ТЕХНИЧЕСКий Проект

Новостной Телеграм-бот «RapidNewsBot»

2023

Содержание

[1 Входные данные 2](#_Toc19491)

[2 Выходные данные 4](#_Toc595)

[3 Разработка алгоритма решения задачи 5](#_Toc18025)

[4 Определение экранных форм 17](#_Toc26147)

[5 Определение конфигурации технических средств 19](#_Toc9013)

[6 Разработка структуры программы 20](#_Toc13997)

[6.1 Структура взаимодействия компонентов программы 20](#_Toc32239)

[6.2 Структура базы данных 20](#_Toc15899)

# Входные данные

Согласно настоящему техническому заданию на разработку новостного Телеграм-бота «RapidNewsBot» система должна обеспечивать сбор новостей, выполнение команд пользователя. Сбор новостей включает в себя следующие этапы:

* Запрос на получение XML документа;
* Получение данных из XML документа;
* Запрос на получение HTML страницы;
* Получение данных из HTML страницы;
* Сохранение полученных данных в базу данных.

Структура XML документа, получаемого в процессе сбора новостей содержит следующие ключевые элементы:

* «<published></published>» - дата опубликования новости в виде строкового значения;
* «<title></title>» - заголовок новости в виде строкового значения;
* «<link></link>» - ссылка на HTML страницу новости в виде строкового значения.

Структура HTML страницы представляет из себя набор HTML-тегов. Ключевыми являются теги «<p></p>», которые содержат тексты новостей в виде строковых значений.

Выполнение команд пользователя включает в себя получение сообщений от пользователя.

Сообщение от пользователя представляет из себя JSON документ, имеющий следующую структуру, представленную на рисунке 1.1.1:

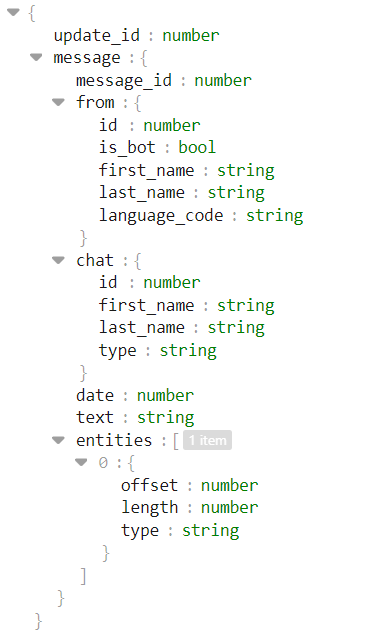


Рисунок 1.1.1 - Структура JSON

# Выходные данные

Согласно настоящему техническому заданию на разработку новостного Телеграм-бота «RapidNewsBot» система должна обеспечивать отправку новостей пользователю по заданным настройкам. Отправка новостей осуществляется с помощью запроса к Telegram API. Структура запроса к Telegram API представлена на рисунке 1.2.1:

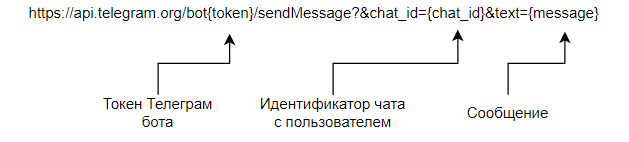


Рисунок 1.2.1 - Структура запроса

Сообщение представляет из себя строковое значение. Для формирования сообщения используются данные из базы данных системы. Структура сообщения представлена на рисунке 1.2.2:

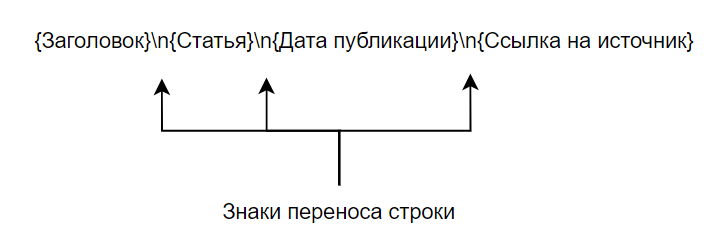


Рисунок 1.2.2 - Структура сообщения

# Разработка алгоритма решения задачи

Согласно настоящему техническому заданию на разработку новостного Телеграм-бота «RapidNewsBot» система должна решать проблему фильтрации новостного потока - осуществлять сбор новостей, отправку новостей на определённые пользователем темы и в определённых временных рамках. Исходя из проблематики и требований к программному продукту можно выделить следующие задачи:

* Сбор новостей;
* Сохранение новостей в базу данных
* Обработка пользовательских сообщений
* Отправка новостей пользователю согласно его настройкам.

Сбор новостей может осуществлялся с помощью следующих новостных агрегаторов, определённых на официальном сайте Федеральной службы по надзору в сфере связи, информационных технологий и массовых коммуникаций (Рисунок 2.1):



Рисунок 2.1 - Реестр новостных агрегаторов

В таблице 2.1 приведено сравнение новостных агрегатором по главным критериям.

Таблица 2.1 - Сравнение новостных агрегаторов

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Интерфейсы | Количество источников | Количество тем |
| Новости@mail.ru | Сайт, RSS | 55-60 | 6 |
| СМИ2 | Сайт | 455 | 9 |
| Рамблер/новости | Сайт, RSS | 120-130 | 10 |
| Новости | Сайт | 5583 | 13 |
| 24СМИ | Сайт | 21 | 11 |

В приведённом сравнении наиболее подходящим новостным агрегатором является Рамблер/новости, так как имеет RSS ленты в качестве интерфейса, что делает доступнее сбор новостей программным способом, также данный источник имеет наибольшее количество тем новостей среди агрегаторов, имеющих RSS ленты в качестве интерфейса.

Программа должна собирать данные из RSS ленты с помощью библиотеки feedparser (п. 5). Из RSS ленты необходимо извлекать следующие данные: заголовок новости, дату опубликования новости, ссылку на полный текст новости. Ссылка на полный текст новости используется для получения полного текста новости, для этого используется библиотека requests (п. 5), для отправки запроса на получение HTML страницы. Извлечение данных происходит с использованием библиотеки bs4 (п. 5), с помощью её возможностей из HTML страницы получается полный текст новости, который всегда заключён в HTML теги «<p></p>». Проверка появления новых новостей осуществляется раз в пять минут. Для обеспечения отбора новых новостей каждой темы из всех имеющихся в RSS лентах требуется конфигурационный файл, который должен хранить данные о темах новостей, ссылках на на RSS ленты, пути до извлекаемых из RSS ленты данных, время последнего обновления для каждой темы. На рисунках 2.1, 2.2 представлена диаграмма процесса сбора новостей.



Рисунок 2.1 - Диаграмма процесса сбора новостей

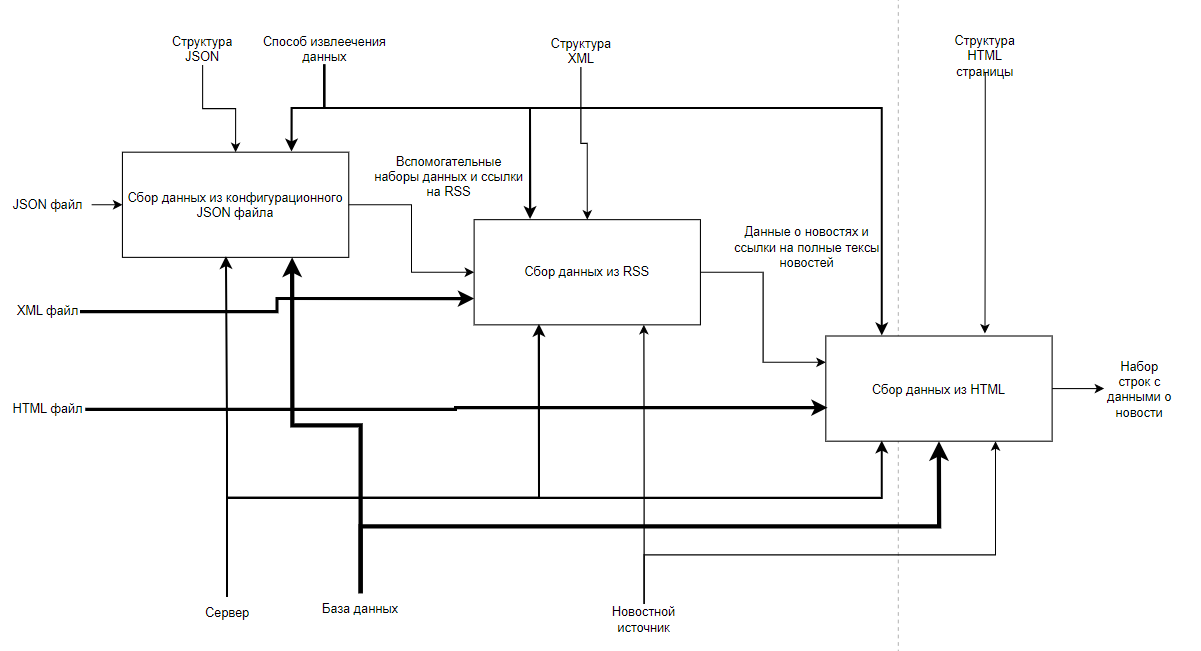


Рисунок 2.2 - Диаграмма процесса сбора новостей

Сохранение новостей в базу данных осуществляется с помощью ORM SQLAlchemy (п. 5), структура базы данных рассмотрена в пункте 5.2.5. На рисунке 2.3 представлена UML Activity diagram, которая отражает алгоритм сбора новостей:

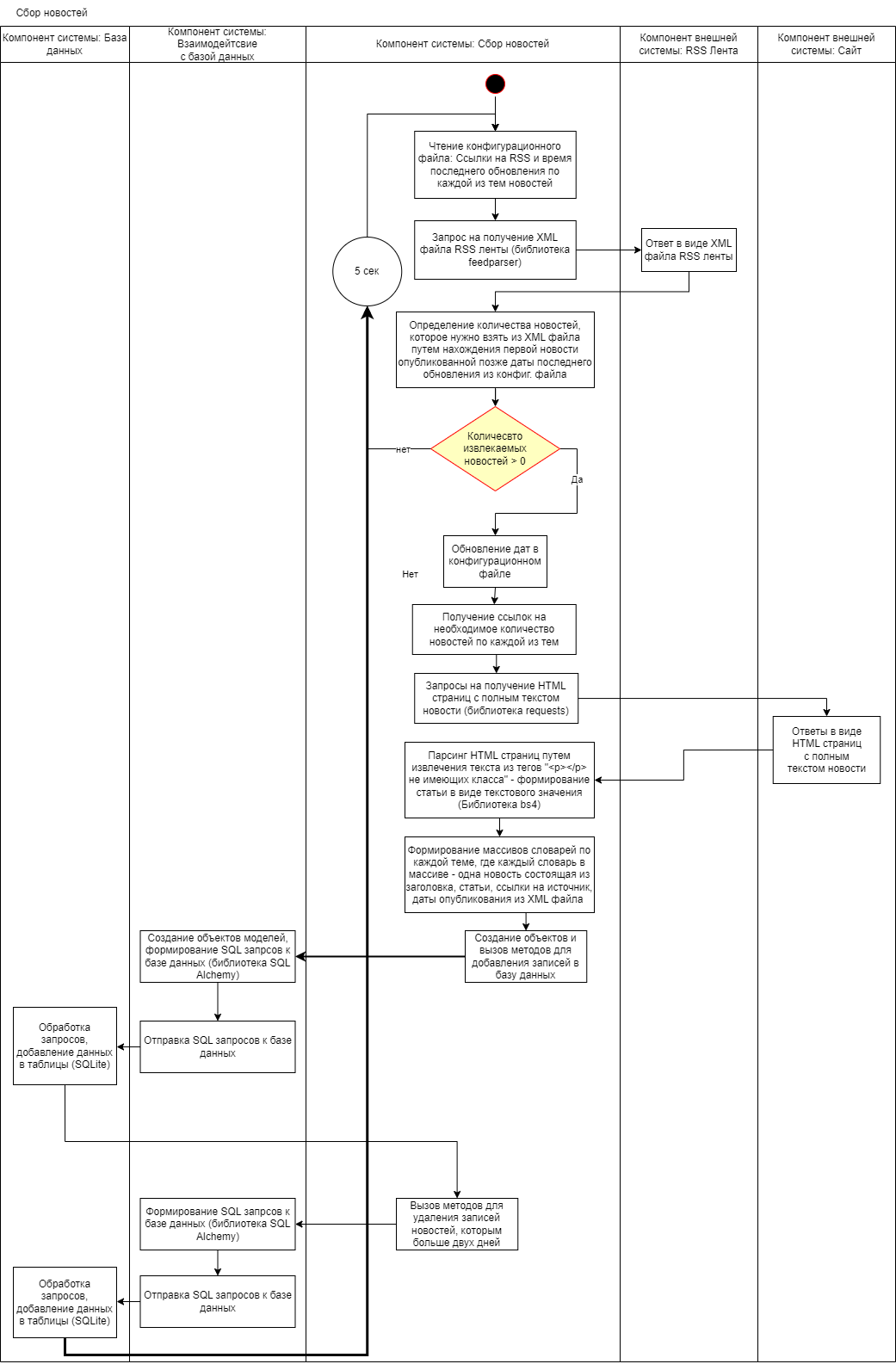




Рисунок 2.3 - Диаграмма алгоритма сбора новостей

Обработка пользовательских сообщений происходит с помощью библиотеки aiogram (п. 5), которая реализует взаимодействие с Telegram API. Обработка пользовательских сообщений осуществляется группой методов, с помощью описанной выше библиотеки, которые вызываются при получении сообщения от пользователя. Каждое сообщение, которое задано в перечне команд изменяет настройки пользователя в зависимости от наполнения сообщения в базе данных. От настроек пользователя зависит перечень тем на которые пользователь получает новости, а также время получения новостей. На рисунке 2.4, 2.5 представлена диаграмма процесса обработки сообщения от пользователя, в данной диаграмме подразумевается, что сообщение от пользователя приходит через Telegram API. На рисунках 2.6 - 2.11 представлены диаграммы алгоритма обработки пользовательских сообщений:

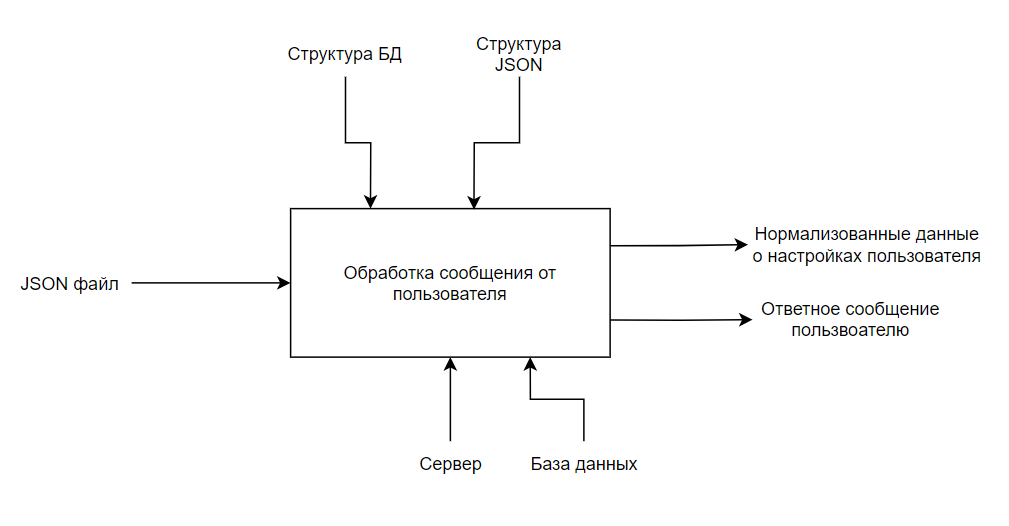


Рисунок 2.4 - Диаграмма процесса обработки сообщения от пользователя

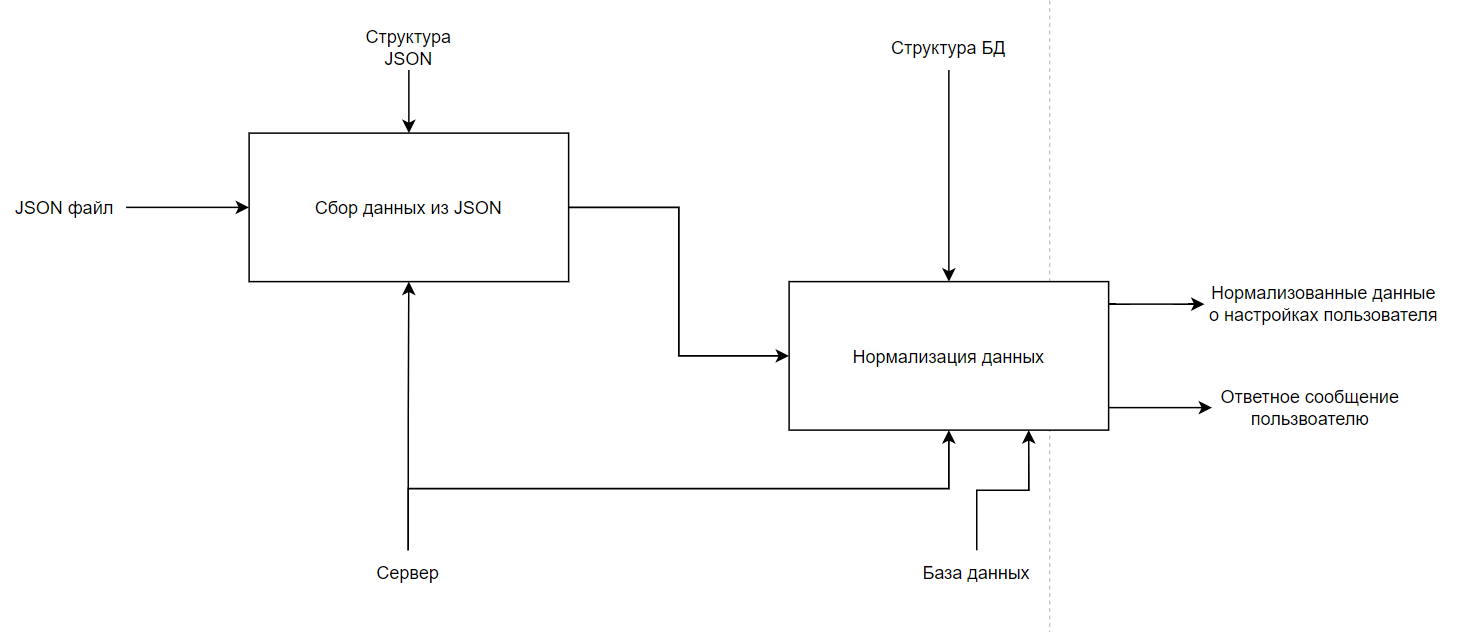


Рисунок 2.5 - Диаграмма процесса обработки сообщения от пользователя

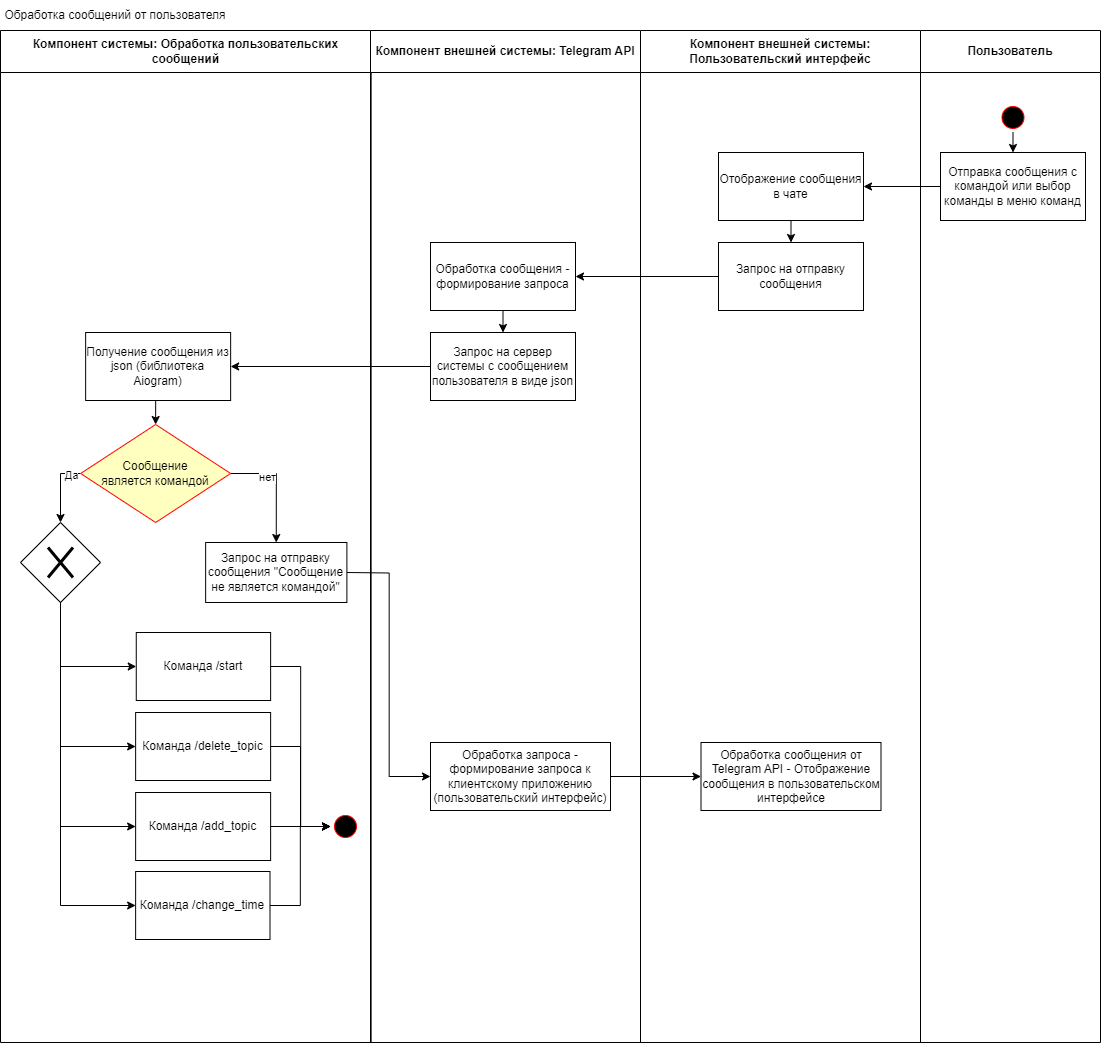


Рисунок 2.6 - Диаграмма алгоритма обработки сообщения от пользователя

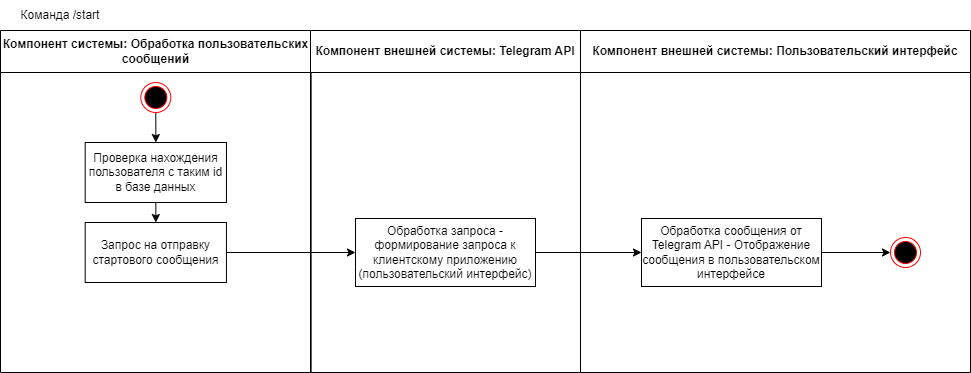


Рисунок 2.7 - Диаграмма алгоритма обработки сообщения от пользователя

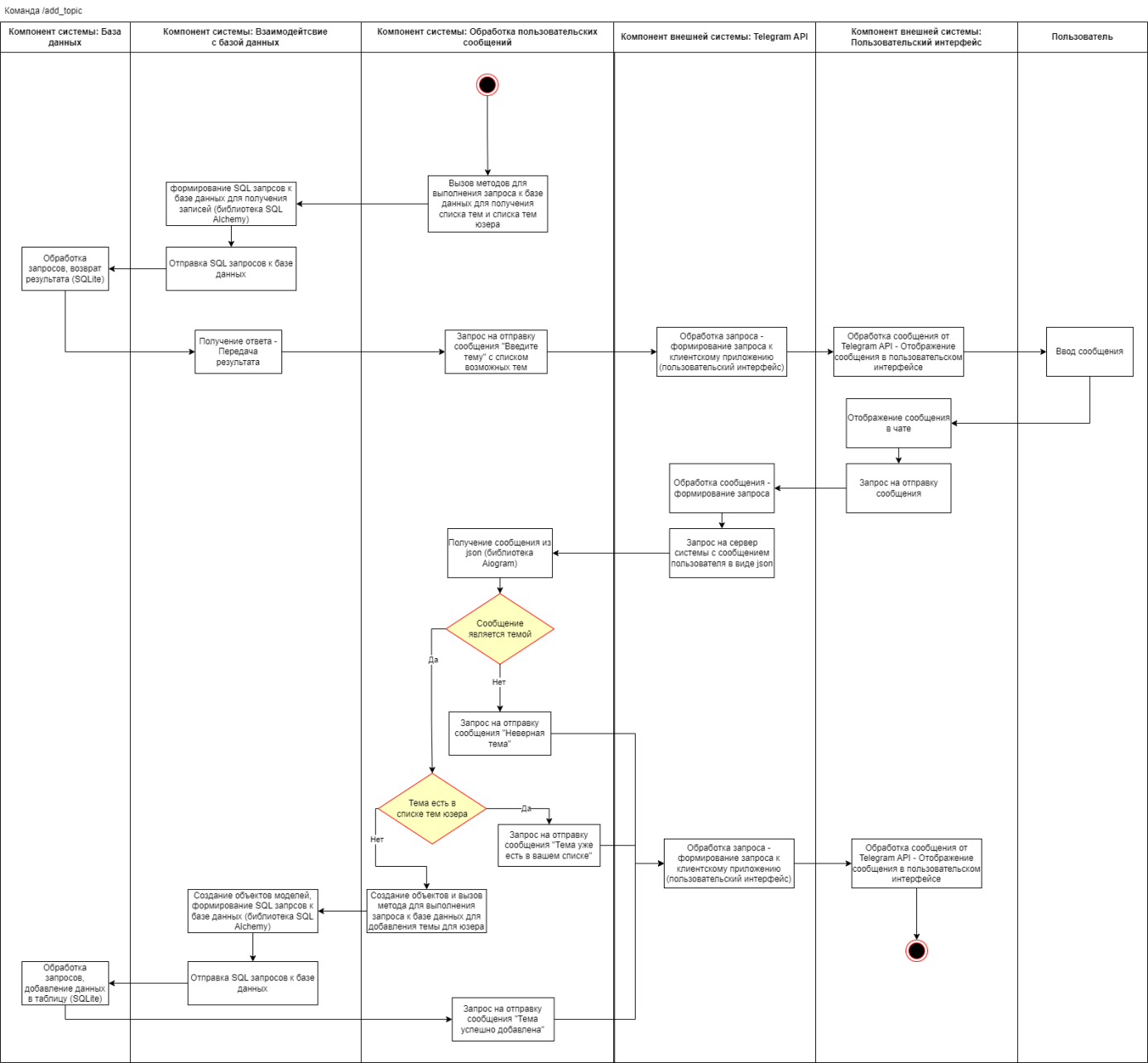


Рисунок 2.8 - Диаграмма алгоритма обработки сообщения от пользователя

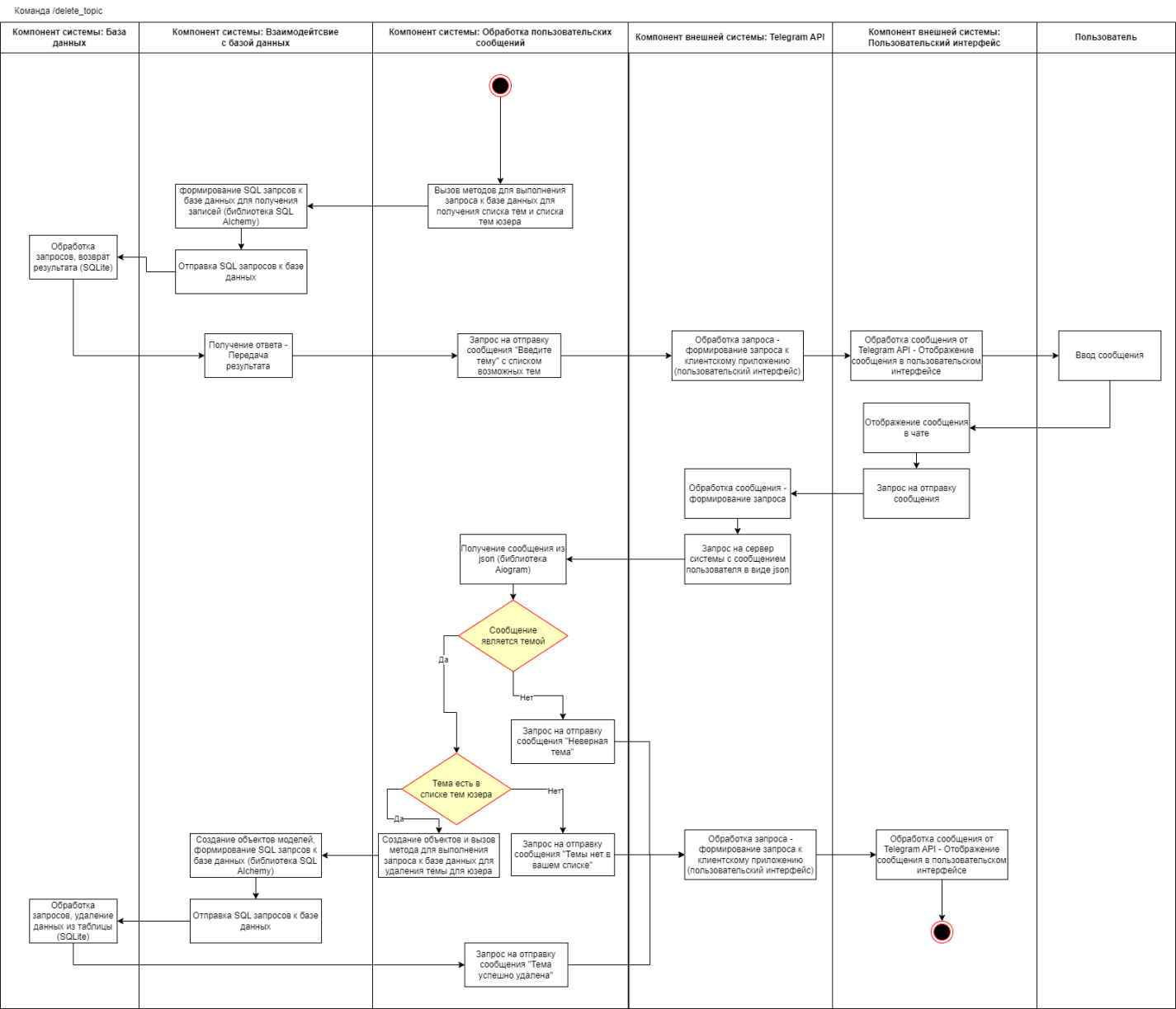


Рисунок 2.9 - Диаграмма алгоритма обработки сообщения от пользователя

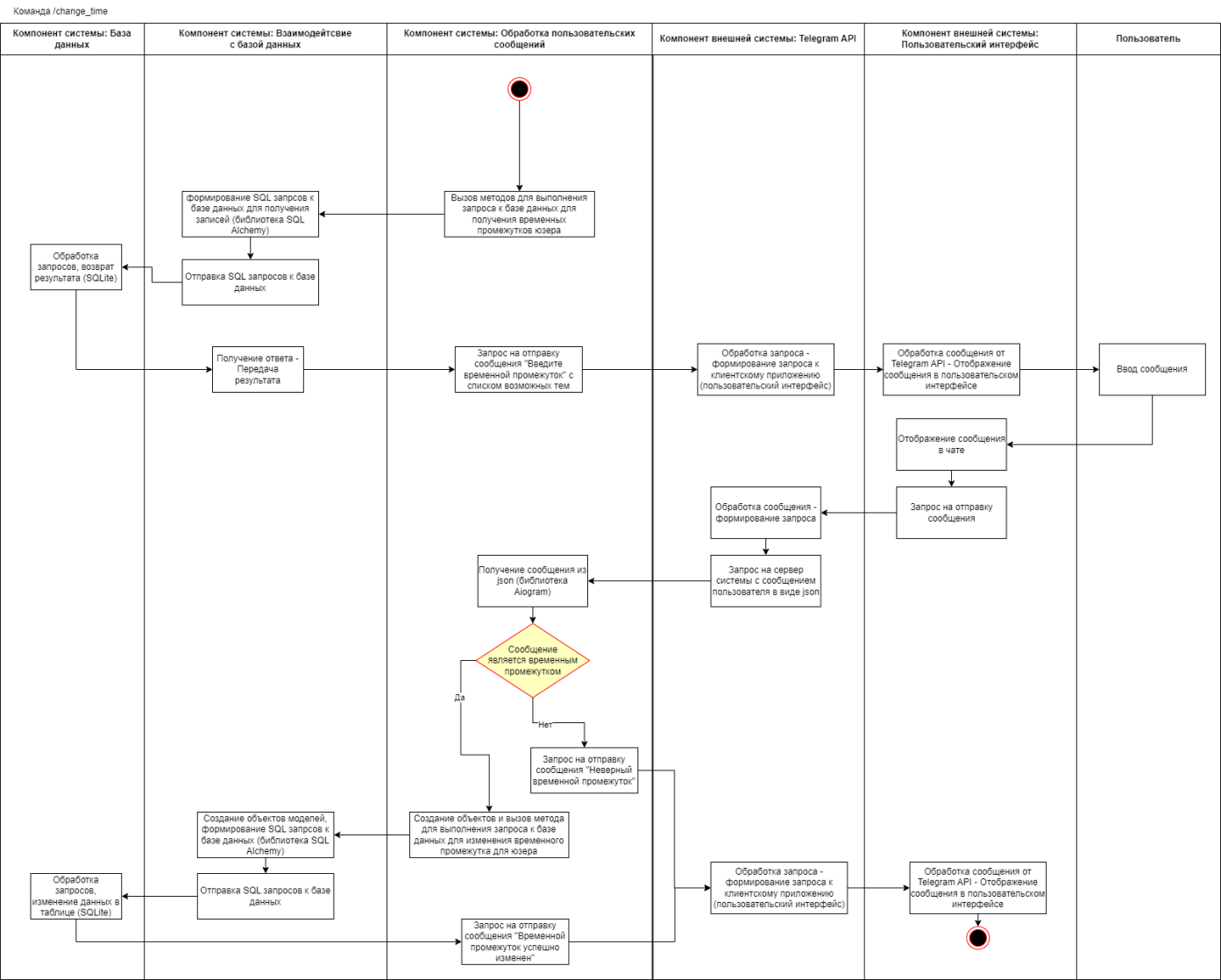


Рисунок 2.10 - Диаграмма алгоритма обработки сообщения от пользователя

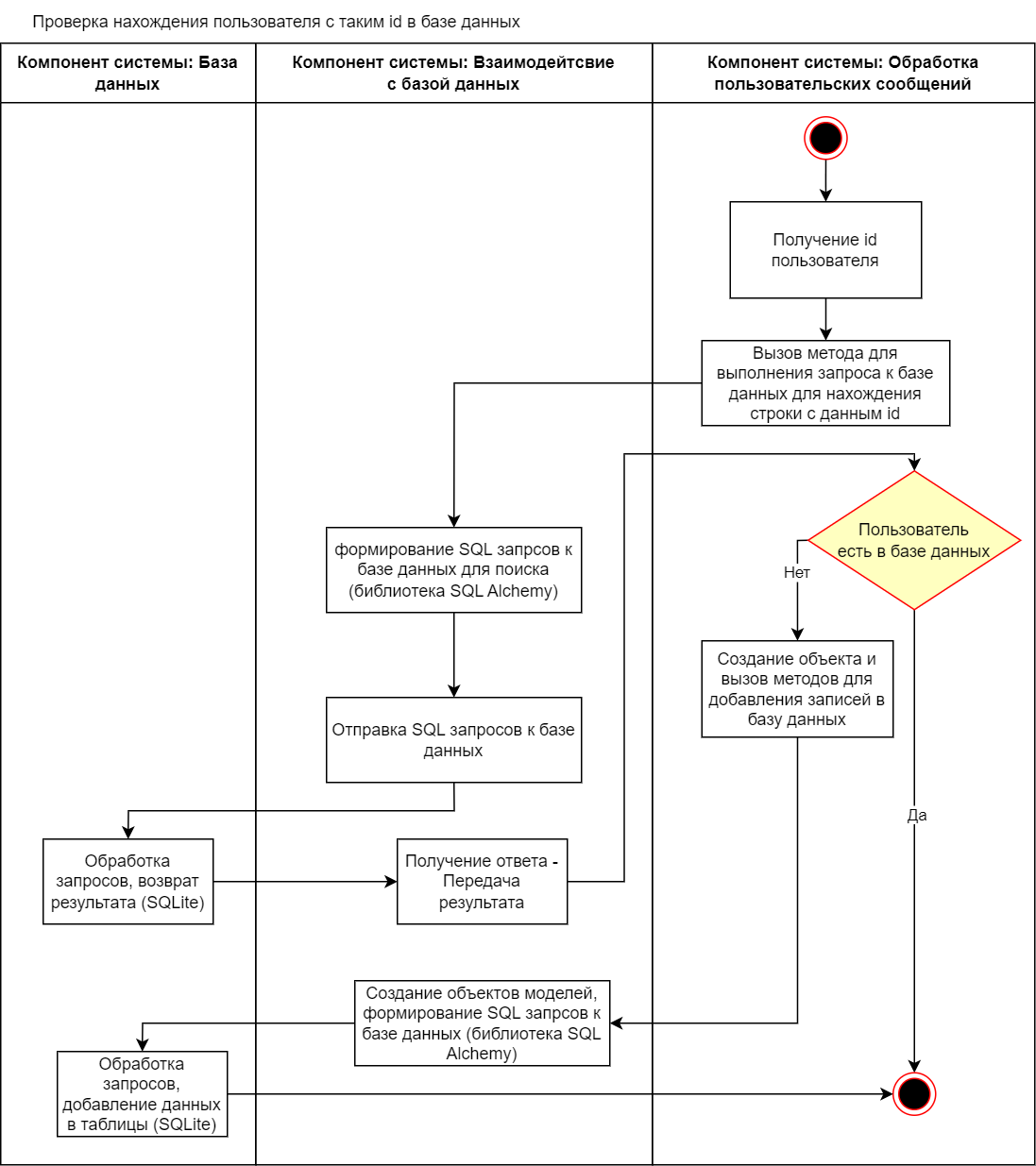


Рисунок 2.11 - Диаграмма алгоритма обработки сообщения от пользователя

Отправка новостей пользователям происходит с помощью библиотеки aiogram (п. 5), которая отправляет запрос к Telegram API. Запуск процесса отправки новостей пользователю происходит каждые 5 сек. Отправка новостей происходит с учётом настроек каждого пользователя, если во время отправки окажется, что пользователь ограничил время отправки на данное время или в базе данных не появилось новых новостей по выбранным пользователем темам, то новости пользователю не отправляются, в другом случае все накопленные новости отправляются пользователю в нескольких запросах - в нескольких сообщениях, где каждому сообщению соответствует одна новость. Новости, которые были опубликованы день назад удаляются. На рисунке 2.12, 2.13 представлена диаграмма процесса отправки новостей пользователю. На рисунке 2.14 представлена диаграмма алгоритма отправки новостей пользователю:

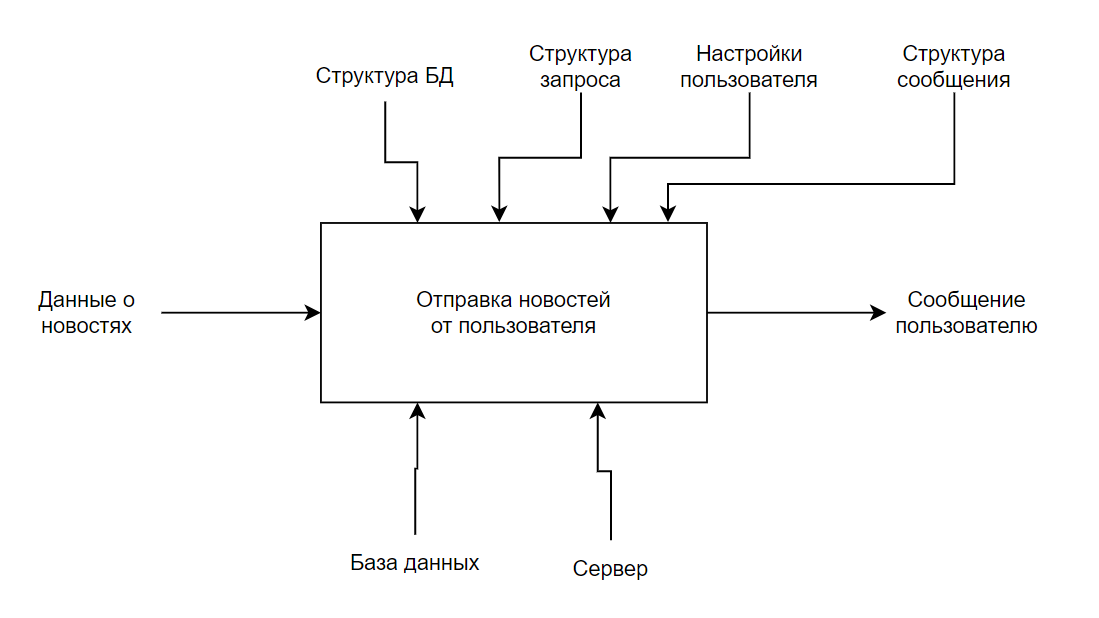


Рисунок 2.12 - Диаграмма процесса отправки новостей пользователю

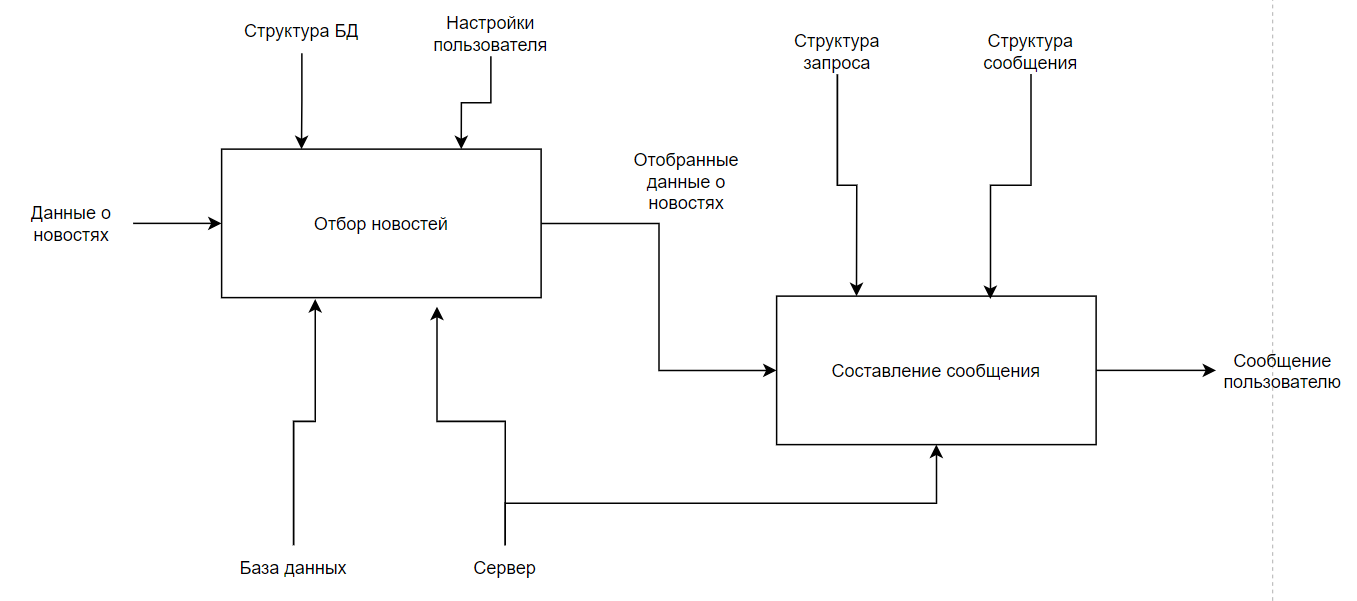


Рисунок 2.13 - Диаграмма процесса отправки новостей пользователю

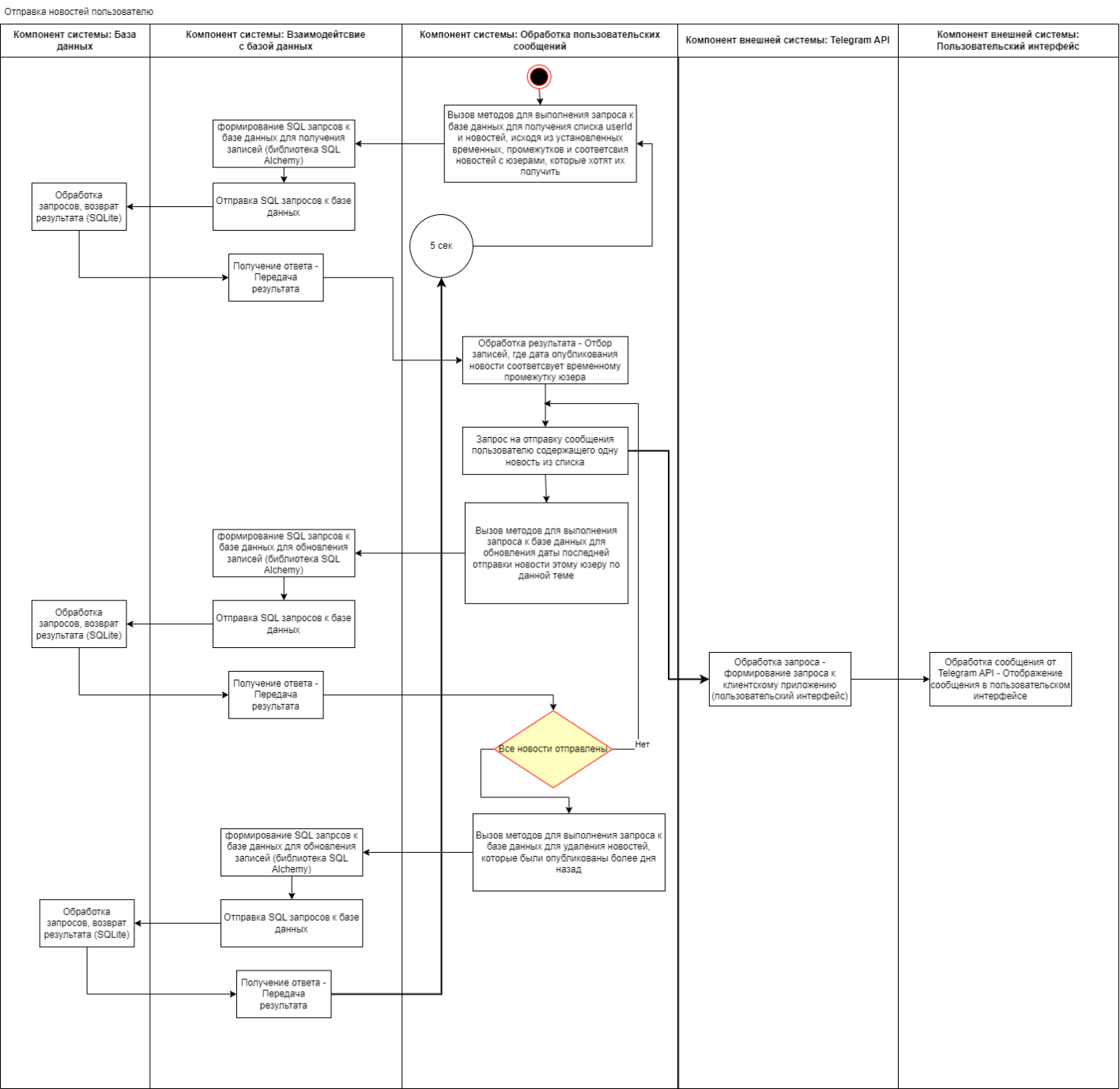


Рисунок 2.14 - Диаграмма процесса отправки новостей пользователю

# Определение экранных форм

В качестве интерфейса для пользователя определён мессенджер Телеграм актуальной версии, который предоставляет возможность создания кнопки «Меню» в чате с ботом, в которой можно определить набор команд, для предоставления пользователю удобного выбора интересующей его команды а также возможность создания набора кнопок для выбора темы. Экранные формы представлены на рисунках 4.1, 4.2, 4.3:

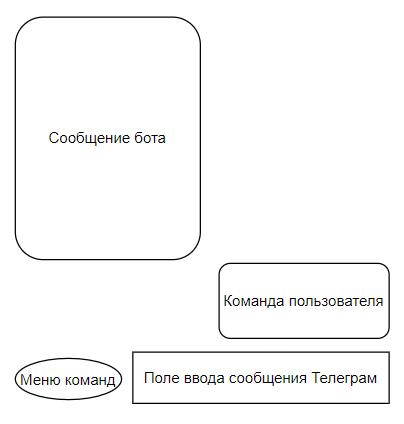


Рисунок 4.1 - Макет пользовательского интерфейса

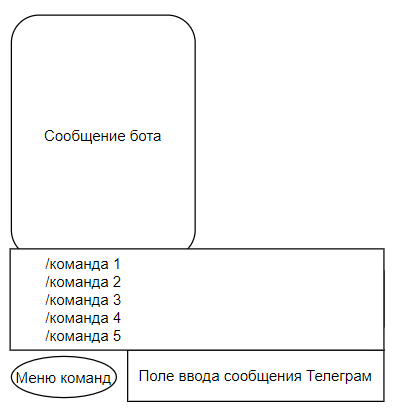


Рисунок 4.2 - Макет пользовательского интерфейса

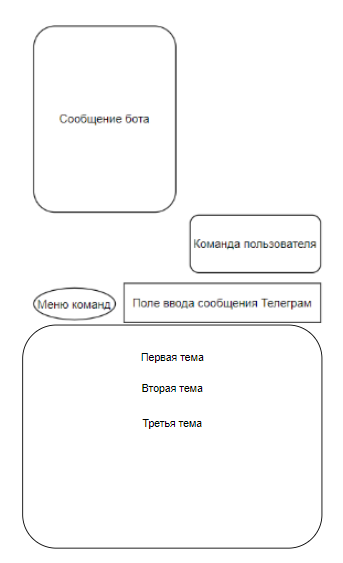


Рисунок 4.3 - Макет пользовательского интерфейса

# Определение конфигурации технических средств

Конфигурация технических средств пользователя системы определяется минимальными системными требованиями актуальной версии Telegram.

Конфигурация технических средств системы:

* Язык программирования - Python 3.11;
* Сбор новостей осуществляется средствами библиотек feedparser 6.0.10, requests 2.31.0, fake-useragent 1.2.1, beautifulsoup4, для обработки xml документа, отправки запросов, обеспечения защиты от блокировок, обработки HTML документа соответственно;
* Взаимодействие с базой данных SQLite 3.4.1 происходит средствами ORM SQLAlchemy 2.0.20;
* Обеспечение обработки и отправки сообщений через Telegram API происходит с помощью aiogram 3.1.1.

# Разработка структуры программы

## Структура взаимодействия компонентов программы

Структура взаимодействия компонентов программы представлена на рисунке 5.1.1:

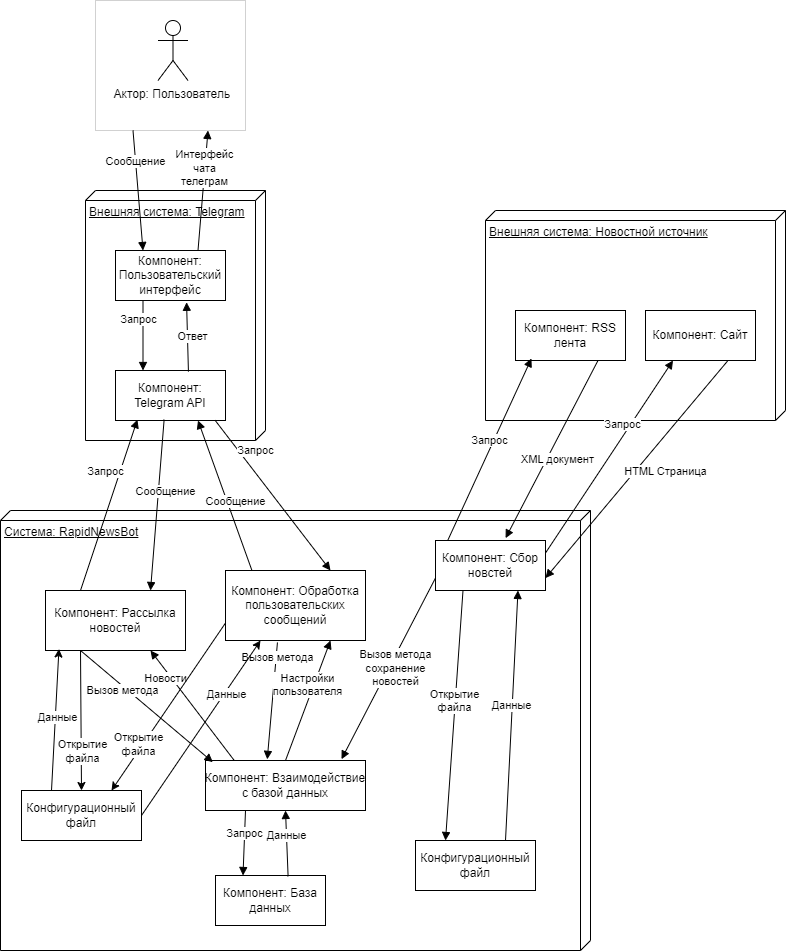


Рисунок 5.1.1 - Структура взаимодействия компонентов программы

## Структура базы данных

В качестве СУБД выбрана встраиваемая файловая реляционная SQLite (п. 5).

База данных должна обеспечивать сохранение настроек пользователя в виде выбранных тем новостей, временного периода, когда пользователь хочет получать новости, сохранение составных частей сообщения пользователю, а также сохранение тем новостей.

На рисунке 5.2.1 представлена структура базы данных:

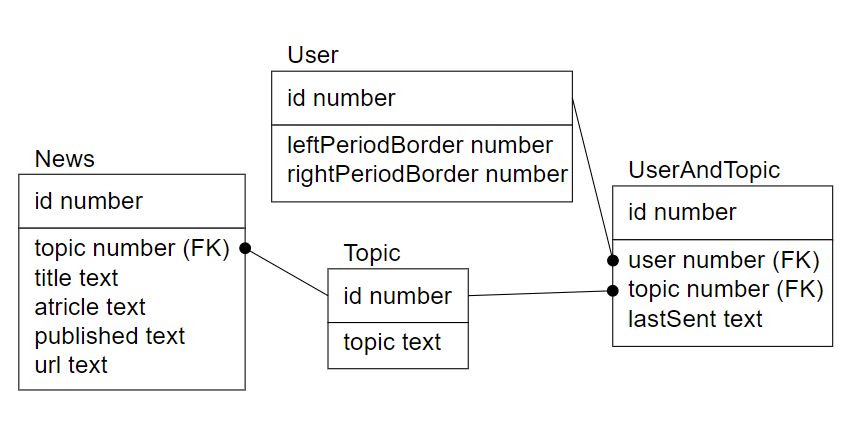


Рисунок 5.2.1 - Структура базы данных

Таблица User содержит поля leftPeriodBorder и rightPeriodBorder, которые нужны для хранения временного периода в часах, где каждое поле содержит число от 0 до 23, при этом разница между числами должна быть не меньше 1. Данная таблица нужна для сохранения настроек пользователей.

Таблица Topic - статическая таблица, которая содержит фиксированный набор тем новостей в виде текстовых значений: World, Moscow, Politics, Society, Incidents, ScienceAndTechnology, ShowBusiness, Military, Games, Analytics. Данная таблица нужна для нормализации структуры базы данных.

Таблица UserAndTopic осуществляет связь многие ко многим между таблицами Topic и User, тем самым обеспечивая сохранение выбранных пользователем тем новостей, которые он хочет получать, также таблица содержит поле lastSent, для сохранения времени последней опубликованной новости по каждой из тем, выбранных пользователем, для обеспечения разделения новостей на уже отправленные и ещё не отправленные по дате.

Таблица News нужна для сохранения составных частей будущих сообщений пользователям, она содержит заголовок, статью, дату опубликования, ссылку на источник новости, а также тему новости для обеспечения разделения новостей на темы - связь один ко многим между таблицами Topic и News.