МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность 1-40 01 01 «Программное обеспечение информационных технологий»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Базы данных»

Тема: «Реализация базы данных для веб-приложения мессенджер с применением технологии мультимедийных типов данных»

**Исполнитель**

студент 3 курса 5 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. В. Гайков

подпись, дата

**Руководитель**

ассистент \_\_\_\_\_\_\_\_\_\_\_\_ О. А. Нистюк

(должность, уч. звание) (подпись, дата)

Допущен(а) к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ О. А. Нистюк

подпись дата инициалы и фамилия

Минск 2023

**Содержание**

[Введение 5](#_Toc135051384)

[1 Анализ требований к программному средству 6](#_Toc135051385)

[1.1 Аналитический обзор аналогов 6](#_Toc135051386)

[1.1.1 Аналог Spotify 6](#_Toc135051387)

[1.1.2 Аналог Apple Music 7](#_Toc135051388)

[1.2 Разработка функциональных требований, определение вариантов использования 8](#_Toc135051391)

[1.3 Вывод 10](#_Toc135051393)

[2 Разработка архитектуры проекта 11](#_Toc135051394)

[2.1 Обобщенная структура управлением приложения 11](#_Toc135051395)

[2.2 Диаграммы UML, взаимосвязь всех компонентов. 11](#_Toc135051396)

[2.3 Описание информационных объектов 12](#_Toc135051397)

[2.4 Вывод 13](#_Toc135051398)

[3 Разработка модели базы данных 14](#_Toc135051399)

[3.1 Создание необходимых объектов 14](#_Toc135051400)

[3.1.1 Представления базы данных 15](#_Toc135051401)

[3.1.2 Индексы базы данных 15](#_Toc135051402)

[3.1.3 Триггеры базы данных 16](#_Toc135051403)

[3.2 Описание используемой технологии 17](#_Toc135051404)

[3.3 Вывод 18](#_Toc135051405)

[4 Установка, настройка и использование PosgtreSQL 14.5 20](#_Toc135051406)

[4.1 Установка PostgreSQL 20](#_Toc135051407)

[4.2 Создание таблиц 20](#_Toc135051408)

[4.3 Создание ролей для разграничения 21](#_Toc135051409)

[4.4 Создание пакетов процедур для базы данных 23](#_Toc135051410)

[4.4.1 Выборка данных из таблиц 24](#_Toc135051411)

[4.4.2 Выборка данных по поисковому запросу 25](#_Toc135051412)

[4.4.3 Заполнение таблиц 100 000 строк 27](#_Toc135051413)

[4.4.4 Добавление данных в таблицы 27](#_Toc135051414)

[4.4.5 Удаление данных в таблицы 29](#_Toc135051415)

[4.4.6 Изменение данных в таблицы 29](#_Toc135051416)

[4.4.7 Дополнительные функции 30](#_Toc135051417)

[4.5 Описание процедур экспорта и импорта 32](#_Toc135051418)

[4.6 Вывод 35](#_Toc135051419)

[5 Тестирование 36](#_Toc135051420)

[5.1 Тестирование производительности базы данных 36](#_Toc135051421)

[5.2 Вывод 37](#_Toc135051422)

[6 Руководство по использованию программного средства 38](#_Toc135051423)

[6.1 Руководство пользователя 38](#_Toc135051424)

[6.2 Установка приложения 43](#_Toc135051425)

[6.3 Вывод 43](#_Toc135051426)

[Заключение 44](#_Toc135051427)

[Список литературных источников 45](#_Toc135051428)

[Приложение А 46](#_Toc135051429)



# Введение

Цель данной работы заключается в создании реляционной базы данных для веб-приложения мессенджер, которое позволит пользователям обмениваться сообщениями, голосовыми сообщениями и сообщениями в виде стикеров.

База данных - это организованное собрание данных, которое обычно хранится в электронном виде в компьютерной системе. БД используются для хранения, организации и управления большим объемом структурированных и неструктурированных данных. Реляционная база данных является наиболее распространенной формой организации данных, в которой данные представлены в виде таблиц, состоящих из строк и столбцов, где каждый столбец представляет атрибут, а каждая строка представляет кортеж или запись. В данной работе для управления базой данных была выбрана СУБД Postgres SQL, поскольку эта система обладает высокой надежностью и производительностью, что позволяет обеспечить эффективное хранение, обработку и управление музыкальными данными.

Также необходимо разработать приложение для демонстрации функциональности базы данных и взаимодействия с ней. Приложение было реализовано с использованием фреймворка NestJS для платформы Node.js и фреймворка React с TypeScript.

Для гарантированной безопасности пользователей приложения в курсовой работе применяется метод шифрования паролей перед их сохранением в базу данных. Также для обеспечения функциональности приложения используются мультимедийные форматы данных при сохранении аудио и картинок.

Основные требования к приложению:

* реализация ролей администратора и пользователя;
* отправка, редактирование и удаление сообщений;
* добавление в друзья, удаление из друзей;
* возможность подписаться на другого или отписать от него;
* взаимодействие с базой данных при помощи хранимых процедур и функций.

В пояснительной записке содержится информация о сопоставимых продуктах, структуре и реализации проекта, а также инструкции по использованию приложения.

1. Анализ требований к программному средству

## Аналитический обзор аналогов

Были проанализированы цели и задачи, поставленные в данном курсовом проекте, а также рассмотрены аналогичные примеры их решений. На основании анализа всех достоинств и недостатков данных альтернативных решений были сформулированы требования к данному программному средству.

Первый аналог — WhatsApp.

WhatsApp обладает широким спектром функций, включая отправку текстовых сообщений, голосовых сообщений, видеозвонки, групповые чаты, обмен файлами и стикерами, а также возможность создания и управления каналами. Этот мессенджер также позволяет пользователям отправлять местоположение и документы.

WhatsApp имеет простой и удобный интерфейс, который легко использовать. Дизайн этого мессенджера состоит из основных функций в левой части экрана и списка чатов в правой части. WhatsApp также предлагает различные темы, которые пользователи могут выбрать, чтобы изменить внешний вид мессенджера.

Дизайн WhatsApp - довольно минималистичный и простой в использовании. Цветовая схема состоит в основном из белого и зеленого, что создает ощущение легкости и чистоты.

Блок сообщения сделан закругленным. Текст для имени пользователя подсвечивается другим цветом. Текст черного цвета на белом фоне. На данное сообщение довольно-таки приятно смотреть.

Интерфейс приложения представлен на рисунке 1.1.

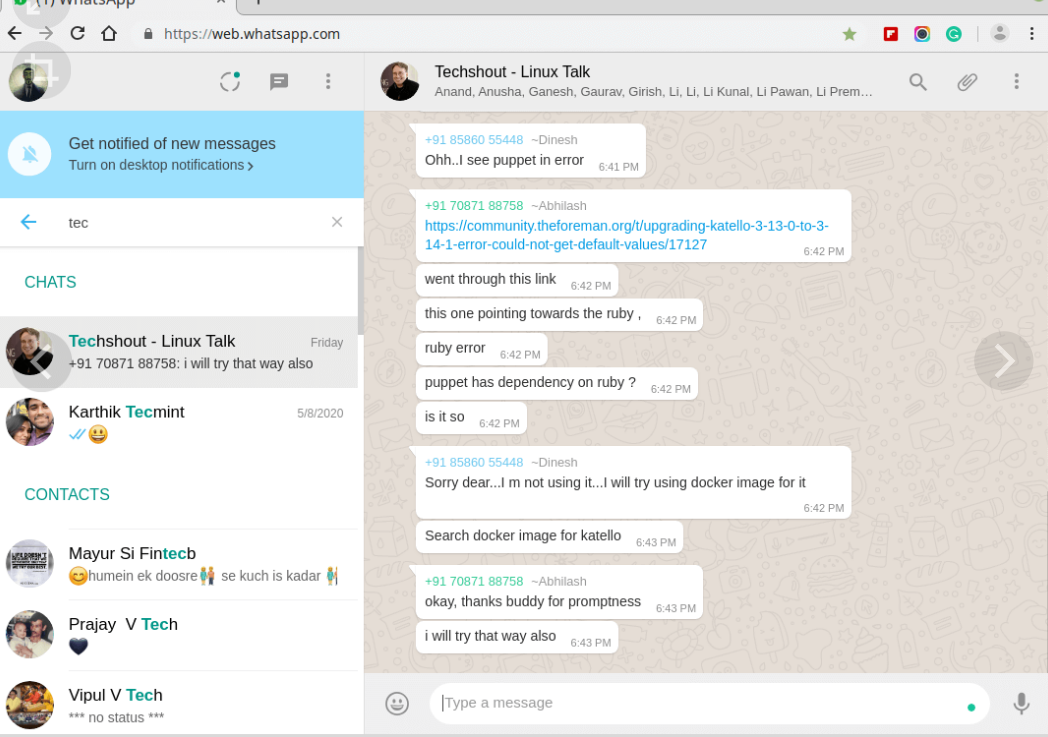


Рисунок 1.1 – Приложение «WhatsApp»

Второй аналог — это приложение Viber.

Viber позволяет отправлять текстовые сообщения, стикеры, фотографии, видео и аудиофайлы. Вы также можете создавать групповые чаты для общения с несколькими людьми одновременно. Viber позволяет совершать бесплатные голосовые звонки и видеозвонки с другими пользователями Viber по всему миру. Для этого необходимо, чтобы оба пользователя были подключены к интернету. имеет светлый интерфейс с фиолетовыми элементами дизайна.

Главный экран отображает список ваших чатов, а также кнопки для совершения голосовых и видеозвонков. В верхней части экрана расположены кнопки для доступа к контактам, магазину стикеров и настройкам. В целом, интерфейс Viber интуитивно понятен и удобен в использовании.

Дизайн Viber представляет собой современный и лаконичный интерфейс с использованием ярких цветов и плоских иконок. Он имеет минималистичный дизайн, который делает приложение легким и простым в использовании.

Страница пользователя имеет картинку человека, на которого мы зашли. Кнопки в нем подсвечены и выделяются на общем фоне. Сразу видно, куда жать.

Интерфейс приложения представлен на рисунке 1.2.

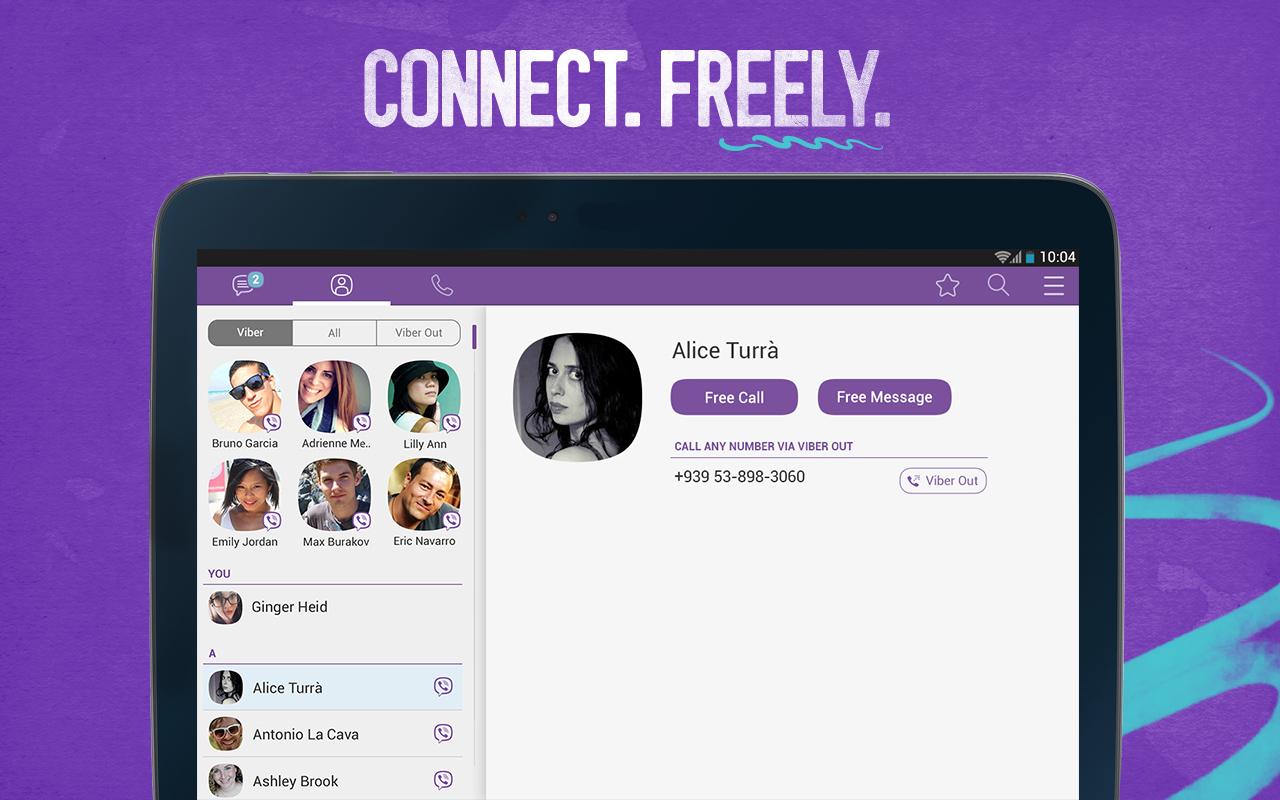


Рисунок 1.2 – Приложение «Viber»

Третий аналог — это Telegram.

Telegram обладает широким спектром функций, включая отправку текстовых сообщений, голосовых сообщений, видеозвонки, групповые чаты, демонстрация экрана, обмен файлами и стикерами, а также возможность создания каналов и ботов. Этот мессенджер также предлагает возможность создания секретных чатов с функцией автоматического удаления сообщений.

Telegram имеет простой и удобный интерфейс, который легко использовать. Дизайн этого мессенджера состоит из основных функций в нижней части экрана и списка чатов в верхней части. Telegram также предлагает различные темы, которые пользователи могут выбрать, чтобы изменить внешний вид мессенджера.

Дизайн Telegram отличается от других мессенджеров своей минималистичностью и простотой в использовании. Он имеет темно-синий цветовой фон, который выделяет контент на экране. В верхней части экрана расположено главное меню, которое позволяет быстро переключаться между разделами мессенджера. В центре экрана находится список чатов и диалогов. Каждый чат представлен в виде миниатюрного значка с изображением фото профиля пользователя или группы, а также отображается последнее сообщение. При нажатии на любой из чатов, открывается окно переписки, которое также имеет темно-синий фон и белый шрифт. Дизайн приятный и легко пользоваться им.

Интерфейс приложения представлен на рисунке 1.3.

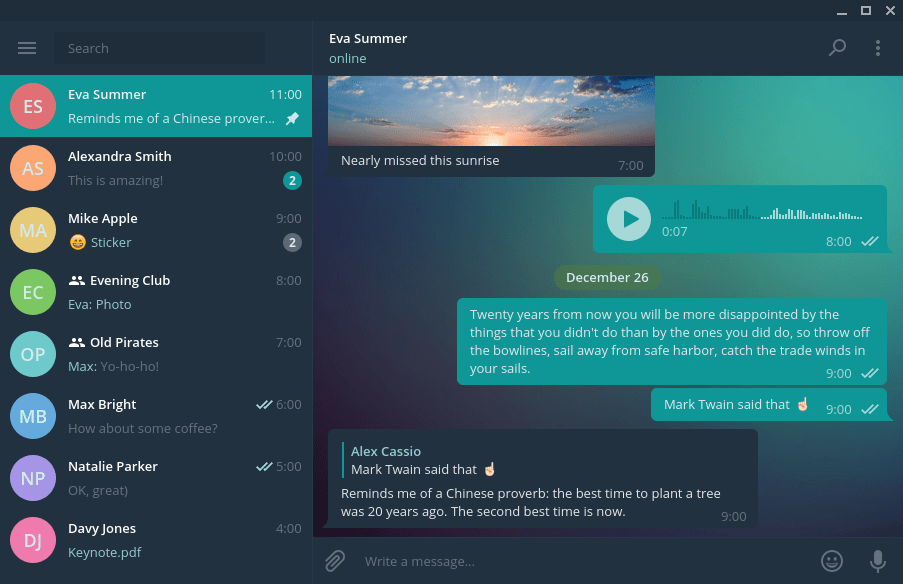


Рисунок 1.3 – Приложение «Telegram»

Анализируя аналоги WhatsApp, Viber и Telegram, можно сделать вывод, что все три приложения предлагают широкий спектр функций, включая обмен сообщениями, голосовые и видеозвонки, групповые чаты и возможность отправки файлов. Каждое из приложений имеет свою уникальную цветовую схему и дизайн интерфейса, но общий тренд — простота и удобство использования. Все они предлагают настраиваемый интерфейс с возможностью выбора тем оформления

## 1.2 Разработка функциональных требований, определение вариантов использования

Функциональные требования базы данных определяют, как база данных должна обрабатывать данные и предоставлять пользовательскому интерфейсу необходимую функциональность. Это может включать в себя описание того, как данные должны храниться и организовываться, как происходит поиск и выборка данных, каким образом обновляются данные и какие механизмы используются для защиты данных. Кроме того, функциональные требования могут определять интеграцию базы данных с другими системами и программами. Например, для мессенджера, как будут хранится сообщения, как будут храниться аудиосообщения, как будут храниться стикеры и фотографии пользователей.

Помимо функциональных требований, важно также определить роли пользователей и их варианты использования системы. Варианты использования описывают, как пользователи будут взаимодействовать с системой в зависимости от своих ролей. Это помогает определить, какие функции должны быть доступны для каждой роли, какие данные должны быть доступны для каждой роли, а также как должна быть организована навигация в системе. Варианты использования обычно представляются в виде UML диаграмм, которые позволяют наглядно отобразить взаимодействие между пользователями и системой.

Роли пользователя — это набор прав, которые пользователь может получить в системе. В зависимости от роли пользователя, он может иметь доступ к различным функциям системы. В данном проекте роли пользователей будут следующими:

* Visitor.
* APP\_USER.
* APP\_ADMIN.

На основе предоставленного списка ролей необходимо построить варианты использование. Варианты использование изображена на рисунке 1.4.

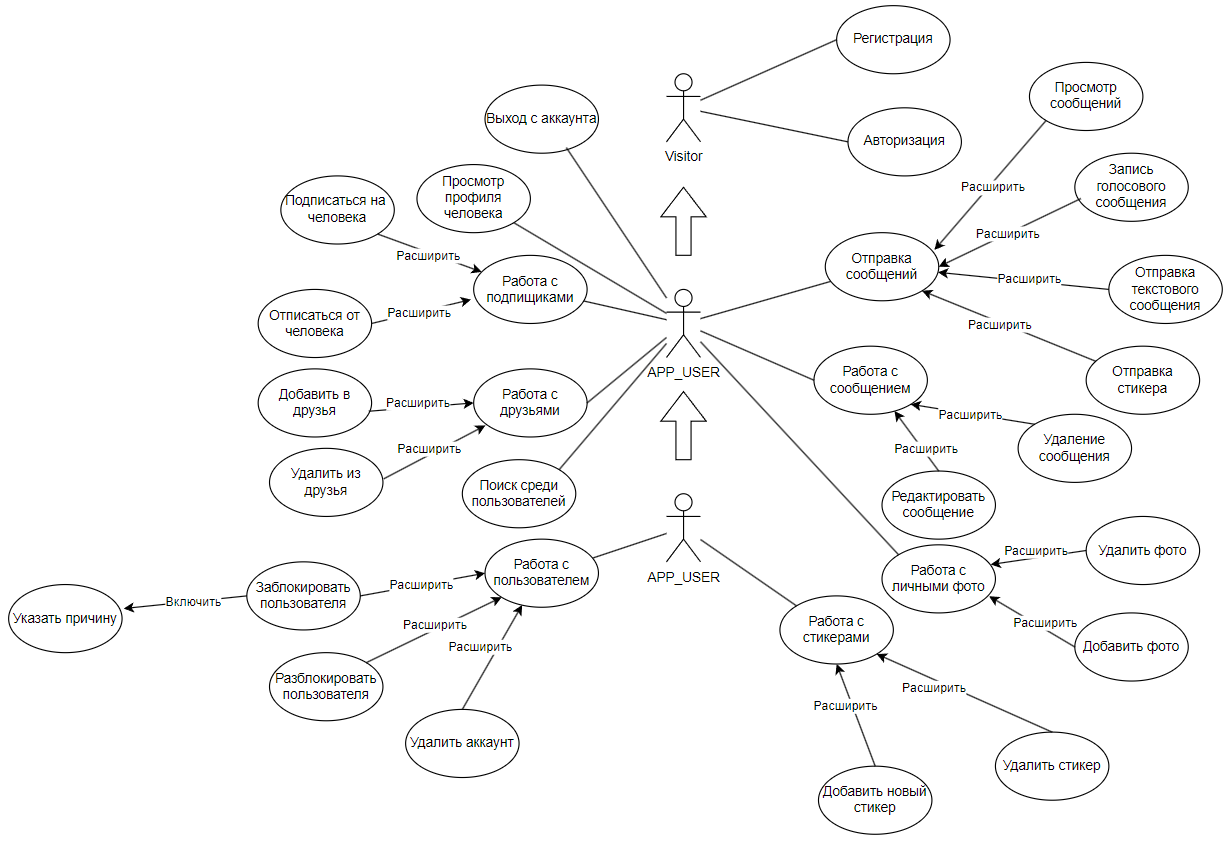


Рисунок 1.4 – UML диаграмма вариантов использования

В начале работы с приложением пользователь является посетителем. Ему будут доступны лишь две функции: авторизация и регистрация. После регистрации пользователь становится пользователем APP\_USER.

Роль APP\_USER дает уже функцию отправлять разные виды сообщений, просматривать все свои сообщения, подписываться на кого-либо или отписаться от кого кого-нибудь, добавлять человека в друзья и удалять людей из друзей, добавлять, удалять фото, искать пользователей, редактировать или удалять свои сообщения.

Роль APP\_ADMIN может также просматривать чужие аккаунты, но цель найти недопустимую информацию. Если такая есть, он может заблокировать пользователя, указав при этом причину, или же вовсе удалить данный аккаунт. Также администратору доступно удаление или добавление стикеров.

Приложение обладает тремя ролями, на которые разложены все основные функции для корректного использования.

## 1.3 Вывод

Итого, был проведен аналитический обзор аналогов мессенджеров и сервисов, которые уже существуют на рынке. Этот обзор позволил определить основные характеристики и функциональные возможности, которые необходимо предусмотреть в разрабатываемой системе. Также были определены функциональные требования базы данных, а также роли пользователей и варианты использования системы в зависимости от этих ролей. Была разработана UML-диаграмма, на которой отображены основные функции, которые доступны для каждой из ролей пользователей.