МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ ДНР

ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

КАФЕДРА ПРОГРАММНОЙ ИНЖЕНЕРИИ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к проекту по курсу

«Практика»

Выполнили:

ст. гр. ПИ–18б

Моргунов А.Г.

ст. гр. ПИ-18а

Данильчук К.М.

Руководитель:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ДОНЕЦК – 2021

# РЕФЕРАТ

Пояснительная записка содержит: \_ страниц, \_ рисунков, \_таблиц, \_источников, \_приложений.

Цель проектирования: выполнить обоснованный проект телеграмм-бота, который удовлетворяет ряду специальных требований и реализует функции, описанные в техническом задании (приложение А).

Для реализации цели курсового проекта необходимо: выявить требования к пользовательскому интерфейсу бота и сравнить его с уже существующими аналогами, выбрать средства реализации, которые удовлетворяют требованиям будущей системы, спроектировать систему, позволяющую реализовать все функции, описанные в техническом задании, спроектировать и реализовать базу данных, которая хранит информацию о подписках (какую группу прослушивать, куда отсылать, по каким ключевым словам фильтровать и т.д.).

Результатом работы является программа, которая позволяет подписываться на обновления групп в телеграмме и пересылать новые сообщения в указанные в подписке группы.

ТЕЛЕГРАММ, ТЕЛЕГРАММ-БОТ, БАЗА ДАННЫХ, СЕРВЕР, НОВОСТЬ, ОБНОВЛЕНИЕ, ПЕРЕСЫЛКА, СООБЩЕНИЕ, ВЛОЖЕНИЕ

Table of Contents

[Ведение 5](#_Toc82438609)

[1 Постановка задачи 6](#_Toc82438610)

[1.1 Задача заказчика 6](#_Toc82438611)

[1.2 Идея команды разработчиков 6](#_Toc82438612)

[1.3 Экономическое обоснование 6](#_Toc82438613)

[2 Архитектура 6](#_Toc82438614)

[2.1 Сущности 6](#_Toc82438615)

[2.2 Диаграммы … 6](#_Toc82438616)

[3 Выбор средств 6](#_Toc82438617)

[3.1 Платформа и окружение 6](#_Toc82438618)

[3.2 Язык программирования 6](#_Toc82438619)

[3.3 Third-party 6](#_Toc82438620)

[3.4 СУБД 6](#_Toc82438621)

[4 Реализация 6](#_Toc82438622)

[4.1 Взаимодействие с telegram API 6](#_Toc82438623)

[4.2 Взаимодействие с Bot API 6](#_Toc82438624)

[4.3 Взаимодействие с PQXX 6](#_Toc82438625)

[4.4 Логирование 6](#_Toc82438626)

[5 Перспективы развития 6](#_Toc82438627)

[5.1 Идея 6](#_Toc82438628)

[5.2 Архитектура 7](#_Toc82438629)

[5.2.1 Сервер 7](#_Toc82438630)

[5.2.2 Сайт 7](#_Toc82438631)

[Выводы 7](#_Toc82438632)

[Литература 7](#_Toc82438633)

[Приложение А. Техническое задание 7](#_Toc82438634)

[Приложение Б. Экранные формы 7](#_Toc82438635)

[Приложение В. Статистика приложения 7](#_Toc82438636)

[Приложение Г. Руководство пользователя 7](#_Toc82438637)

[Приложение Е. Руководство администратора 7](#_Toc82438638)

[Приложение Ё. Код 7](#_Toc82438639)

# Ведение

Telegram — кроссплатформенная система мгновенного обмена сообщениями (мессенджер) с функциями VoIP, позволяющая обмениваться текстовыми, голосовыми и видеосообщениями, стикерами и фотографиями, файлами многих форматов. Клиентские приложения Telegram доступны для Android, iOS, Windows Phone, Windows, macOS и GNU/Linux. Количество ежемесячных активных пользователей сервиса по состоянию на январь 2021 года составляет около 500 млн человек. [1]

Телеграмм предоставляет API для создания ботов – специальных аккаунтов телеграмм, которые могут быть запрограммированы и выполнять определенные действия на заранее сформулированные команды или действия.

При выполнении проекта были проанализированы интерфейсы различных телеграмм-ботов и выбран наиболее понятный для пользователя, спроектирована структура базы данных, необходимая для хранении информации о пользователях и созданных ими подписках, распределены функции системы между ботом и сервером, а также был внедрен механизм подписки на использование услуг бота, который автоматически отключает пользователя от системы при отсутствии продления аренды.

# 1 Постановка задачи

## 1.1 Задача заказчика

## 1.2 Идея команды разработчиков

## 1.3 Экономическое обоснование

# 2 Архитектура

## 2.1 Сущности

В процессе анализа требований к проекту были выявлены следующие сущности:

* Телеграмм;
* Телеграмм-бот;
* База данных;
* Медиатор.

## 2.1.1 Телеграмм

Эта сущность отвечает за работу с информацией в телеграмме, а именно за просмотр обновлений каналов, на которые была осуществлена подписка, получение служебной информации из опубликованного сообщения и выделение из нее значимой части, пересылку сообщений из одного канала в другой, осуществляет поиск по ключевым словам в сообщениях сообщества.

## 2.1.2 Телеграмм-бот

Телеграмм-бот выступает пользовательским интерфейсом, то есть позволяет пользователю взаимодействовать с системой. Эта сущность интерпретирует команды, которые вводит пользователь и выполняет связанные с этой командой действия. Расширяет функциональность базового API бота путем предоставления возможности ведения цепочки сообщений для диалога с пользователем.

## 2.1.3 База данных

База данных – сущность, которая предназначена для хранения информации, которая может понадобится системе или администраторам во время работы. База данных хранит в себе: информацию о пользователях, оформленных ими подписках, сроке окончания аренды для каждого пользователя, а также информацию о наличии прав администратора каждого пользователя. Также система сохраняет историю запросов в удобном для просмотра виде.

## 2.1.4 Медиатор

Медиатор является связующим звеном системы. Он принимает сообщения, которые приходят из сущности телеграмма, берет из базы данных связи, которые относятся к сообществу из которого пришло сообщение, фильтрует его по ключевым словам, определенным для каждой связи и рассылает ко всем адресатам, которые подходят по связям и ключевым словам. Телеграмм-бот является интерфейсом к данной сущности и позволяет менять конфигурацию связей в базе данных, регистрировать новых пользователей и администраторов, продлевать аренду. Преобразует список найденных новостей в сообщение с постраничным просмотром всех выданных результатов.

## 2.2 Диаграмма классов

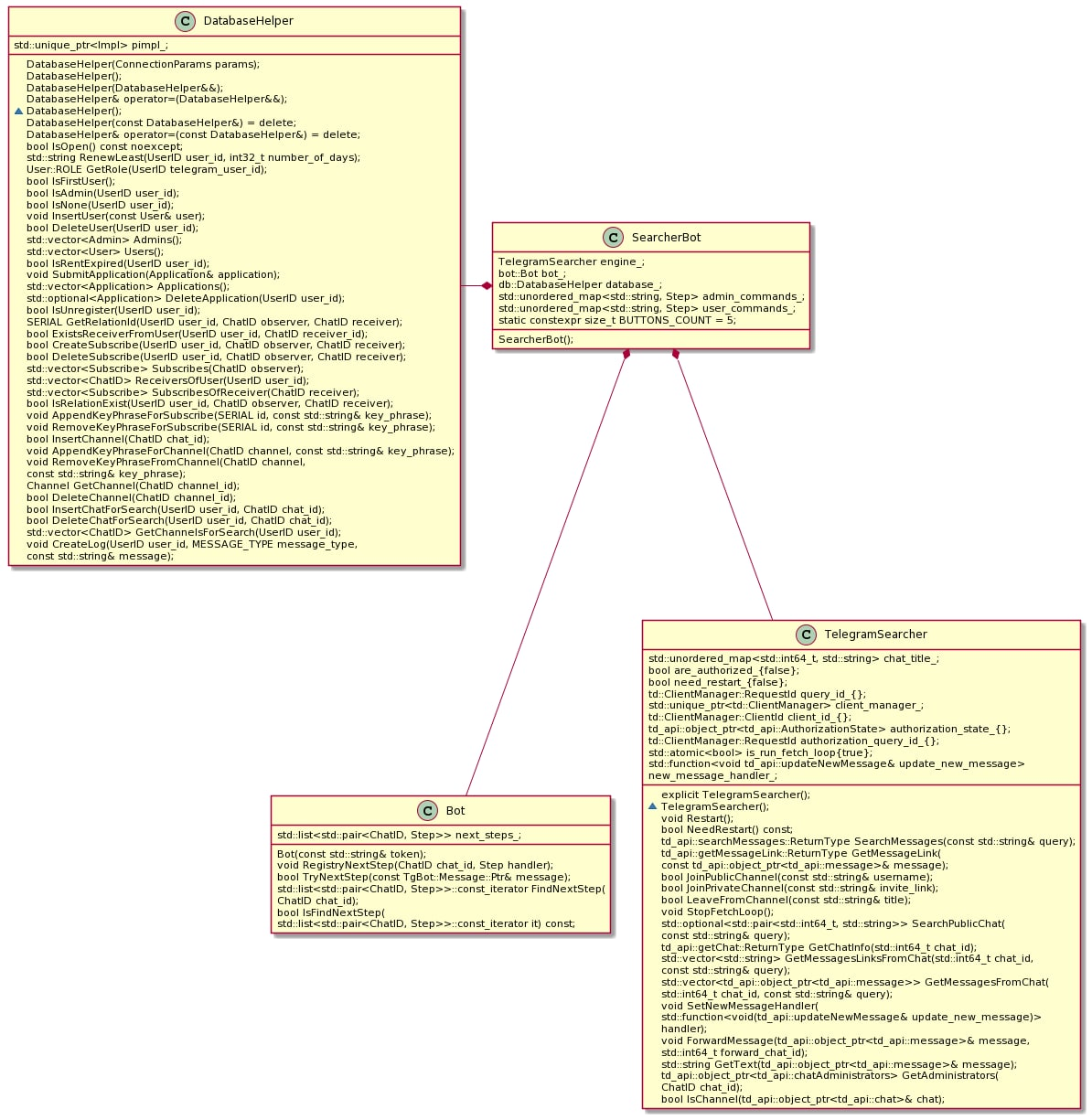


Рисунок 2.1 – Диаграмма классов

## 2.3 Диаграмма прецедентов

Варианты использования системы пользователем и администратором приведены на рисунках 2.2 и 2.3 соответственно. Администратор обладает всем функционалом пользователя, а также может администрировать систему.

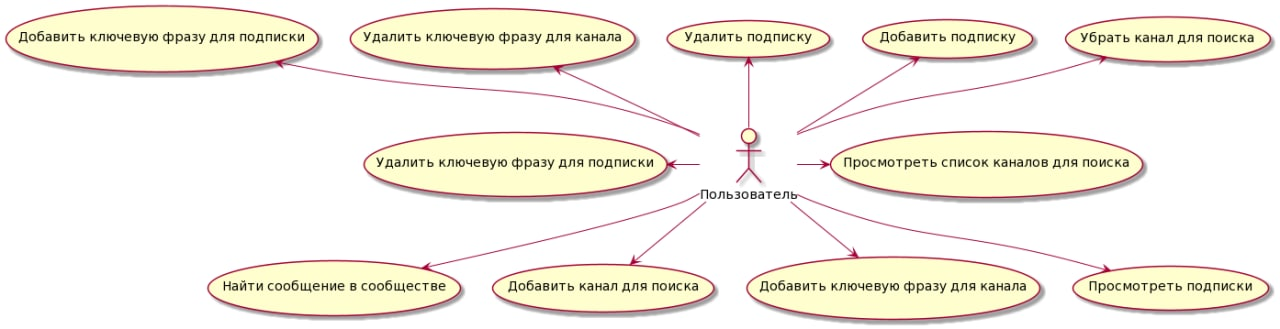


Рисунок 2.2 – Диаграмма прецедентов (Пользователь)

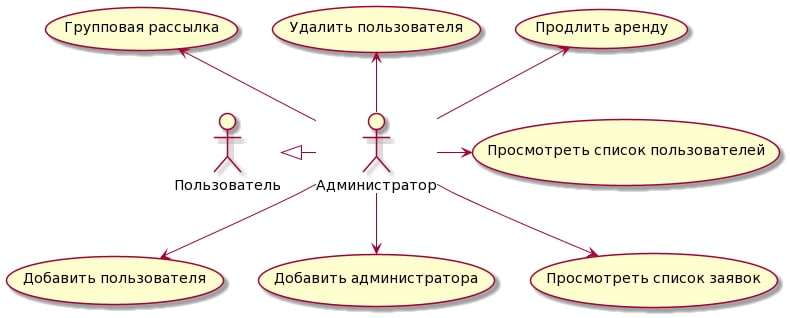


Рисунок 2.3 – Диаграмма прецедентов (Администратор)

## 2.4 Диаграмма размещения

Все модули системы располагаются на одной машине, которая имеет связь с сервером Telegramm.

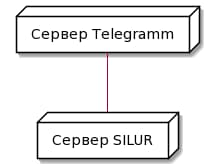


Рисунок 2.4 – Диаграмма размещения

## 2.5 Диаграмма объектов

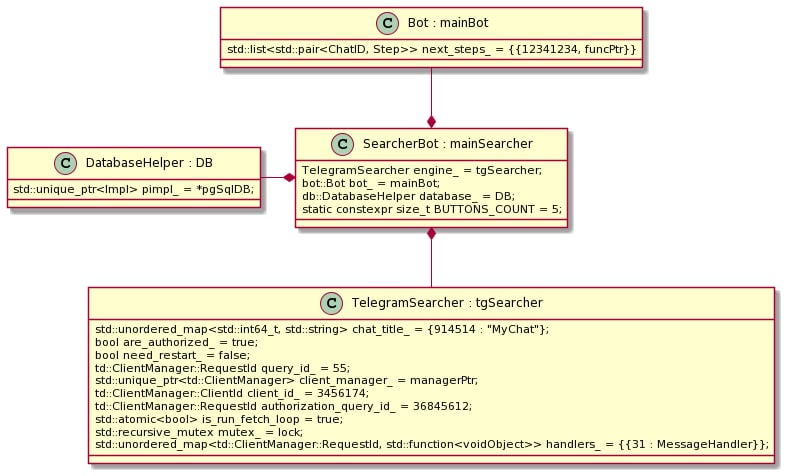


Рисунок 2.5 – Диаграмма объектов

## 2.6 Диаграмма состояний

На рисунке 2.6 приведен пример диаграммы состояний для функции «Добавить ключевую фразу к связи».

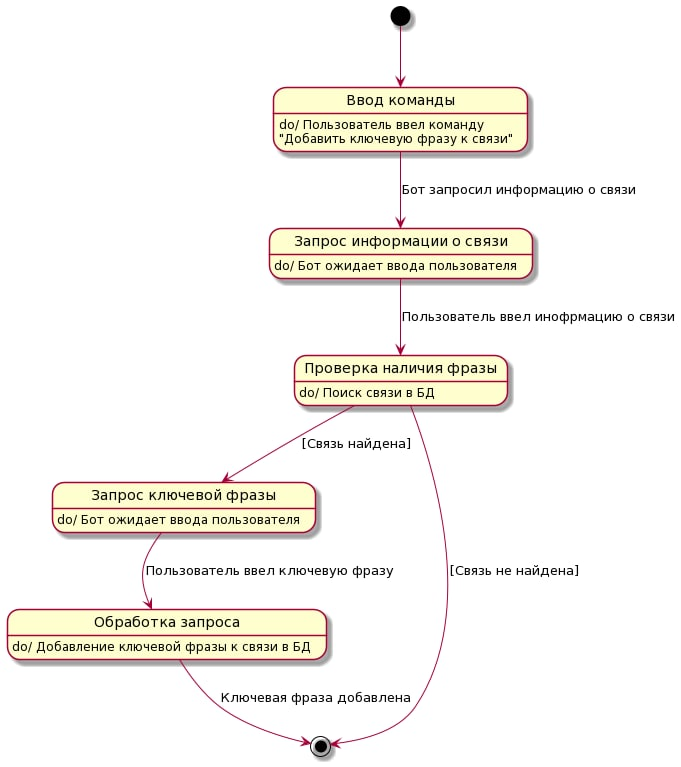


Рисунок 2.6 – Диаграмма состояний

## 2.7 Диаграмма последовательности

На рисунке 2.7 приведен пример диаграммы последовательноси для функции «Добавить ключевую фразу к связи».

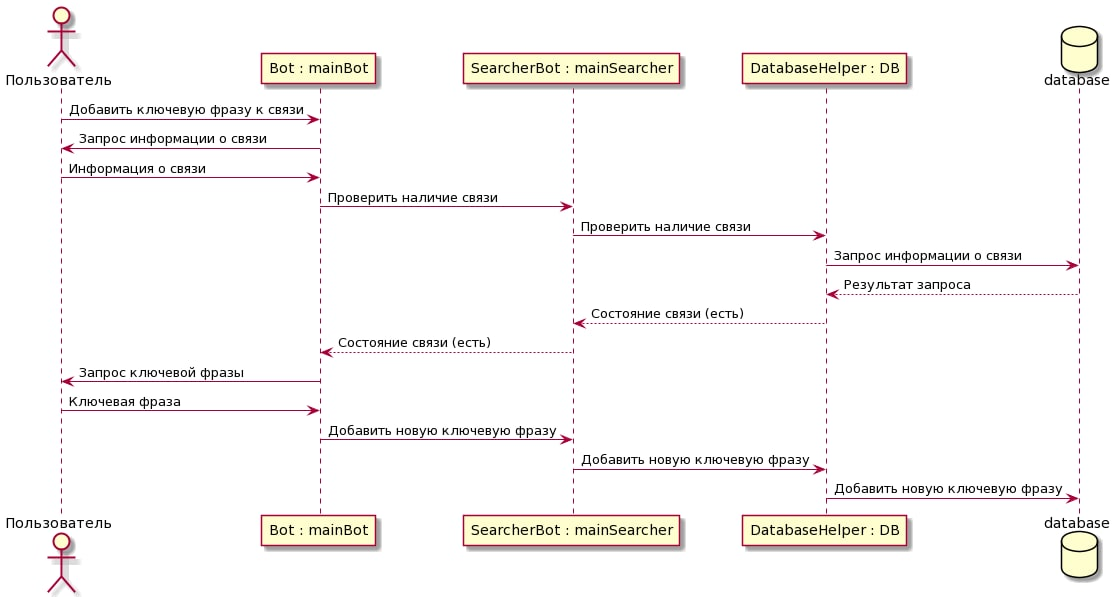


Рисунок 2.7 – Диаграмма последовательности

## 2.8 Диаграмма компонентов

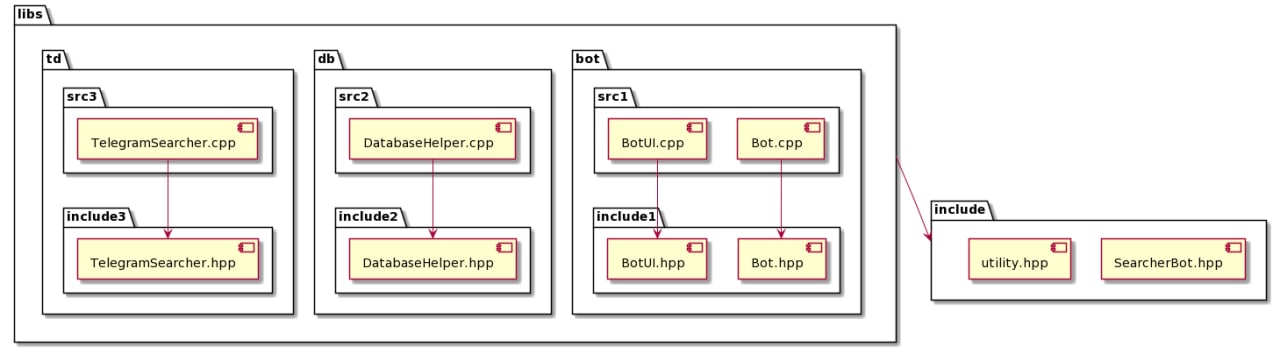


Рисунок 2.8 – Диаграмма компонентов

# 3 Выбор средств

## 3.1 Платформа и окружение

## 3.2 Язык программирования

## 3.3 Third-party

## 3.4 СУБД

# 4 Реализация

## 4.1 Взаимодействие с telegram API

## 4.2 Взаимодействие с Bot API

## 4.3 Взаимодействие с PQXX

## 4.4 Логирование

# 5 Перспективы развития

## 5.1 Идея

Во время автоматизации пересылки сообщений из одного канала в другой было выделено три этапа обработки сообщения:

1. Получение нового сообщения;
2. Обработка сообщения;
3. Отправка сообщения.

Каждый из этих этапов проходит последовательно, а главное оперирует с содержимым сообщения, и использует Telegramm как инструмент для получения и отправки сообщений. Соответственно можно отделить получение, обработку и отправку сообщения, что позволит считывать сообщения из одного источника, обрабатывать сообщение и отправлять сообщение в абсолютно другой информационный источник.

## 5.2 Архитектура

Идейно архитектура сервера будет заключатся в разделении системы на модули:

* Модуль приема сообщений;
* Модуль обработки сообщений;
* Модуль отправки сообщений.

Связи модулей изображены на рисунке 5.1.

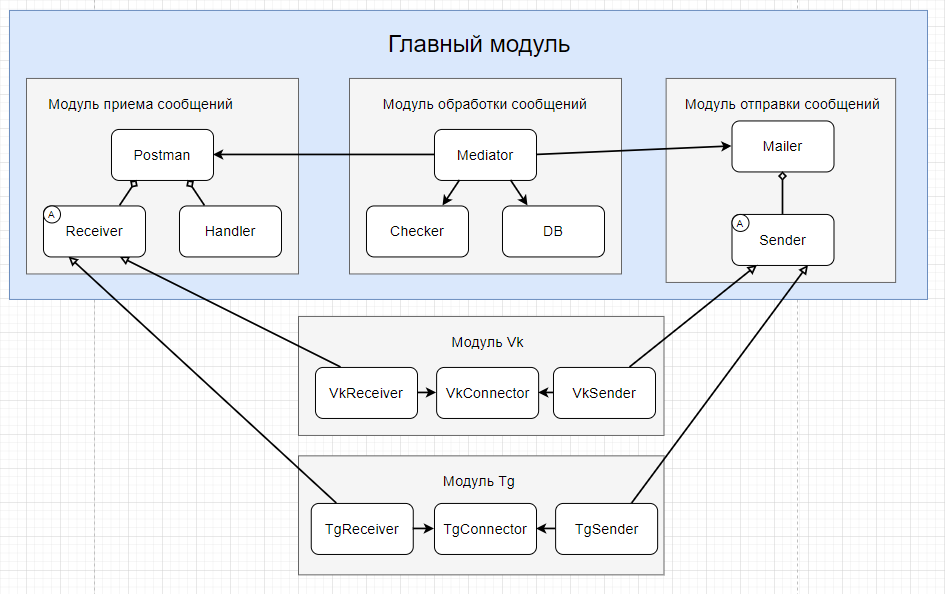


Рисунок 5.1 – Архитектура системы

## 5.2.1 Сервер

На сервере реализуется вся логика проекта, хранится информация в базе данных, происходит обмен информацией с клиентом (сайтом), принимаются и обрабатываются запросы с клиента.

## 5.2.2 Сайт

На сайте реализуется пользовательский интерфейс для администраторов системы. Для управления пользователь должен пройти авторизацию, после чего у него будет возможность добавлять, удалять, просматривать связи между источниками информации. Для администраторов будет возможность добавлять, удалять, просматривать пользователей системы.

# Выводы

Результат работы – система, реализующая агрегатор для пересылки сообщений из одних каналов в другие в мессенджере Telegramm, с возможностью поиска сообщений по ключевым фразам, реализованная при помощи языка программирования С++.

К преимуществам программы относятся: многоуровневый пользовательский интерфейс, обеспечивающий интуитивно понятную навигацию по меню команд, ведение диалога с пользователем при выполнении команд, стабильная работа системы, возможность добавлять, удалять, просматривать связи между каналами и их ключевыми фразами, возможность для администраторов добавлять, удалять, просматривать пользователей системы, наличие групповой рассылки для администраторов, защита системы от несанкционированного использования, возможность искать сообщения в каналах по указанным ключевым словам или фразам.

При дальнейшей работой над системой можно смасштабировать ее до более крупных размеров, а именно добавить возможность работать не только с Telegramm, а и с другими информационными источниками.

# Литература

1. Telegram – Википедия [Электронный ресурс] // Telegram. – Режим доступа: <https://ru.wikipedia.org/wiki/Telegram> .

# Приложение А. Техническое задание

# Приложение Б. Экранные формы

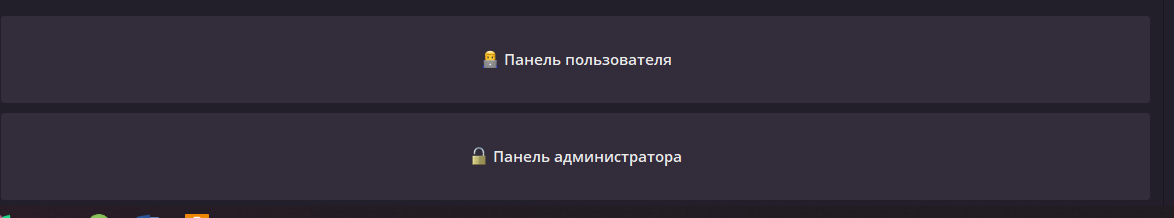


Рисунок Б.1 – Начальное меню

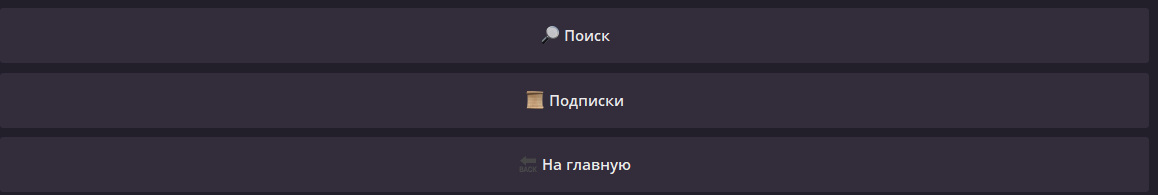


Рисунок Б.2 – Панель пользователя

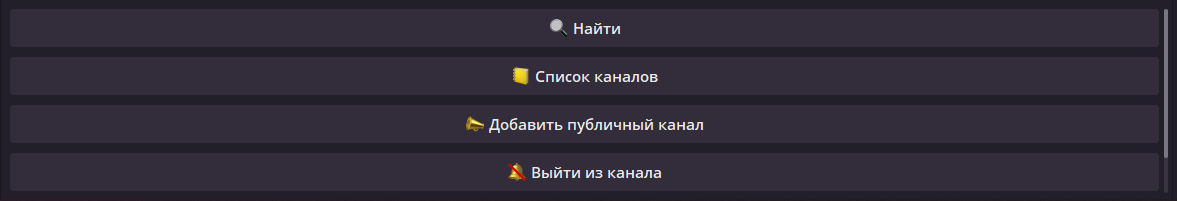


Рисунок Б.3 – Меню «Поиск»

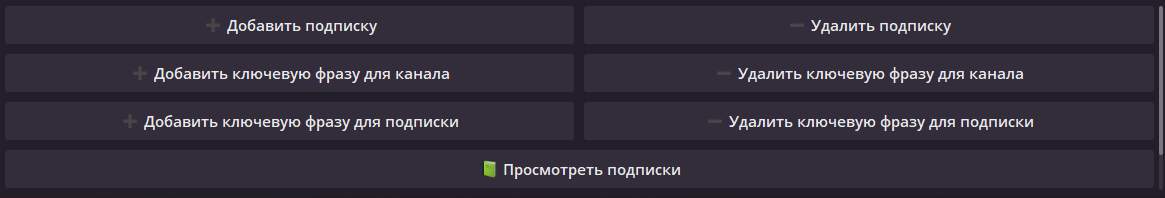
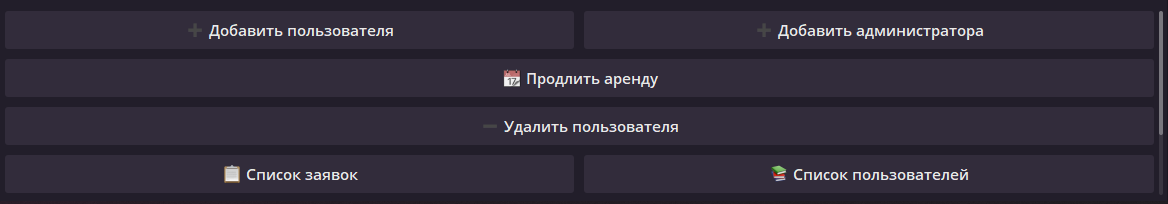


Рисунок Б.4 – Меню «Подписки»



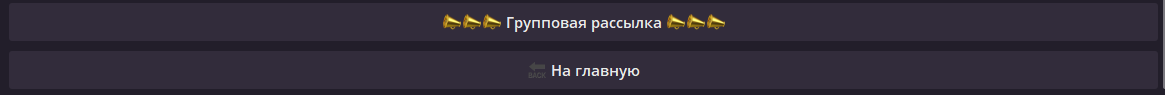


Рисунок Б.5 – Панель администратора

# Приложение В. Статистика приложения

# Приложение Г. Руководство пользователя

Для начала использования необходимо подать заявку на использования системы при помощи Telegram-бота @test\_silur\_bot, а также продлить аренду использования бота. После того, как заявка принята открывается доступ к системе.

Для того чтобы найти сообщение в определенном сообществе необходимо зайти в подменю «Поиск». Перед поиском необходимо добавить канал в список каналов для поиска при помощи пункта «Добавить публичный канал». После того, как требуемый канал добавлен в список каналов нужно нажать на кнопку «Найти». После этого нужно ввести ключевую фразу, по которой будет осуществляться поиск. Результаты поиска выводятся в сообщении с постраничной ориентацией. Для перехода между результатами поиска необходимо нажать на кнопки с цифрами под сообщением. Для удаления канала из списка каналов нужно нажать кнопку «Выйти из канала», после чего ввести название канала. Для просмотра списка каналов нужно нажать на «Список каналов». Для выхода в главное меню нужно нажать «Выбор функции».

Для того чтобы создать подписку нужно перейти в подменю «Подписки». Для добавления подписки нужно нажать на кнопку «Добавить подписку», ввести какой канал слушать, ввести в какой канал публиковать. Для добавления ключевой фразы для канала нужно нажать «Добавить ключевую фразу для канала», после чего указать к какому каналу будет привязана фраза, а после ввести саму фразу. Для добавления ключевой фразы для подписки нужно нажать «Добавить ключевую фразу для подписки» после чего ввести ID канала откуда слушают и куда публикуют через пробел, после чего ввести ключевую фразу. Для просмотра информации о подписках нажмите «Просмотреть подписки». Для удаления подписки нажмите «Удалить подписку», после чего нужно ввести имя подписки. Для удаления ключевой фразы для канала нужно нажать «Удалить ключевую фразу для канала» после чего указать канал и ключевую фразу, которую нужно удалить. Для удаления ключевой фразы для подписки нужно нажать «Удалить ключевую фразу для подписки», ввести ID канала, откуда слушают и куда публикуют через пробел, затем выбрать ключевую фразу, которую нужно удалить.

# Приложение Е. Руководство администратора

Панель пользователя была рассмотрена в руководстве пользователя (Приложение Д).

Для использования функций администрирования нужно нажать «Панель администратора».

Для добавления пользователя нужно нажать «Добавить пользователя», после чего ввести ID пользователя. Для добавления администратора нужно нажать «Добавить администратора», после чего ввести ID пользователя. Для продления аренды пользователю нужно нажать «Продлить аренду», ввести ID пользователя и срок, на который будет продлена аренда. Для удаления пользователя нужно нажать на «Удалить пользователя» и ввести ID пользователя. Для просмотра списка заявок нужно нажать «Список заявок». Для просмотра списка пользователей нужно нажать на «Список пользователей». Для осуществления групповой рассылки нужно нажать «Групповая рассылка», после чего ввести сообщение, которое будет отослано всем пользователям.

# Приложение Ё. Код