

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

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Мессенджер

Курсовой проект

09.03.04 Программная инженерия

Допущен к защите

Заказчик \_\_\_\_\_ В.С. Тарасов \_\_\_\_\_.20\_\_

Обучающийся \_\_\_\_\_ Н. А. Кожевников, 3 курс, д/о

Обучающийся \_\_\_\_\_ Д. А. Лепехин, 3 курс, д/о

Воронеж 2020

# Оглавление

Оглавление	2
Введение	3
1 Постановка задачи	4
2 Анализ предметной области	5
2.1 Анализ существующих решений	5
2.2 Преимущества	6
3 Анализ задачи	6
3.1 Варианты использования приложения	6
3.2 Варианты состояния системы	8
3.3 Варианты действия в системе	10
4 Реализация	16
4.1 Сущности	16
4.2 Серверная часть приложения	16
4.3 Графический интерфейс	17
4.3.1 Экран диалогов	18
4.3.2 Экран переписки	19
5 Тестирование	21

## Введение

Сегодня очень легко связаться с любым человеком, после изобретения интернета расстояния стали намного меньше. Можно написать человеку на электронную почту или в соц сети, но не все быстро отвечают на электронные письма из-за принятой вежливой формы общения и заполнения множества посторонних строк таких как получатель, тема письма и обязательная вежливая подпись внизу. Так же можно написать в соц сети но многие не хотят давать свои контакты в соц сети из за личных фотографий, групп к которым они принадлежат, музыки которую слушают. Хороший выбором в этой ситуации являются мобильные мессенджеры в них предельно простая регистрация не нужно лишней информации в некоторых ты просто можешь ввести логин и начать переписываться с такими же пользователями. Есть мессенджеры с более ответственной регистрацией где нужно ввести логин, пароль и имя учетной записи (имя которое все в мессенджере будут видеть). Смартфон всегда под рукой если вам кто-то написал, он оповестит вас об этом и вы можете так же легко ответить.

Желаемое приложение должно осуществлять быстрый обмен сообщениями между зарегистрированными пользователями. Приложения должны быть легковесным и предоставлять следующую функциональность:

- Обмен сообщениями между пользователями
- Возможность добавить пользователя в друзья
- Возможность добавлять сообщения в категорию избранное

Данная курсовая работа посвящена именно такому приложению с легким инструментарием и минимально необходимыми функциями.

# 1 Постановка задачи

Цель курсовой работы: реализовать Android приложение, которое позволит обмениваться сообщениями сразу после входа в приложение без ненужных действий. Приложение должно работать в фоне и иметь следующие основные функции:

- Стабильная работа на базе операционной системы android 4.1 и выше;
- Интуитивный пользовательский интерфейс;
- Возможность выполнения основных задач приложения:
  - Создание пользователя;
  - Возможность добавление друзей;
  - Возможность удалять друзей;
  - Создание диалогов с несколькими пользователями;
  - Создание переписки с другом;
  - Удаления переписки с другом;
  - Отправка картинок сообщением;
- MVC архитектура приложения с разделением на front-end и back-end;

Для достижения данной цели были выделены следующие задачи:

1. Разработка Front-end части приложения, находящиеся на телефоне пользователя;
2. Использование Back-end(Firebase), на удаленном сервере приложений;
3. Создание связи между Front-end и Back-end частями приложения;
4. Настройка базы данных, расположенной на удаленном сервере.

## 2 Анализ предметной области

**Мессенджер** — это программа (приложение) смартфона, позволяющая мгновенно обмениваться с друзьями текстовыми сообщениям.

**CourseWorkMessenger** — классический мессенджер с возможностью добавлять сообщения в избранное и просматривать их отдельно от диалогов.

### 2.1 Анализ существующих решений

- **Viber** — Мобильное VoIP приложение для Смартфонов. Интегрируется в адресную книгу и авторизует по номеру телефона. Позволяет осуществлять бесплатные звонки в высоком качестве между смартфонами с установленным Viber. Можно звонить на обычные и мобильные телефоны по низким тарифам.
- **WhatsApp** — Кроссплатформенный мобильный чат. Есть функция группового чата, передачи изображений.
- **Telegram** — Защищенный мобильный мессенджер. Умеет синхронизировать сообщения и файлы между мобильной и десктоп-версиями, отображать статусы сообщений (прочитано или нет) и организовывать большие групповые чаты.
- **ICQ** — Популярный на отечественном интернет-пространстве мессенджер с возможностью аудио и видеосвязи. Работает под Windows и Mac, имеет приложения для мобильных устройств на всех популярных платформах (iOS, Android, Symbian, Java, BlackBerry, Windows Mobile).

### 2.2 Преимущества

Возможность добавления сообщений в избранное и просмотр отдельно от чатов.

### 3 Анализ задачи

#### 3.1 Варианты использования приложения

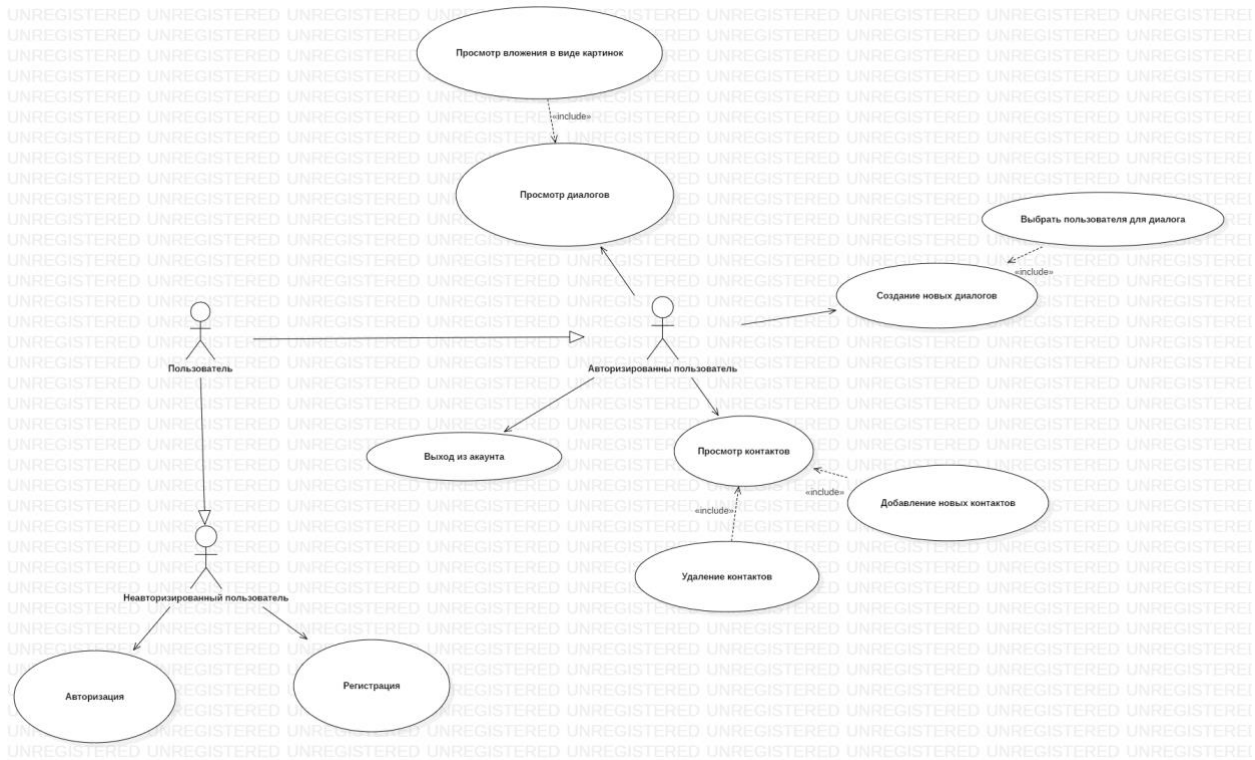


Рисунок 1 –Диаграммы прецедентов

При взаимодействии с приложением у пользователя есть определенный список возможностей, который более наглядно изображен на рисунке 1.

- Авторизация
- Регистрация
- Просмотр контактов
  - Удаление контактов
  - Добавление новых аккаунтов
- Просмотр диалогов
  - Просмотр вложения в виде картинок
- Создание новых диалогов

- Выбор пользователя для диалога
- **Варианты состояний в системе**

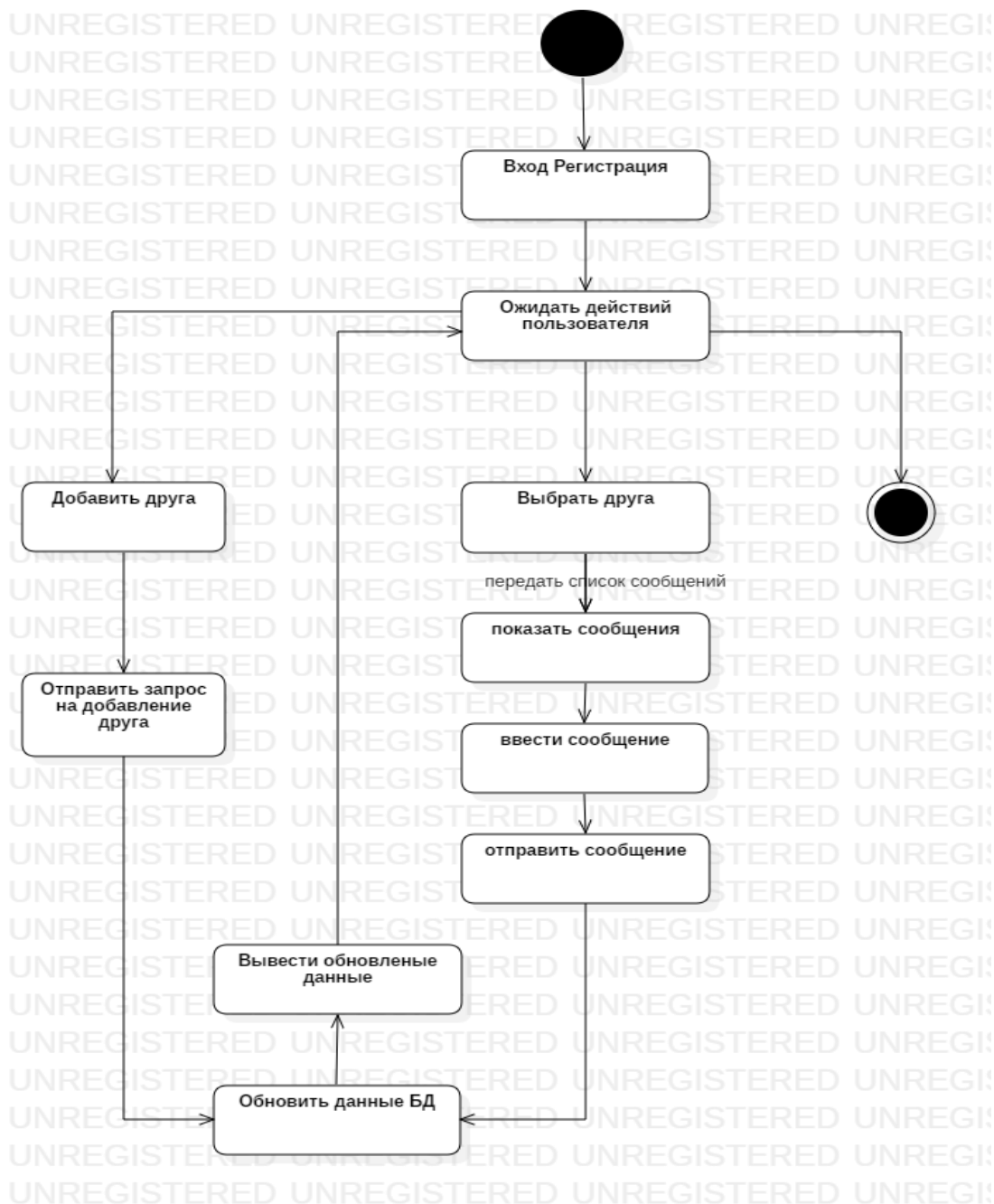


Рисунок 2 – Диаграмма состояний

Диаграмма состояний, изображенная на Рисунке №, отражает возможные состояния системы. При запуске приложения система находится на экране Входа/Регистрации. В зависимости от выбора пользователя возможны 3 основные цепочки состояний:

- Войти
- Зарегистрироваться
- Выйти из приложения

При входе в приложение пользователь должен авторизоваться или создать новый аккаунт путем ввода логина (логином будет имя его личного профиля в мессенджере) и пароля. После чего пользователь попадает на экран переписок. После система переходит в состояние ожидания действия пользователя.

В зависимости от выбора пользователя возможны 3 основные цепочки состояний:

- Добавить друга
- Выбрать друга для переписки
- Открыть переписку

Если пользователь выбрал добавление друга, то система переходит в состояние отправки запроса на добавления друга, если пользователь которому пришел запрос его подтверждает, то система переходит в состояние обновления база данных.

Если пользователь выбрал друга для переписки, то система покажет ему последние сообщения, если сообщений нет, система предложит начать переписку с выбранным другом. Система перейдет в ожидание ввода сообщения пользователем после того как пользователь нажал на отправку сообщения система перейдет в состояние отправки сообщения если сообщение отправлено удачно то оно дойдет до получателя если возникла какая-то ошибка, то система выдаст ошибку и предложит действия.

После входа пользователь может сразу выйти, не производя каких-либо действий с системой.

## ○ **Варианты действия в системе**



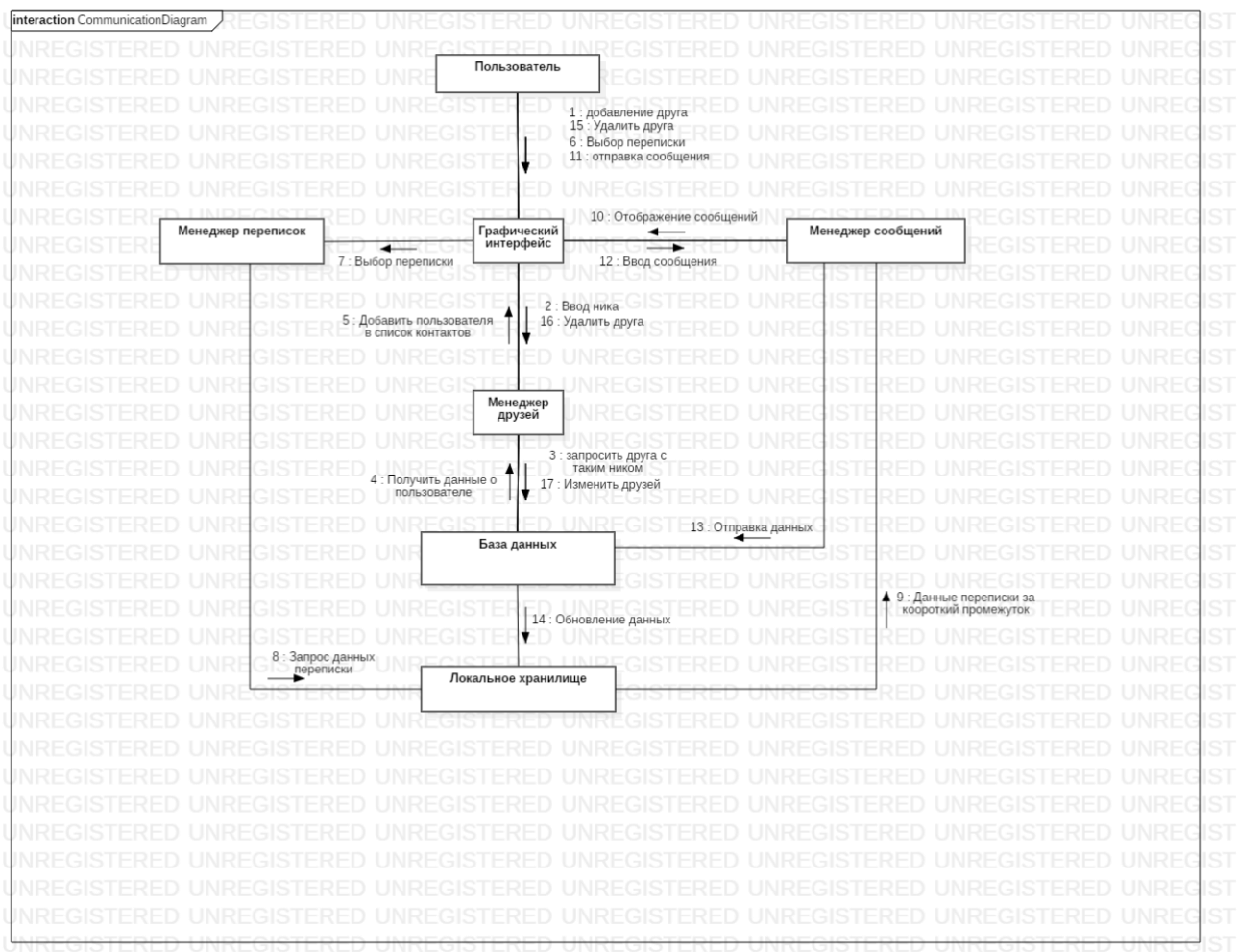


Рисунок 3 – Диаграмма активности.

На рисунке №, показана диаграмма активностей, на которой отображаются возможные действия, которые описаны на диаграмме состояний.

Для добавления нового друга пользователь А обращается к графическому интерфейсу приложения, который в свою очередь просит менеджера друзей добавить соответствующего пользователя Б в список друзей пользователя А. Если пользователь Б подтвердил, что он принимает запрос от пользователя А. Происходит обновление удаленной базы данных и добавление данных в локальную базу данных и обновление списка друзей.

Для начала переписки пользователь обращается к графическому интерфейсу приложения, который в свою очередь просит менеджера переписок обратиться к удаленной базе данных, которая возвращает данные менеджеру сообщений за последний день. Если сообщений в удаленной базе нет, он

вернет пустой список и графический интерфейс просигнализирует что пользователь не разу не писал выбранному другу и предложит ввести сообщение если сообщение было введено, менеджер сообщений отправит его на удаленную базу которая в свою очередь обновит локальную базу данных о обновить список сообщений у обоих пользователей.

Для удаления друга пользователь обращается к графическому интерфейсу , который в свою очередь просит менеджер друзей обновить список друзей на удалённой базе данных. А потом изменит список друзей на локальной базе данных. Сообщения переписки будут хранится еще некоторое время если пользователь удалил друга случайно и восстановить сообщения было бы легко.

Если пользователь не долгое время не заходил в сеть его аккаунт замораживается и после очень долгого периода времени удаляется.

## ○ Взаимодействие компонентов системы

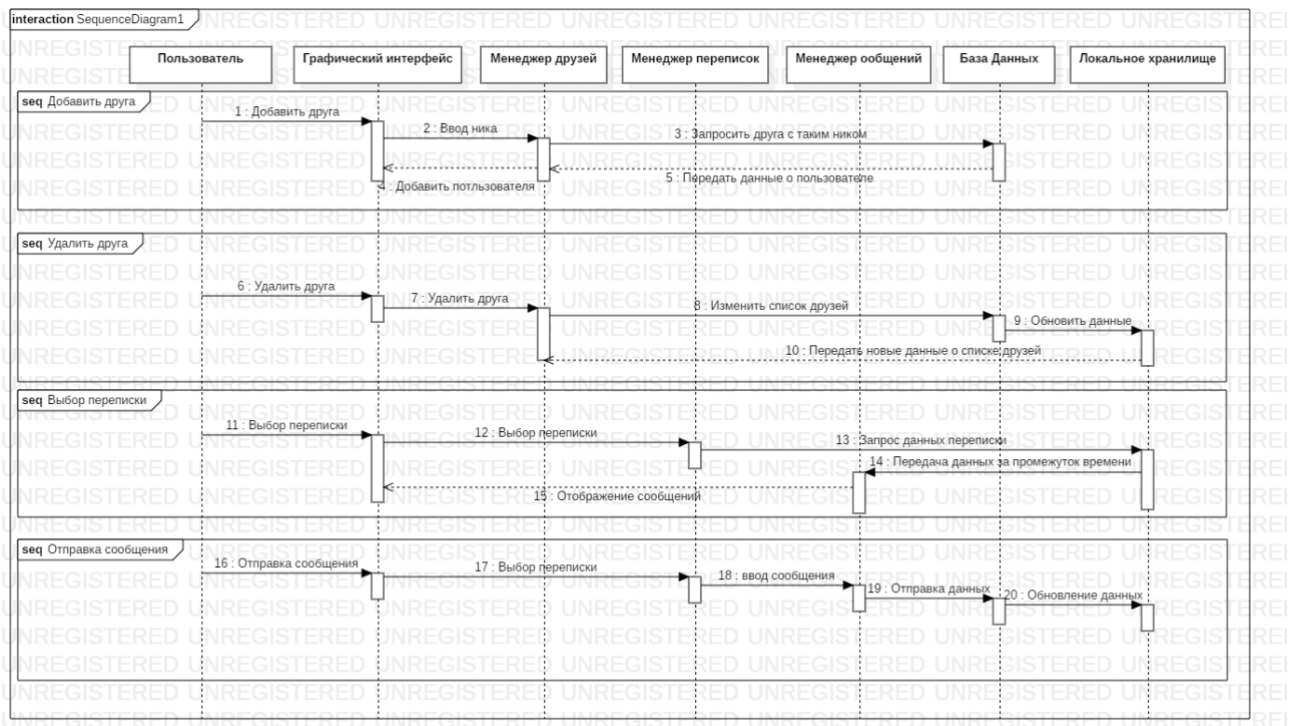


Рисунок 4 – Диаграмма последовательности

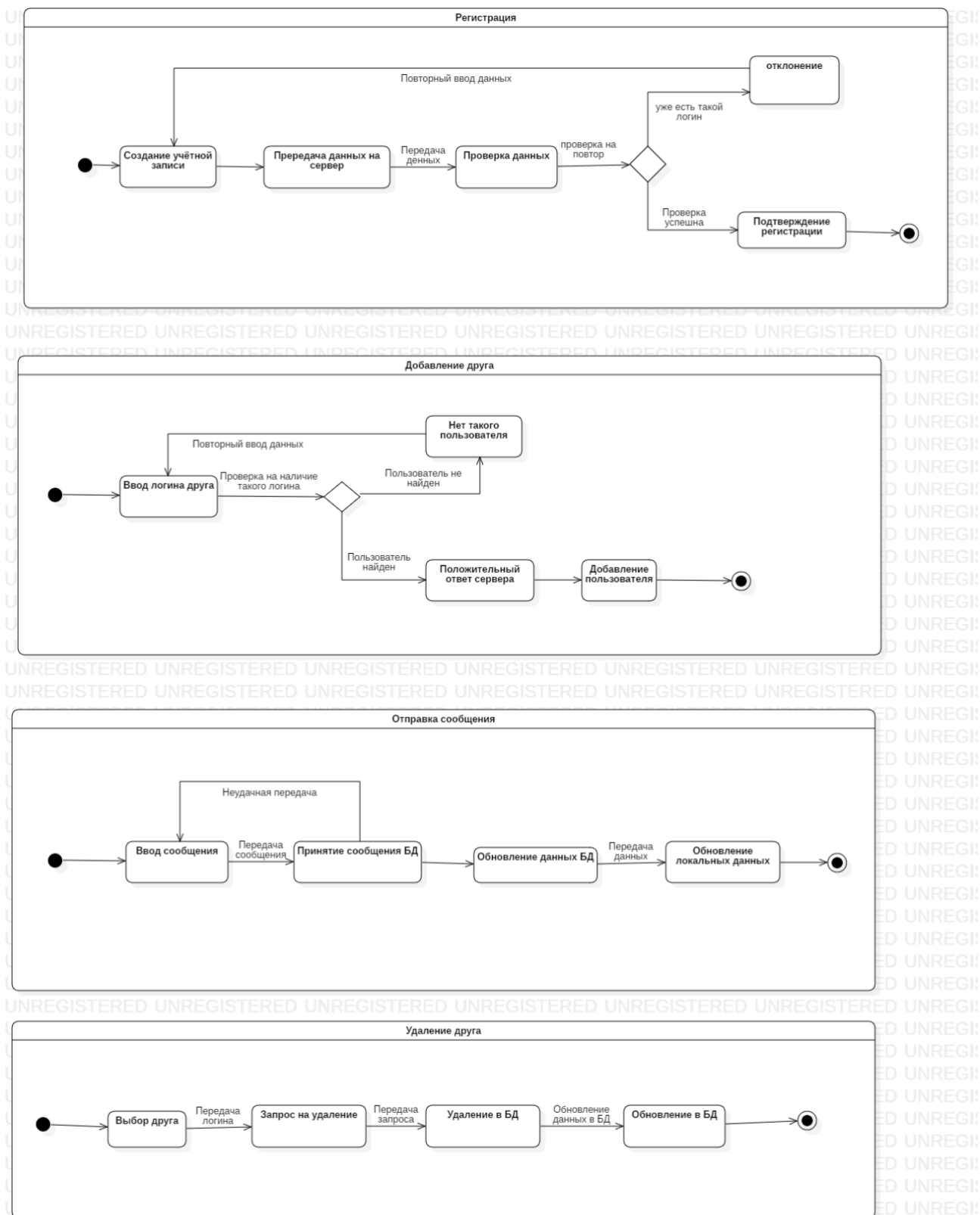


Рисунок 5 – Диаграмма коммуникации

На рисунке №, показана общая диаграмма коммуникации, на которой указывается отношение между объектами в приложении а на рисунке № изображена диаграмма последовательности на которой изображено

упорядоченное во времени взаимодействие объектов в приложении.

Для добавления друга пользователь обращается к графическому интерфейсу который вызывает менеджер друзей. Менеджер друзей ожидает ввода логина пользователя, если пользователь принял запрос в друзья то менеджер друзей отправляет данные на удалённую базу данных после чего показывает на графическом интерфейсе что пользователь был успешно добавлен.

Для удаления друга пользователь обращается к графическому интерфейсу который передает менеджеру друзей команду на удаление. Менеджер друзей делает запрос на удаленную базу данных об удалении файлов в таблице друзей. Удаленная база данных обновляются затем передавая данные приходит на локальную базу данных. Графический интерфейс обновляется.

Для выбора переписки пользователь обращается к графическому интерфейсу который передает менеджеру переписок команду на вывод последних сообщений. Менеджер переписок обращается к локальной базе данных с запросом на передачу списка последних сообщений. Локальная база данных ищет у себя сообщения если таковых не имеется то она делает запрос на удаленную базу данных. Удаленная база данных ищет у себя сообщения если она ничего не находит она передает графическому интерфейсу что история переписки с выбранным человеком нет. Графический интерфейс показывает пользователю что переписка не была начата и просит начать ее.

Для отправки сообщения пользователь обращается к графическому интерфейсу который передает менеджеру переписок. Менеджеру переписок вызывает менеджер сообщений и передает введенное сообщение. Менеджер сообщений отправляет сообщения на удаленную базу данных. Удаленная база данных обновляет данные о переписке и обновляет локальную базу данных.

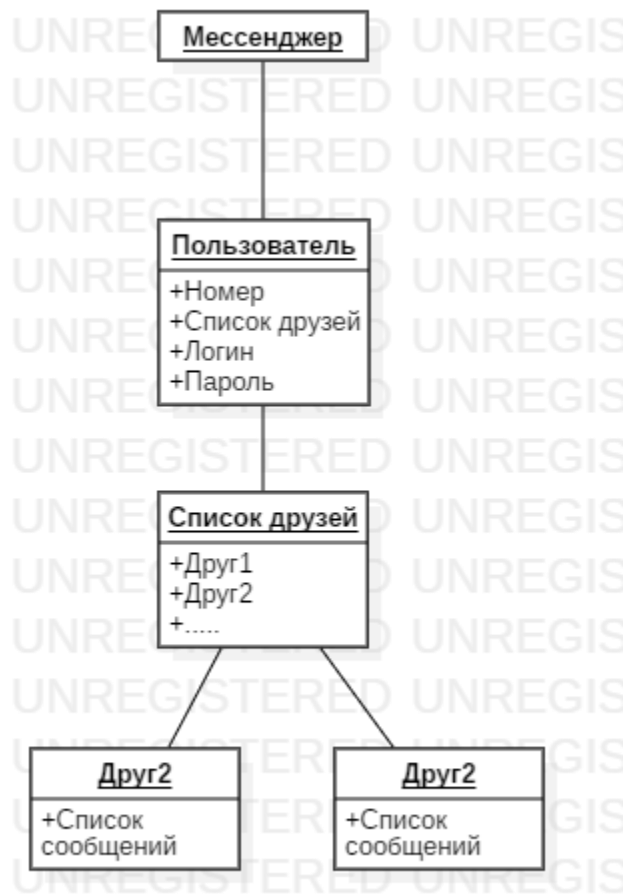


Рисунок 6 – Диаграмма объектов



Рисунок 7 – Диаграмма развертывания

## Анализ средств реализации

В качестве средств реализации приложения были выбраны следующие технологии:

- Android SDK – средство разработки мобильных приложений для операционной системы Android. Чертой, отличающей от других средств разработки, является наличие широких функциональных возможностей, позволяющих запускать тестирование и отладку исходных кодов, оценивать работу приложения в режиме совместимости с различными версиями ОС Android.
- В качестве СУБД для обмена сообщениями и возможности хранения данных пользователей была выбрана блочная СУБД FireBase, позволяющая разработчикам приложений хранить и синхронизировать данные между несколькими клиентами. Поддержаны особенности

интеграции с приложениями под операционные системы Android, реализовано API для приложений на java.

- В качестве основного языка разработки использовался язык java, так как имеет весь необходимый для работы набор инструментов.

Приложение разрабатывается под версии Android 4.1 и выше, по причине того, что больше 99% пользователей работают на устройствах под этой системой, а оставшийся 1% на устройствах с уже не поддерживаемой версией Android.

## **4 Реализация**

### **4.1 Сущности**

- Клиент
- Сервер

### **4.2 Серверная часть приложения**

Серверная часть основана на FireBase. Firebase является удобной в исполнении, реализации и эксплуатации. Firebase служит базой данных, которая изменяется в реальном времени и хранит данные в JSON. Любые изменения в базе данных тут же синхронизируются между всеми клиентами, или девайсами, которые используют одну и ту же базу данных. Другими словами, обновление в Firebase происходят мгновенно.

Вместе с хранилищем, Firebase также предоставляет пользовательскую аутентификацию, и поэтому все данные передаются через защищенное соединение SSL. У Firebase есть SDK для Android, IOS и JavaScript. Все платформы могут использовать одну базу данных.



### **4.3 Графический интерфейс**

Весь графический интерфейс был реализован в Android studio с использованием XML. Графический интерфейс приложения можно разделить на два основных окна. Главный экран или экран диалогов и экран переписки. Интерфейс создавался на основе аналогов, он простой для понимания и удобный в эксплуатации. Минималистичность интерфейса позволяет оставаться ему простым и функциональным.

### 4.3.1 Экран диалогов

Главный экран или экран диалогов. На нём можно выбрать и войти в диалог, создать новый диалог или выйти из аккаунта.

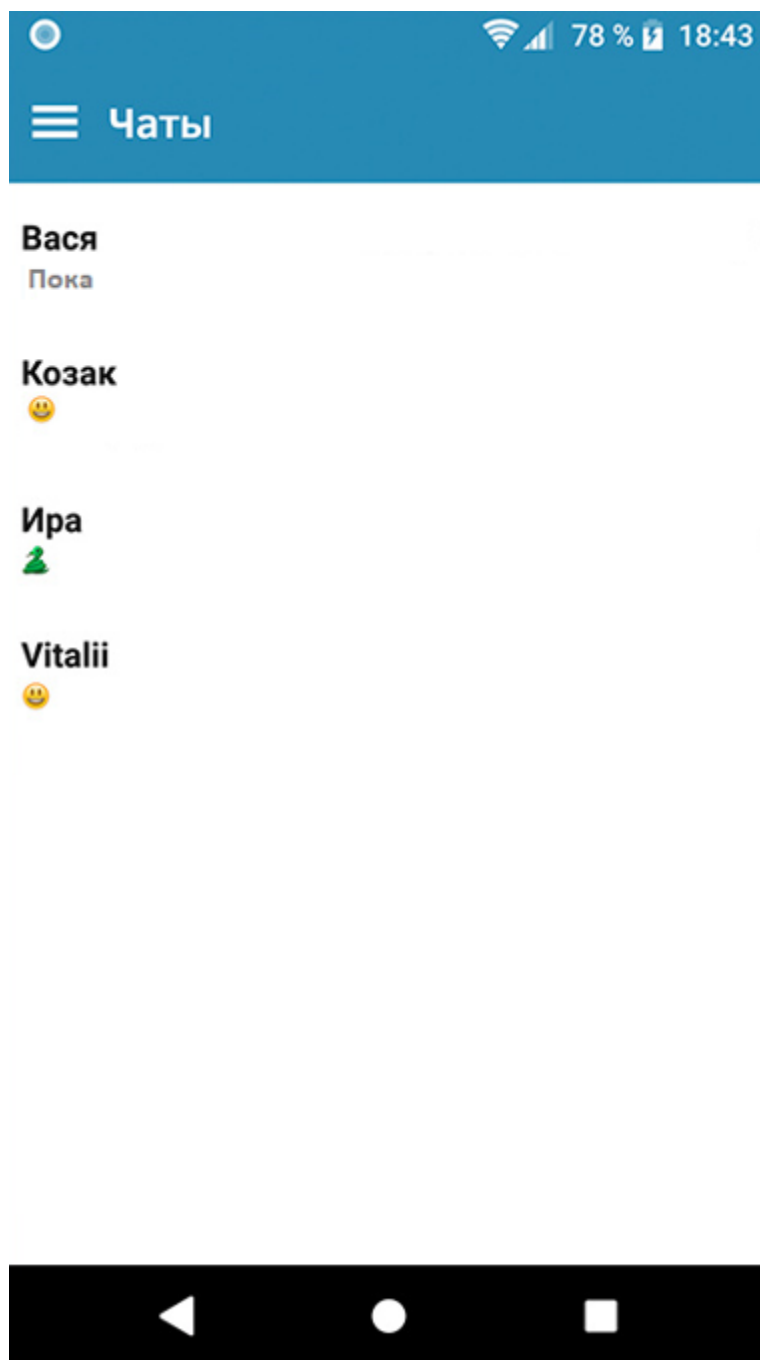


Рисунок 8 – Экран диалогов.

### 4.3.2 Экран переписки

На экране переписки имеется значок выхода на главный экран с диалогами, возможность просмотра сообщений, поле ввода сообщений и кнопка отправки.



Рисунок 9 – Экран переписки.

При нажатии на сообщение они подсвечиваются, что свидетельствует об их выделении, и в верхней части экрана появляется кнопка, в виде мусорной корзины, по нажатию на которую можно удалить сообщения.



Рисунок 10 – Выделение и удаление сообщений.

## 5 Тестирование

После реализации всех задач, был проведен запланированный набор тестов. Он включает 3 вида тестирования:

- Дымовое тестирование
- UI тесты
- Юзабилити тесты