

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Мессенджер MeSI

Курсовой проект

09.03.02 Программирования и информационных технологий

Допущен к защите

Обучающийся _____ Лепехин Д. А., 3 курс, д/о

Обучающийся _____ Кожевников Н. А., 3 курс, д/о

Руководитель _____ В.С. Тарасов

Воронеж 2020

Введение

21 век обусловлен быстрым темпом роста информационных технологий, а с появлением новых технологий, появляется много возможностей, в том числе и общение друг-с-другом, так и появились месенджеры, но их появление было за долго до этого, в 1965 году, когда сотрудник Массачусетский технологический институт (MIT) написали программу mail в операционной системе CTSS, в наше время это развилось в большой сервис обмена текстовых, голосовых сообщениями и медиа контента.

Так что же такое Мессенджер? Мессенджер - это система для мгновенного обмена сообщениями, с возможностью отправлять текст, голос, изображение или видео в реальном времени, через всемирную сеть интернет.

Данная курсовая работа представляет собой разработку приложения для коммуникации между людьми в котором они смогут обмениваться информацией и медиа контентом.

1. Постановка задачи

Цель курсовой работы: разработать веб приложение, которое отвечает следующим требованиям:

1. Стабильная работа в современных браузерах
2. Создание интуитивного пользовательского интерфейса
3. Возможность выполнения основных задач приложения:
 - 3.1. Регистрация пользователя
 - 3.2. Авторизация пользователя
 - 3.3. Поиск пользователей по e-mail, номеру телефона и индивидуального id пользователя
 - 3.4. Добавление пользователей в список контактов

3.5. Добавление пользователей в черный список

3.6. Обмен сообщениями между пользователями

3.7. Просмотр диалогов

3.8. Смена пароля

3.9. Редактирование профиля

Основную функциональность разрабатываемого приложения отражает диаграмма прецедентов, изображенная на рисунке 1.



Рисунок 1 - Диаграмма прецедентов

Финальная часть проекта представляет собой полностью функционирующее веб-приложение, соответствующее требованиям, написанным выше.

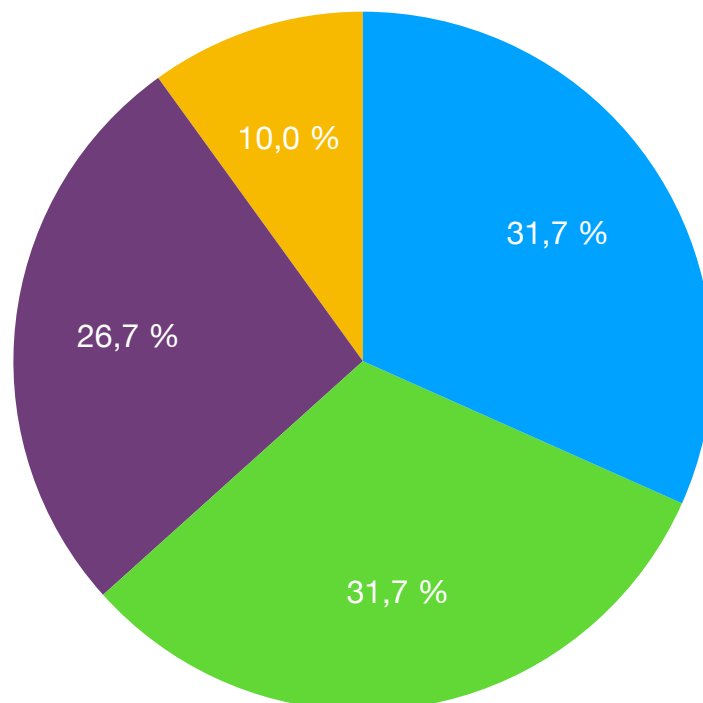
2. Анализ

2.1. Анализ предметной области

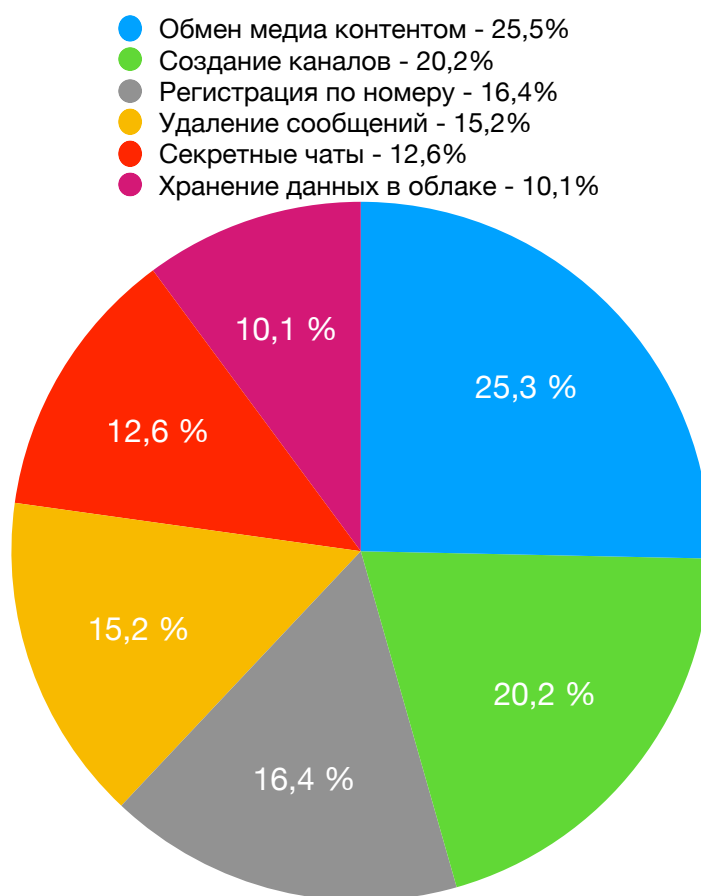
Проведя опрос среди студентов, мы получили статистику часто используемых мессенджеров, плюсы и минусы с которыми они сталкивались. Вся статистика представлена в обобщенном виде на диаграммах ниже:

1) Какими мессенджерами пользуются студенты:

● Telegram ● Whats' App ● Viber ● Другие



2) Причины использования менеджера



3) Что не нравится в используемом месенджере

1. Блокировка в России
2. Дизайн
3. Перегруженный интерфейс
4. Большое количество используемых ресурсов телефона
5. Качество контента

2.2 Сравнение с аналогами

Почти каждая крупная компания заинтересована в том, чтобы создать свой месенджер, в наше время их стало довольно много.

Рассмотрим наиболее крупные из них:

Telegram - кроссплатформенный месенджер, позволяющий обмениваться сообщениями и медиафайлами многих форматов. Минусом данного приложения является запрет и не стабильная работа в Российской Федерации, требующей в некоторых случаях использование VPN(**Virtual Private Network** - виртуальная частная сеть) или Проxy(промежуточный сервер, выполняющий роль посредника между пользователем и целевым сервером), данные методы подходят не всем пользователям, например обычным людям, чья область не соприкасается с информационными технологиями.

WhatsApp - это бесплатное приложение, которое предлагает простой, безопасный и надёжный обмен сообщениями и звонками, доступное на мобильных телефонах по всему миру. Основными недостатками данного приложения являются: отсутствие возможности хранить данные и переписки на сервере(icloud и google drive не в счет так как их пространство ограничено тарифом, в отличии от того же телеграмма), нельзя использовать на нескольких устройствах одновременно(привязка к одному устройству).

3. Анализ Задачи

3.1. Хранение данных пользователя и приложения

Данная задача представляет собой:

1. Хранение персональных данных(ФИО, номер телефона, о себе)
2. Хранение диалогов пользователя

3.1.1. Хранение персональных данных

Хранение персональных данных содержит в себе следующую информацию:

1. Имя, Фамилия
2. Номер телефона или e-mail
3. Информацию о себе
4. Никнейм

Выполняя данную задачу мы должны учесть некоторые аспекты

1. Сохранение конфиденциальности данных
2. Возможность изменения собственных данных
3. Просмотр своих данных

3.1.2. Хранение диалогов

В этом пункте рассмотрим хранение диалогов пользователя, содержащую следующую информацию:

1. История переписки
2. Вложения(медиа, файлы, ссылки)

При выполнении данной задачи мы учитываем следующие пункты

1. Сохранение от несанкционированного доступа к перепискам пользователей
2. Хранение данных в облаке и на устройстве пользователя

4. Анализ средств реализации

В качестве средств реализации были выбраны следующие технологии:

1. Python - высокоуровневый язык программирования общего назначения ориентированный на повышение производительности разработчика и читаемости кода.

2. В качестве фреймворка для языка описанного выше, был выбран Django - Свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC.

3. За облачную основу приложения был взят Google Cloud - предоставляемый компанией Google набор облачных служб, которые выполняются на той же самой инфраструктуре, которую Google использует для своих продуктов, предназначенных для конечных потребителей, таких как YouTube и Google Search.

4. В качестве СУБД вышла выбрана MySQL, являющаяся свободной и простой для реализации системы управления базами данных.

5. Для реализации каркаса и стиля приложения были выбраны такие языки как:

5.1 HTML 5 - язык для структурирования и предоставления данных

5.2 CSS 3 - каскадная таблица стилей, преимущественно используемая как средство описания, оформления внешнего вида веб-страниц

5.3 JS - язык программирования используемый как встраиваемый для программного доступа к объектам приложения, а так же поддерживаемый объектно-ориентированный стиль

6. В качестве реализации внешнего вида веб-приложения был выбран фреймворк Bootstrap 3 - набор инструментов для создания сайтов и веб-приложений

5. Графическое описание работы системы

В качестве описания работы системы был использован язык графического описания UML. В данном разделе представлены диаграммы и описания. Описание того, что может делать авторизованный пользователь и неавторизованный пользователь

1. Диаграмма состояний

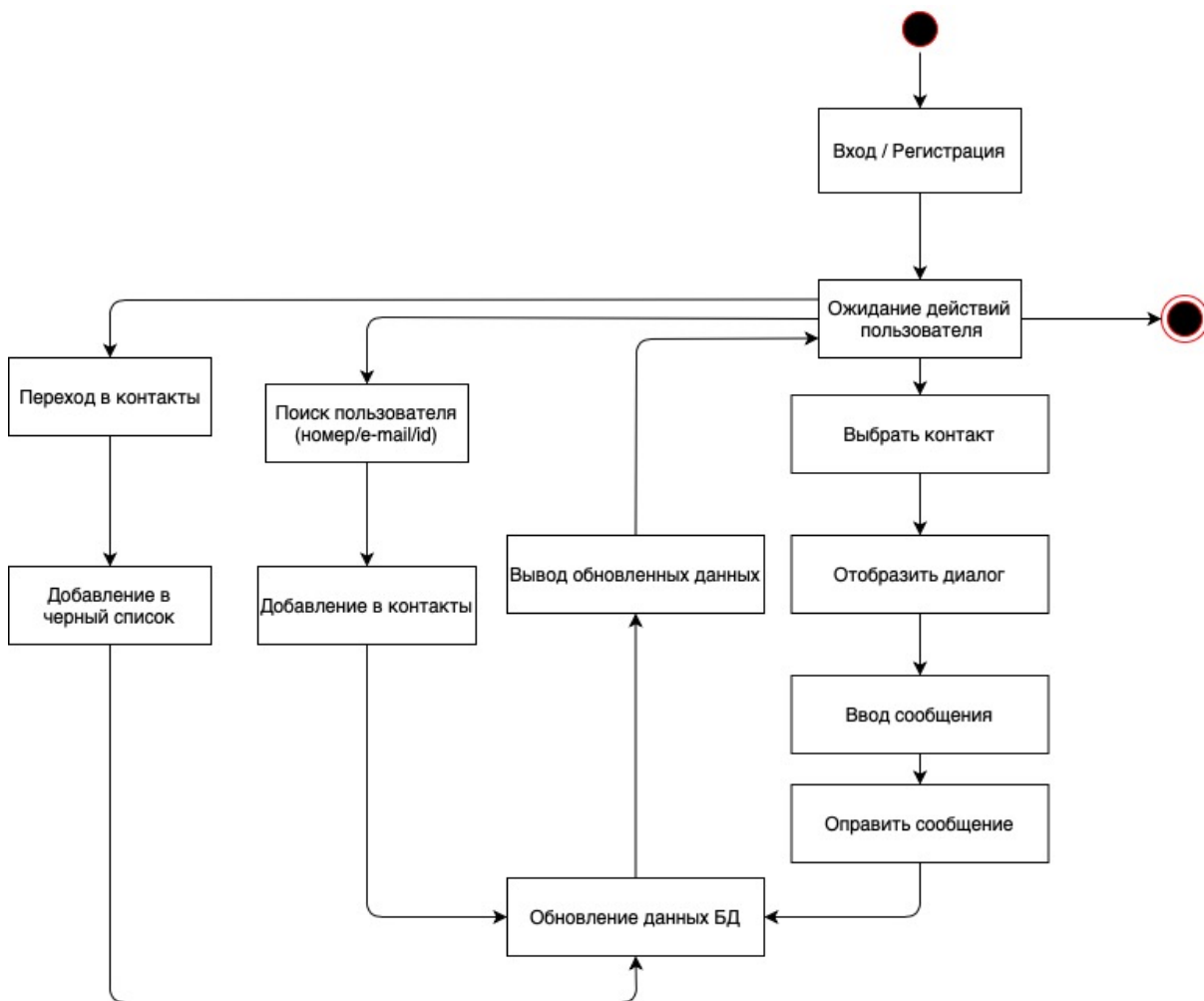


Рисунок №2 Диаграмма Состояний

Диаграмма состояний на рисунке №2, отображает все возможные состояния системы, при переходе в веб-приложение пользователь попадает на страницу авторизации/регистрации, перед входом в систему у пользователя есть две цепочки событий:

1. Авторизация

2. Регистрация

При входе в систему он должен ввести логин(номер телефона/e-mail) и пароль или зарегистрироваться путем ввода логина(номер телефона или e-mail) и придумать пароль. После чего пользователь попадает на главную страницу с переписками. После система переходит в состояние ожидания действия пользователя и тут возможны 4 основные цепочки

1. Поиск пользователя по id, e-mail или номеру телефона

2. Добавление в контакты

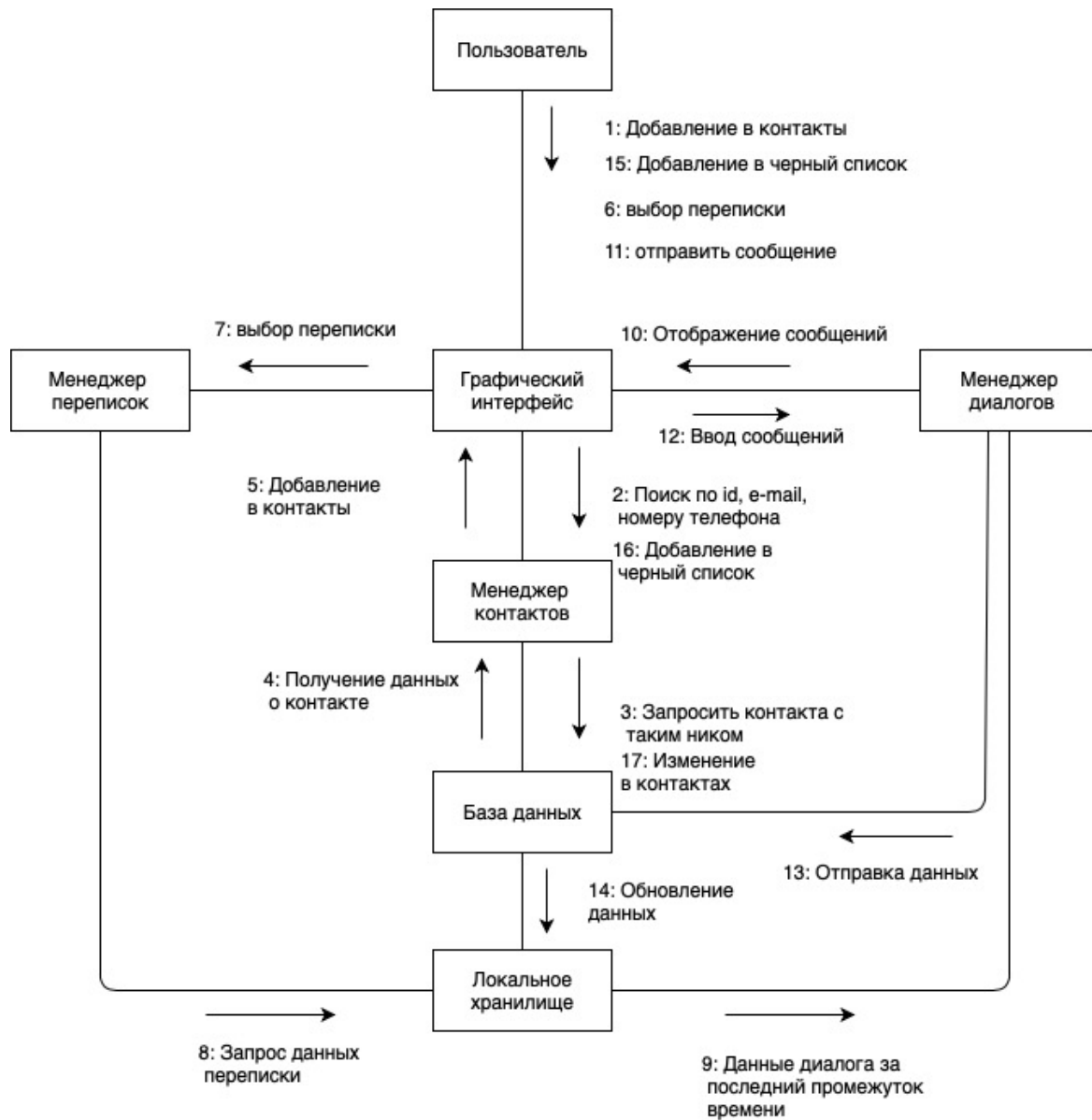
3. Открыть переписку

4. Внесение в черный список

Когда пользователь выбирает состояние добавление в контакты, то после происходит обновление базы данных и после он отображается в списке контактов.

При выборе состояния выбора контакта отображается переписка с пользователем, если сообщений нет, система предложит начать переписку с выбранным контактом. 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. При переходе в контакты пользователь может выбрать конкретного контакта и добавить его в черный список

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



2. Диаграмма активности

Рисунок 3 Диаграмма активности

На рисунке №3 , показана диаграмма активности, на которой отображаются возможные действия, которые описаны на диаграмме состояний(Рисунок 2).

Для добавления нового контакта пользователь А обращается к графическому интерфейсу приложения, который в свою очередь просит менеджер контактов добавить соответствующего пользователя Б с список контактов пользователя А после чего происходит обновление удаленной базы данных и добавление данных в локальную базу данных и обновление списка контактов.

Для начала переписки пользователь обращается к графическому интерфейсу веб-приложения, который в свою очередь просит менеджер переписок обратиться к удаленной базе данных, которая возвращает данные менеджеру сообщений данные за последний день. Если сообщений в удаленной базе нет он вернет пустой список и графический интерфейс предложит ввести сообщение если сообщение было введено, менеджер сообщений отправит его на удаленную базу которая в свою очередь обновит локальную базу данных и обновит список сообщений у обоих пользователей.

Для добавления в черный список пользователь обращается к графическому интерфейсу , который в свою очередь просит менеджер контактов обновить список контактов на удаленной базе данных. А потом изменит список контактов на локальной базе данных. Сообщения переписки будут храниться ещё некоторое время.

3. Диаграмма последовательности

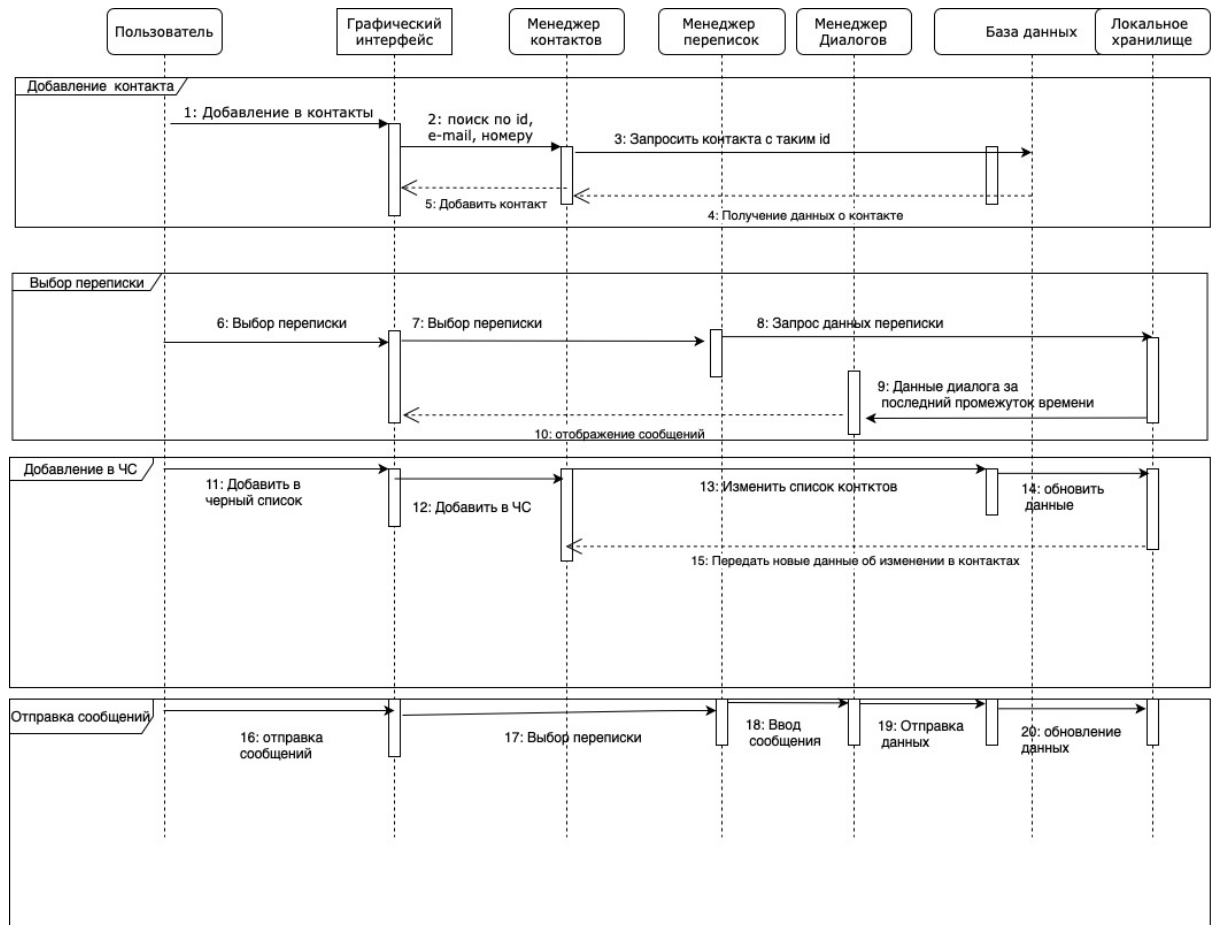


Рисунок № 4