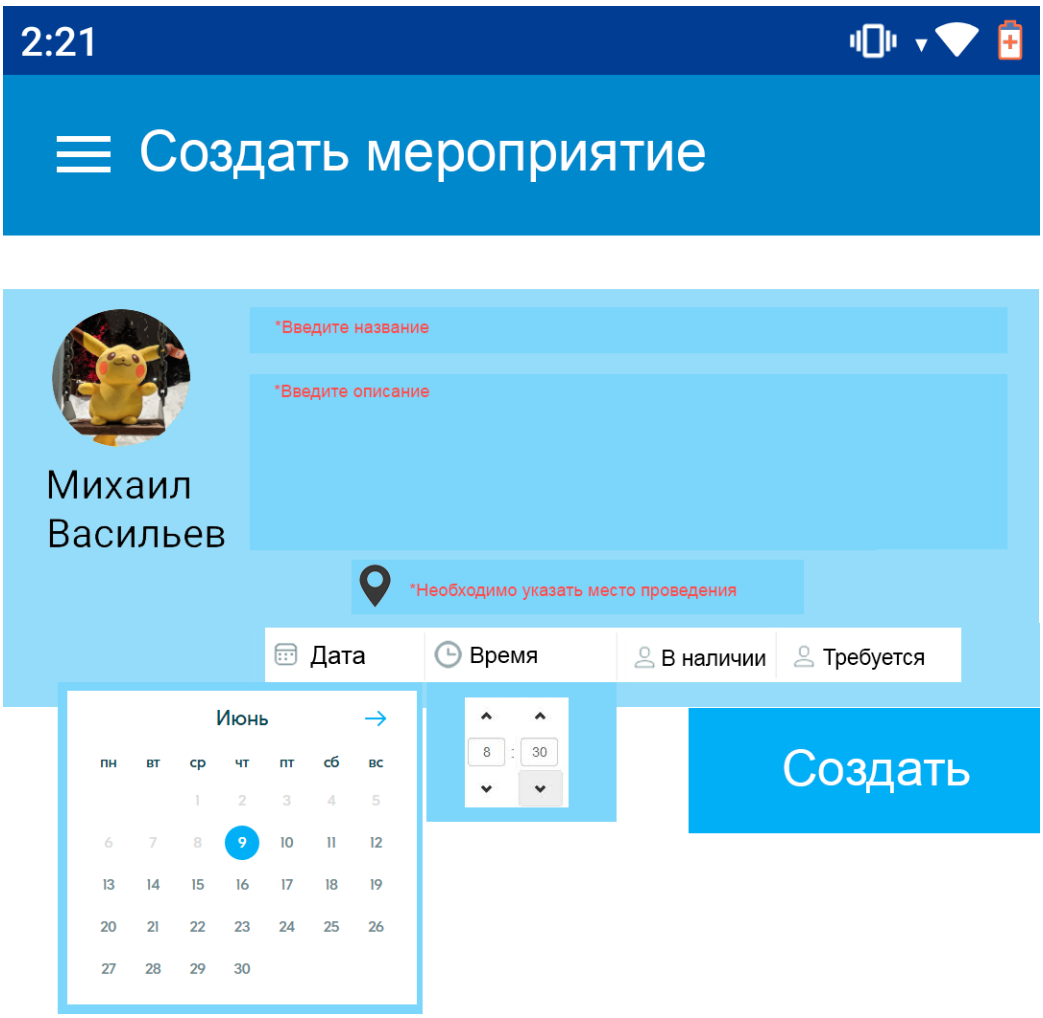

Рисунок 28 – Корректность заполнения данных «Создать мероприятие»

Поля «Название мероприятия», «Описание мероприятия», «Место проведения мероприятия» не могут оставаться незаполненными.

5.3 Тестирование формы редактирования профиля

Попробуем поля «Имя пользователя» и «Напишите немного о себе» оставить пустыми. Результаты тестирования продемонстрированы на рисунках 29, 30.

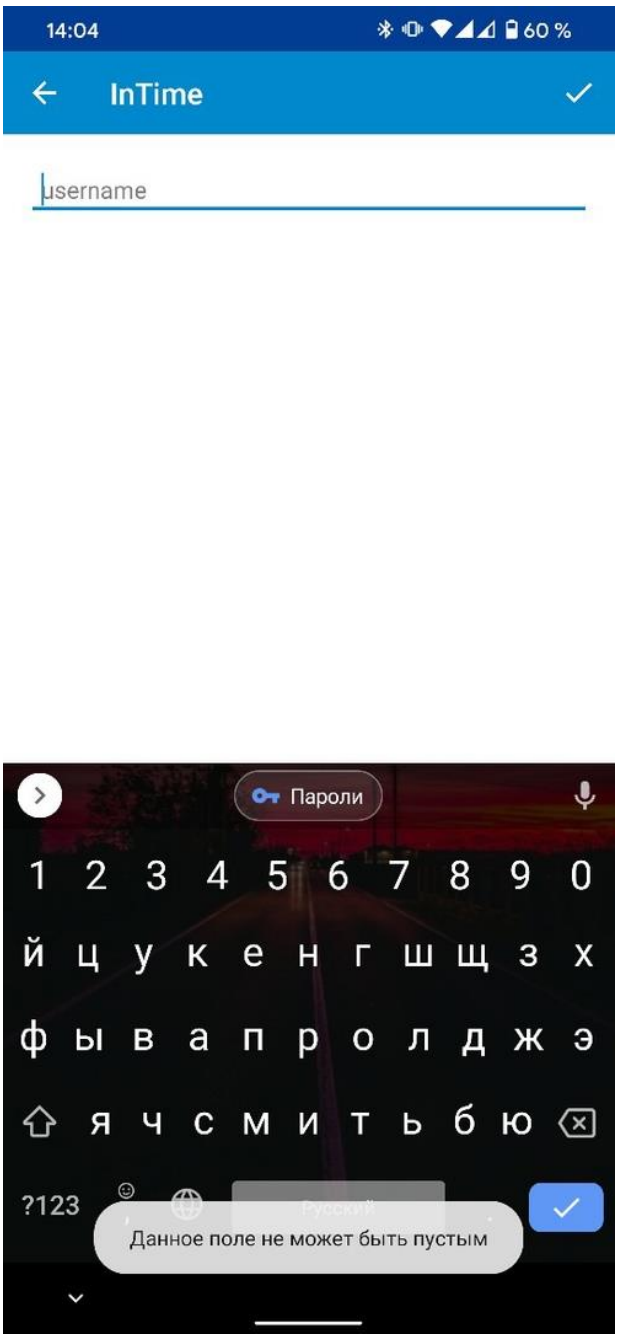

Рисунок 29 – Корректность заполнения «Имя пользователя»

Поле «Имя пользователя» необходимо заполнить, так как данное поле является идентификатором пользователя в приложении.

Рисунок 30 – Корректность заполнения «Напишите немного о себе»

Данное поле можно оставлять пустым, так как пользователь сам решает заполнять информацию о себе или нет. Как видно из рисунка 30 данное поле в профиле отображается пустым.

5.4 Тестирование фильтрации мероприятий в приложение

Фильтрация происходит по заданным пользователем параметрам:

– по городу;

– по дате;

– по времени;

– по количеству требуемых участников.

Заполним данные фильтры и проверим список мероприятий, как показано на рисунке 31.

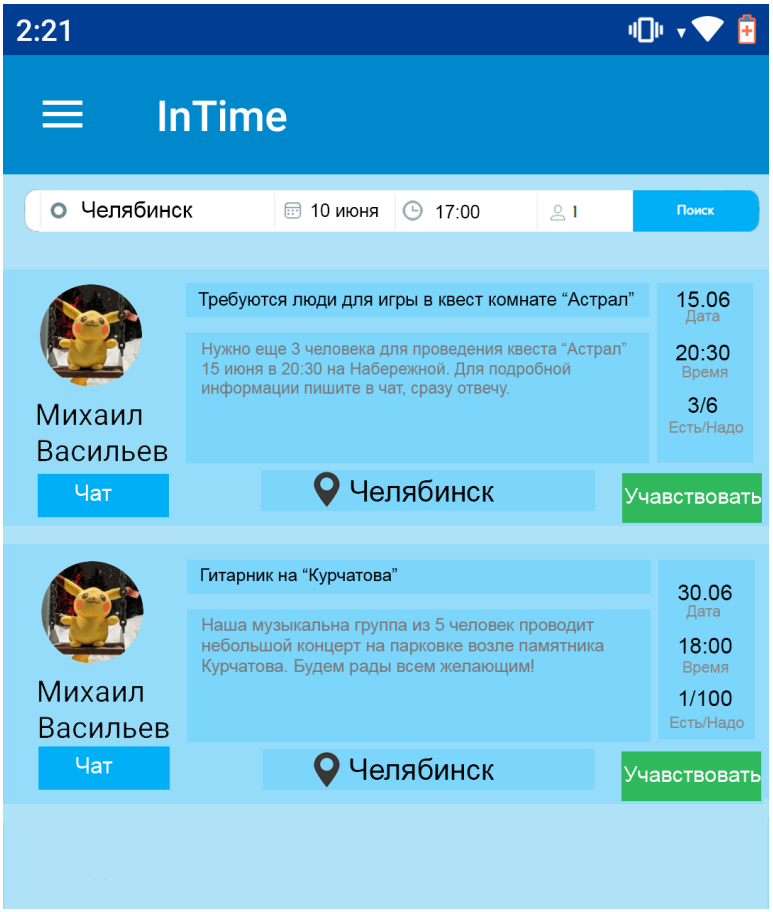

Рисунок 31 – Поиск мероприятий по фильтрам

Нашлось два мероприятия по заданным пользователем фильтрам. Зададим новые фильтры, как показано на рисунке 32.

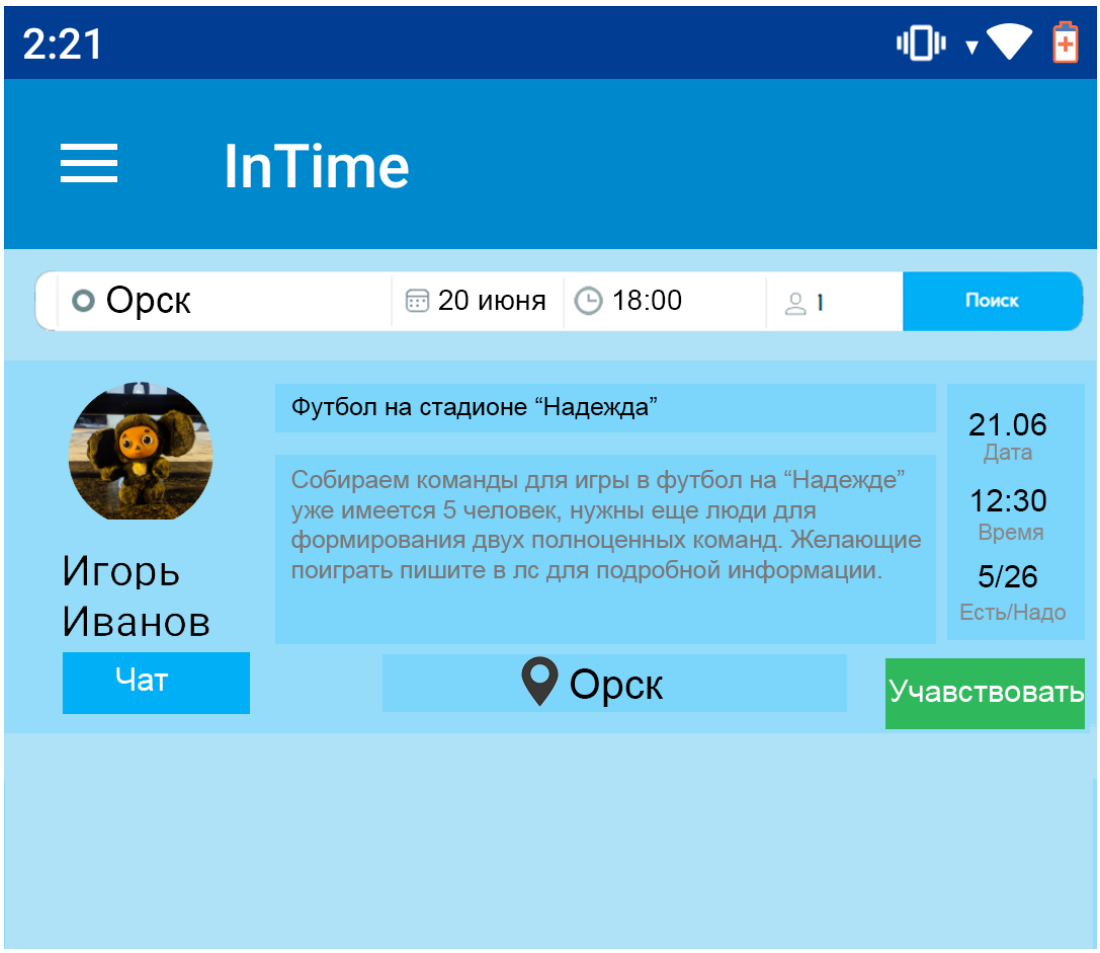

Рисунок 32 – Поиск мероприятий по фильтрам

Фильтр «дата» и «время» указывают точку отсчета, с которой начинается поиск мероприятий, как продемонстрированно на данном рисунке.

5.5 Тестирование отправки и получение ленного сообщения

Проведем тестирования чата внутри приложения и проверим отправку, рисунок 33, и получение сообщений, рисунок 34.

Рисунок 33 – Отправка сообщений

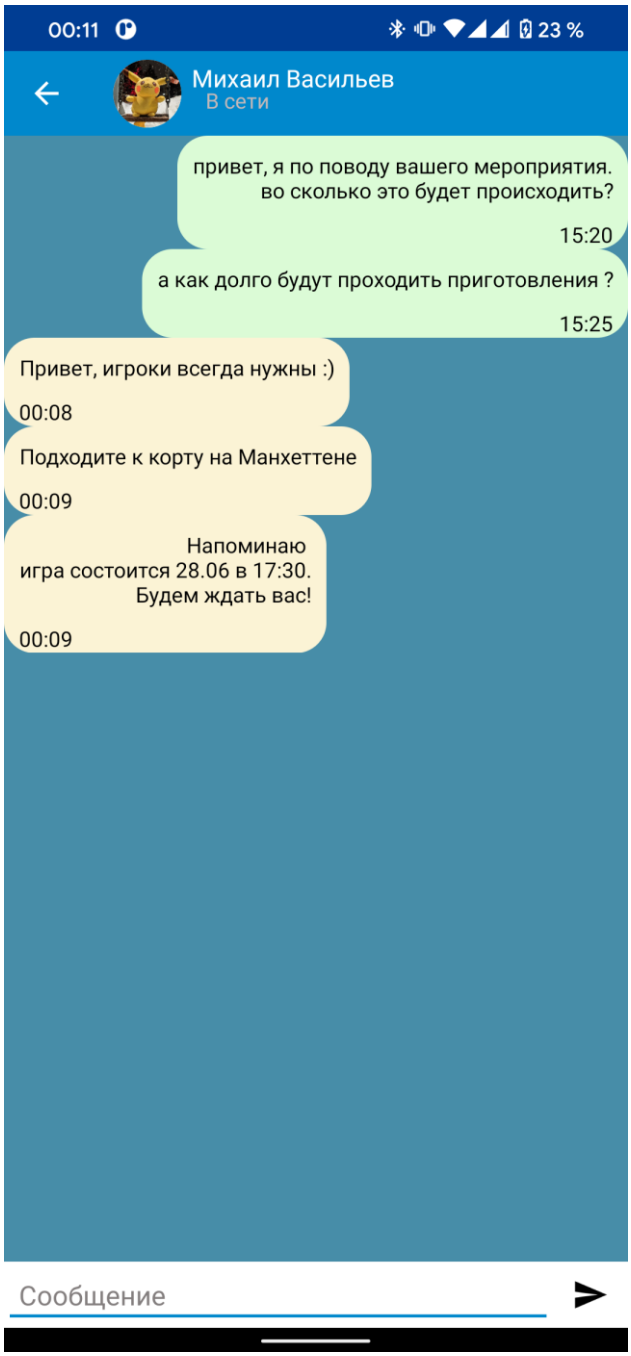

Рисунок 34 – Получение сообщений от другого пользователя

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы было разработано мобильное приложение для организации мероприятий, при этом были решены следующие задачи:

– проведен анализ существующих приложений, обладающих схожим функционалом:

1) Event.Rocks;

2) Eventee;

3) Meetup;

4) Eventzilla;

5) InVitor;

– был определен функционал приложения на основе анализа существующих систем:

1) возможность создавать мероприятие;

2) возможность осуществлять поиск мероприятие;

3) возможность связи между пользователями по средством чата;

4) возможность устанавливать фильтры для поиска;

5) возможность производить поиск мероприятий по карте;

– изучены и выбраны средства для осуществления разработки мобильного приложения для ОС Android:

1) среда разработки Android Studio;

2) язык программирования Kotlin;

– спроектирована разработка приложения;

– разработано приложение;

– проведено функциональное тестирование разработанного приложения.

В дальнейшем планируется расширение функционала приложения:

– добавление звукового уведомления при получение нового сообщения;

– добавление звукового уведомления при получение запроса на участие;

– добавление сортировки мероприятий по категориям.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Статистика мобильных операционных систем. – Текст. Изображение : электронные // Statcount : [сайт]. – 2022. – URL: https://gs.statcounter.com/os-market-share/mobile/worldwide/ (дата обращения: 15.05.22).

2. Event.Rocks. – Текст. Изображение : электронные // Google Play : [сайт]. – 22 февраля 2022. – URL: https://play.google.com/store/apps/details?id=event.rocks.yndx&hl=ru&gl=US (дата обращения: 15.05.22).

3. Eventee. – Текст. Изображение : электронные // Google Play : [сайт]. – 8 июня 2022. – URL: https://play.google.com/store/apps/details?id=com.touchart.eventee&hl=en_US&gl=US (дата обращения: 15.05.22).

4. Meetup. – Текст. Изображение : электронные // Google Play : [сайт]. – 8 июня 2022. – URL: https://play.google.com/store/apps/details?id=com.meetup&hl=ru&gl=US (дата обращения: 16.05.22).

5. Eventzilla. – Текст. Изображение : электронные // Google Play : [сайт]. – 20 мая 2022. – URL: https://play.google.com/store/apps/details?id=com.eventzilla.attendee.app&hl=ru&gl=US (дата обращения: 16.05.22).

6. InVitor. – Текст. Изображение : электронные // Google Play : [сайт]. – 30 мая 2022. – URL: https://play.google.com/store/apps/details?id=com.invitor.client&hl=ru&gl=US (дата обращения: 16.05.22).

7. Статистика использования мобильных ОС. – Текст. Изображение : электронные // Адаптист.ру : [сайт]. – 2022. – URL: http://adaptist.ru/blog/statistikamobile/ (дата обращения: 16.05.22).

8. Мобильный интернет в России. – Текст. Изображение : электронные // Mail.ru Group : [сайт]. – 2022. – URL: https://corp.imgsmail.ru/media/files/40314-researchmobilemail.pdf (дата обращения: 16.05.22).

9. Xamarin.Forms. – Текст. Изображение : электронные // Microsoft.com : [сайт]. – 2022. – URL: https://docs.microsoft.com/ru-ru/xamarin/get-started/what-is-xamarin (дата обращения: 16.05.22).

10. Обзор платформы Eclipse. – Текст. Изображение : электронные // hightech.in.ua : [сайт]. – 2022. – URL: https://hightech.in.ua/content/art-eclipse-platform (дата обращения: 16.05.22).

11. Android Studio. – Текст. Изображение : электронные // javarush.ru : [сайт]. – 2022. – URL: https://javarush.ru/groups/posts/1438-sredih-razrabotki-dlja-android (дата обращения: 16.05.22).

12. Другие языки объектно-ориентированного программирования. – Текст : электронный // melimde : [сайт]. – 2022. – URL: https://melimde.com/semestr-lekciya-1-tema-elementi-interfejsa-delphi-nachalo-rabo.html?page=38 (дата обращения: 16.05.22).

13. Введение в язык Kotlin. – Текст. Изображение : электронные // metanit.com : [сайт]. – 2022. – URL: https://metanit.com/kotlin/tutorial/1.1.php (дата обращения: 16.05.22).

14. Что такое база данных? – Текст. Изображение : электронные // oracle.com : [сайт]. – 2022. – URL: https://www.oracle.com/cis/database/what-is-database/ (дата обращения: 16.05.22).

15. SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems. – Текст. Изображение : электронные // DigitalOcean : [сайт]. – 2022. – URL: https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems (дата обращения: 16.05.22).

16. 1 General Information. – Текст. Изображение : электронные // MySQL : [сайт]. – 2022. – URL: https://dev.mysql.com/doc/refman/8.0/en/introduction.html (дата обращения: 16.05.22).

17. Что такое Firebase. – Текст. Изображение : электронные // kolmogorov.pro : [сайт]. – 2022. – URL: https://kolmogorov.pro/what-is-firebase-chto-takoe (дата обращения: 16.05.22).

ПРИЛОЖЕНИЕ А

Исходный код фрагмента MainActivity

Листинг А.1 – Исходный код MainActivity

```
package com.example.intime

import android.content.pm.PackageManager
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.appcompat.widget.Toolbar
import androidx.core.content.ContextCompat
import com.example.intime.UI.fragments.MainFragment
import com.example.intime.UI.fragments.register.EnterPhoneNumberFragment
import com.example.intime.UI.objects.AppDrawer
import com.example.intime.databinding.ActivityMainBinding
import com.example.intime.utilits.*
import kotlinx.coroutines.CoroutineScope
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.launch


class MainActivity : AppCompatActivity() {

    private lateinit var mainBinding: ActivityMainBinding
    lateinit var mToolbar: Toolbar
    lateinit var mAppDrawer: AppDrawer

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        mainBinding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(mainBinding.root)
        APP_ACTIVITY = this
        initFirebase()
        initUser{
            CoroutineScope(Dispatchers.IO).launch {
                initContacts()
            }
            initFields()
            initFunc()
        }
    }


    private fun initFunc() {
        setSupportActionBar(mToolbar)
        if (AUTH.currentUser != null) {
            mAppDrawer.create()
            replaceFragment(MainFragment(), false)
        } else {
            replaceFragment(EnterPhoneNumberFragment(), false)
        }
    }


    private fun initFields() {
        mToolbar = mainBinding.mainToolbar
        mAppDrawer = AppDrawer()

    }
```

```
        override fun onStart() {
        super.onStart()
        AppStates.updateState(AppStates.ONLINE)
    }

    override fun onStop() {
        super.onStop()
        AppStates.updateState(AppStates.OFFLINE)
    }

    override fun onRequestPermissionsResult(
        requestCode: Int,
        permissions: Array<out String>,
        grantResults: IntArray
    ) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults)
        if(ContextCompat.checkSelfPermission(APP_ACTIVITY,
READ_CONTACT)==PackageManager.PERMISSION_GRANTED){
            initContacts()
        }
    }

}
```

ПРИЛОЖЕНИЕ Б

Исходный код фрагмента AppDrawer

Листинг Б.1 – Исходный код AppDrawer

```
package com.example.intime.UI.objects

import android.graphics.drawable.Drawable
import android.net.Uri
import android.view.View
import android.widget.ImageView
import androidx.appcompat.app.AppCompatActivity
import androidx.appcompat.widget.Toolbar
import androidx.drawerlayout.widget.DrawerLayout
import com.example.intime.R
import com.example.intime.UI.fragments.ContactsFragment
import com.example.intime.UI.fragments.CreateMeroFragment
import com.example.intime.UI.fragments.MainFragment
import com.example.intime.UI.fragments.SettingsFragment
import com.example.intime.utilits.APP_ACTIVITY
import com.example.intime.utilits.USER
import com.example.intime.utilits.downloadAndSetImage
import com.example.intime.utilits.replaceFragment
import com.mikepenz.materialdrawer.AccountHeader
import com.mikepenz.materialdrawer.AccountHeaderBuilder
import com.mikepenz.materialdrawer.Drawer
import com.mikepenz.materialdrawer.DrawerBuilder
import com.mikepenz.materialdrawer.model.DividerDrawerItem
import com.mikepenz.materialdrawer.model.PrimaryDrawerItem
import com.mikepenz.materialdrawer.model.ProfileDrawerItem
import com.mikepenz.materialdrawer.model.interfaces.IDrawerItem
import com.mikepenz.materialdrawer.util.AbstractDrawerImageLoader
import com.mikepenz.materialdrawer.util.DrawerImageLoader

class AppDrawer() {
    private lateinit var mDrawer: Drawer
    private lateinit var mHeader: AccountHeader
    private lateinit var mDrawerLayout: DrawerLayout
    private lateinit var mCurrentProfile: ProfileDrawerItem

    fun create() {
        inintLoader()
        createHeader()
        createDrawer()
        mDrawerLayout = mDrawer.drawerLayout
    }

    fun disableDrawer() {
        mDrawer.actionBarDrawerToggle?.isDrawerIndicatorEnabled = false
        APP_ACTIVITY.supportActionBar?.setDisplayHomeAsUpEnabled(true)
        mDrawerLayout.setDrawerLockMode(DrawerLayout.LOCK_MODE_LOCKED_CLOSED)
        APP_ACTIVITY.mToolbar.setNavigationOnClickListener {
            APP_ACTIVITY.supportFragmentManager.popBackStack()
        }
    }

    fun enableDrawer() {
        APP_ACTIVITY.supportActionBar?.setDisplayHomeAsUpEnabled(false)
        mDrawer.actionBarDrawerToggle?.isDrawerIndicatorEnabled = true
        mDrawerLayout.setDrawerLockMode(DrawerLayout.LOCK_MODE_UNLOCKED)
        APP_ACTIVITY.mToolbar.setNavigationOnClickListener {
            mDrawer.openDrawer()
        }
```

```
    private fun createDrawer() {
        mDrawer = DrawerBuilder()
            .withActivity(APP_ACTIVITY)
            .withToolbar(APP_ACTIVITY.mToolbar)
            .withActionBarDrawerToggle(true)
            .withSelectedItem(-1)
            .withAccountHeader(mHeader)
            .addDrawerItems(
                PrimaryDrawerItem().withIdentifier(100)
                    .withIconTintingEnabled(true)
                    .withName("Чат")
                    .withSelectable(false)
                    .withIcon(R.drawable.ic_chat),
                PrimaryDrawerItem().withIdentifier(101)
                    .withIconTintingEnabled(true)
                    .withName("Карта")
                    .withSelectable(false)
                    .withIcon(R.drawable.ic_map),
                PrimaryDrawerItem().withIdentifier(102)
                    .withIconTintingEnabled(true)
                    .withName("Создать мероприятие")
                    .withSelectable(false)
                    .withIcon(R.drawable.ic_plus),
                PrimaryDrawerItem().withIdentifier(103)
                    .withIconTintingEnabled(true)
                    .withName("Мои заявки")
                    .withSelectable(false)
                    .withIcon(R.drawable.ic_menu_create_groups),
                PrimaryDrawerItem().withIdentifier(104)
                    .withIconTintingEnabled(true)
                    .withName("Прием заявок")
                    .withSelectable(false)
                    .withIcon(R.drawable.ic_menu_invate),
                DividerDrawerItem(),
                PrimaryDrawerItem().withIdentifier(105)
                    .withIconTintingEnabled(true)
                    .withName("Настройки")
                    .withSelectable(false)
                    .withIcon(R.drawable.ic_menu_settings)

            ).withOnDrawerItemClickListener(object :
Drawer.OnDrawerItemClickListener {
                override fun onItemClick(
                    view: View?,
                    position: Int,
                    drawerItem: IDrawerItem<*>
                ): Boolean {
                    clickToItem(position)
                    return false
                }
            }).build()
    }

    private fun clickToItem(position:Int){
        when (position) {
            1 -> replaceFragment(ContactsFragment())
            2 -> replaceFragment(MapFragment())
            3 -> replaceFragment(CreateMeroFragment())
            4 -> replaceFragment(MyRequestsFragment())
            5 -> replaceFragment(ClaimRequestsFragment())
            7 -> replaceFragment(SettingsFragment())}}
```

```
     private fun createHeader() {
        mCurrentProfile = ProfileDrawerItem()
            .withName(USER.fullname)
            .withEmail(USER.phone)
            .withIcon(USER.photoUrl)
            .withIdentifier(200)
        mHeader = AccountHeaderBuilder()
            .withActivity(APP_ACTIVITY)
            .withHeaderBackground(R.drawable.header)
            .addProfiles(
                mCurrentProfile
            ).build()
    }

    fun uptadeHeader() {
        mCurrentProfile
            .withName(USER.fullname)
            .withEmail(USER.phone)
            .withIcon(USER.photoUrl)
        mHeader.updateProfile(mCurrentProfile)
    }

    private fun inintLoader(){
        DrawerImageLoader.init(object : AbstractDrawerImageLoader(){
            override fun set(imageView: ImageView, uri: Uri, placeholder:
Drawable) {
                imageView.downloadAndSetImage(uri.toString())
            }
        })
    }
}
```

ПРИЛОЖЕНИЕ В

Исходный код фрагмента SettingsFragment

Листинг Б.1 – Исходный код SettingsFragment

```
package com.example.intime.UI.fragments

import android.app.Activity.RESULT_OK
import android.content.Intent
import android.net.Uri
import android.view.Menu
import android.view.MenuInflater
import android.view.MenuItem
import com.example.intime.R
import com.example.intime.utilits.*
import com.google.firebase.storage.StorageReference
import com.theartofdev.edmodo.cropper.CropImage
import com.theartofdev.edmodo.cropper.CropImageView
import kotlinx.android.synthetic.main.fragment_settings.*


class SettingsFragment : BaseFragment(R.layout.fragment_settings) {

    override fun onResume() {
        super.onResume()
        setHasOptionsMenu(true)
        initFields()
    }

    private fun initFields() {
        settings_bio.text = USER.bio
        settings_full_name.text = USER.fullname
        settings_phone_number.text = USER.phone
        settings_status.text = USER.state
        settings_username.text = USER.username
        settings_btn_change_username.setOnClickListener {
replaceFragment(ChangeUsernameFragment()) }
        settings_btn_change_bio.setOnClickListener {
replaceFragment(ChangeBioFragment()) }
        settings_change_photo.setOnClickListener { changePhotoUser() }
        settings_user_photo.downloadAndSetImage(USER.photoUrl)
    }

    private fun changePhotoUser() {
        CropImage.activity()
            .setAspectRatio(1, 1)
            .setRequestedSize(600, 600)
            .setCropShape(CropImageView.CropShape.OVAL)
            .start(APP_ACTIVITY, this)
    }


    override fun onCreateOptionsMenu(menu: Menu, inflater: MenuInflater) {
        activity?.menuInflater?.inflate(R.menu.settings_action_menu, menu)
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        when (item.itemId) {
            R.id.settings_menu_exit -> {
                AppStates.updateState(AppStates.OFFLINE)
                AUTH.signOut()
                restartActivity()
            }
```

```
      R.id.settings_menu_change_name -> replaceFragment(ChangeNameFragment())
        }
        return true
    }

    override fun onActivityResult(requestCode: Int, resultCode: Int, data:
Intent?) {
        super.onActivityResult(requestCode, resultCode, data)
        if (requestCode == CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE
            && resultCode == RESULT_OK && data != null
        ) {
            val uri: Uri = CropImage.getActivityResult(data).uri
            val path: StorageReference =
REF_STORAGE_ROOT.child(FOLDER_PROFILE_IMAGE)
                .child(CURRENT_UID)
            putImageToStorage(uri, path) {
                getUrlFromStorage(path) {
                    putUrlToDatabase(it) {
                        settings_user_photo.downloadAndSetImage(it)
                        showToast(getString(R.string.toast_data_update))
                        USER.photoUrl = it
                        APP_ACTIVITY.mAppDrawer.uptadeHeader()
                    }
                }
            }
        }
    }
}
```