МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»
Кафедра математической кибернетики и компьютерных наук

РАЗРАБОТКА МИКРОСЕРВИСНОГО ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ СБОРА ДАННЫХ О ЖИЛИЩНО-КОММУНАЛЬНЫХ АВАРИЯХ В ГОРОДЕ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Ивановой Анастасии Дмитриевны

Научный руководитель
доцент, к. ф.-м. н. _____ Ю. Н. Кондратова

Зав.кафедрой,
к. ф.-м. н., доцент _____ С. В. Миронов

Саратов 2024

ВВЕДЕНИЕ

Актуальность темы. В наше время цифровые технологии используются во всех сферах жизни, решение множества проблем доступно онлайн, и это, несомненно, делает жизнь людей более комфортной и удобной, позволяет им рациональнее управлять своими временем и силами. Но в то же время, вместе с цифровизацией резко увеличился объем информации, которую человеку ежедневно приходится обрабатывать, и зачастую в таком большом потоке данных бывает сложно найти нужную и достоверную информацию, увидеть полую картину при обзоре различных ситуаций.

Для быстрой и качественной работы с большим объем данных существует множество инструментов: парсинг для сбора информации, машинное обучение — для ее обработки, а также разнообразные способы ее визуализации, выбор которых зависит от предметной области и целей проекта. Одной из задач, которую можно решить с использованием вышеописанных средств, является задача визуализация состояния городских объектов на основе жалоб населения. На территории города существуют районы и дома, которые оказываются менее благоприятными для проживания, чем остальные, ввиду ряда факторов. Таким образом, проблемы, собранные из открытых источников напрямую с населения, могут быть оформлены в единую базу для информирования тех, кто выбирает жилье, а также для органов государственной и муниципальной власти, осуществляющих контроль над городским хозяйством Саратова.

Целью работы является создание веб-приложения для визуализации жилищно-коммунальных неполадок города Саратов. Программа должна собирать жалобы горожан с открытых источников, распознавать адреса и категорию жалоб, а после отображать на карте в виде соответствующих меток.

Результатом работы программы должна быть микросервисная система, включающая в себя сайт, отображающий основные проблемы городского хозяйства, влияющие на качество жизни горожан. Для достижения этой цели были поставлены **следующие задачи**:

- изучение и выбор методов сбора информации, разработка парсера;
- изучение методов анализа данных для обработки текста, разработка отвечающей за это подсистемы;
- изучение инструментов API Яндекс Карт;
- разработка базы данных для хранения информации об авариях;

— разработка системы взаимодействия подсистем;
— разработка сайта для отображения собранной и обработанной информации на карте.

Исследование в области разработки системы мониторинга состояния городских объектов представляет собой важный этап в цифровизации городской жизни, повышении качества жилищно-коммунальных услуг и обеспечения осведомленности населения.

Структура и объём работы. Выпускная квалификационная работа состоит из введения, 2 разделов, заключения, списка использованных источников и 4 приложений. Общий объем работы — 100 страниц, из них 72 страницы — основное содержание, включая 20 рисунков, цифровой носитель в качестве приложения, список использованных источников информации — 31 наименование.

Основное содержание работы

Первый раздел «Постановка задачи и инструменты для ее решения» посвящен описанию решаемой в работе проблемы, анализу источников информации по жилищно-коммунальным авариям в городе Саратов, построению архитектуры приложения и обзору инструментов, которые применяются для разработки.

Описание проблемы. К основным жилищно-коммунальным проблемам можно отнести проблемы с водоснабжением, горячим водоснабжением и теплоснабжением, электроснабжением, вывозом твердых бытовых отходов и дорожным хозяйством. По ряду факторов определенные территории города оказываются менее благоприятными для проживания, чем остальные. Поэтому единая база, содержащая информацию о всех жилищно-коммунальных авариях, отражала бы состояние городских объектов и была бы полезна для людей, которые планирует покупать или снимать жилье, а также для органов государственной и муниципальной власти, осуществляющих контроль над городским хозяйством, для повышения качества и эффективности их работы.

Обзор существующих решений. На данный момент в Саратовской области нет единой базы и визуальной модели территории, отображающих все жилищно-коммунальные аварии, но есть открытые источники, из которых можно извлечь информацию о них. Наиболее содержательным и непредвзятым источником можно назвать Telegram-чат СМИ ЧП-Саратов (`@chpsaratovlive`), в котором жители Саратовской области делятся жалобами и получают оперативную поддержку от администраций районов Саратова и других компетентных источников. Также некоторые ресурсоснабжающие компании выкладывают на свои сайты список аварийных отключений, например, сайты водоснабжающих организаций: ООО «Концессии водоснабжения — Саратов» и МУПП «Саратовводоканал».

Архитектура системы. Для создания требуемой системы была выбрана микросервисная архитектура, в том числе для реализации ее серверной части использовался шаблон MVC. Система, приведенная на рисунке 1, состоит из четырех компонентов. За парсинг и обработку текста отвечают два самостоятельных сервиса, главный сервис обращается к ним при помощи HTTP-запросов, а также взаимодействует с базой данных. При этом основной сервис возвращает на запросы представления в виде веб-страниц, то есть представляет собой сайт.

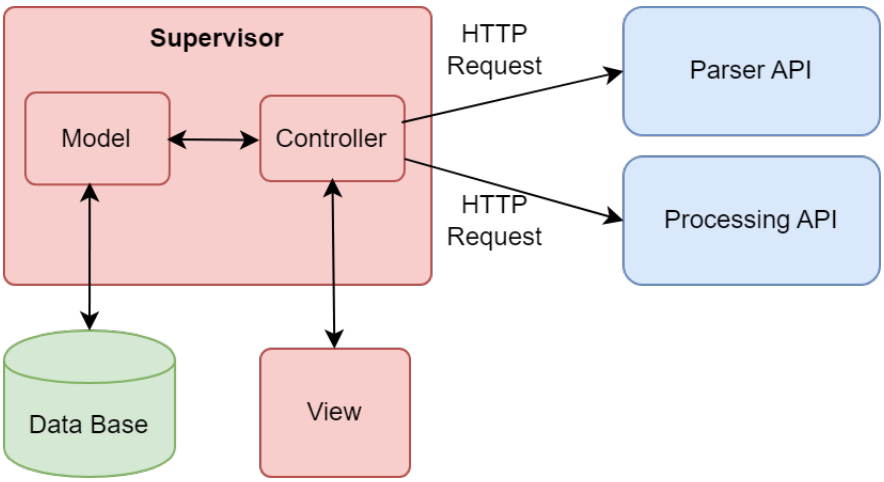

Рисунок 1 – Архитектура разрабатываемой системы

Инструменты для решения задачи. Для сбора данных из Telegram был выбран API-фреймворка Python Pyrogram. Он позволяет легко взаимодействовать с основным Telegram API через учетную запись пользователя или идентификатор бота с использованием Python, а также предоставляет множество методов для получения практически любой информации: считывать сообщения из чатов, извлекать из них текст, медиафайлы, ссылки и прочее, получать список подписчиков каналов или участников чатов.

Для парсинга веб-сайтов была выбрана библиотека BeautifulSoup, предназначенная для извлечения данных из файлов HTML и XML. BeautifulSoup предоставляет возможность навигации по деревьям синтаксического анализа и удобную работу с тегами и их атрибутами.

Собранную при помощи парсинга информацию необходимо обработать и проанализировать. В этой работе извлеченные данные представляют собой совокупность текстов, содержащих адреса, поэтому рассматриваться будет обработка естественного языка.

В данной работе использовалась Python-библиотека Natasha. Эта библиотека решает базовые задачи обработки естественного русского языка: сегментация на токены и предложения, морфологический и синтаксический анализ, лемматизация, извлечение именованных сущностей, в том числе адресов. Для новостных статей качество на всех задачах сравнимо или превосходит существующие решения.

В совокупности с данной библиотекой использовался простой, но эффективный модуль re для повышения качества извлечения адресов из неформальных текстов. Этот модуль предназначен для анализа и обработки текстов при

помощи регулярных выражений, который предоставляет возможность поиска, замены и разбиения по определенному шаблону, а также группировку результатов для извлечения частей совпадений.

Для визуализации собранных и обработанных данных о жилищно-коммунальных авариях была выбрана карта на веб-сайте, встроенная при помощи JavaScript API Яндекс.Карт. API позволяет интегрировать интерактивные Яндекс Карты на сайт, предоставляет возможность работы с базовыми картографическими сервисами Яндекса в браузере, а также гибкую настройку карты для разработчика.

В качестве инструмента для создания веб-приложения был выбран фреймворк FastAPI. Это быстрый, высокопроизводительный веб-фреймворк, основанный на стандартах OpenAPI и JSON Schema для создания сервисов на языке Python. Ключевую роль в управлении этими сервисами играет Docker. Он обеспечивает упаковку каждого сервиса в изолированный контейнер, включая все его зависимости, что гарантирует переносимость и согласованность среды выполнения.

Второй раздел «Разработка программного комплекса» посвящен реализации всех микросервисов веб-приложения, которые включают в себя сервис для парсинга, сервис для обработки текста и серверную часть системы.

Разработка сервиса для парсинга. В разрабатываемой системе парсер — это самостоятельный сервис, задача которого заключается в сборе данных с заданных источников и их предоставлении по запросу. В данном разделе представлено описание модулей для двух используемых источников: Telegram-чата и вебсайтов водоснабжающих организаций. Требуемый функционал для каждого источника включает в себя два GET-контроллера с параметром `last_message_id` (локальный идентификатор крайнего сообщения), предназначенных для получения всех новых записей и получения фиксированного числа старых записей относительно имеющихся.

Разработка сервиса для обработки данных. Сервис для обработки данных в разрабатываемой системе так же является самостоятельным API, которое включает в себя следующий функционал: определение категории проблемы по тексту жалобы, извлечение из текста адреса объекта, на котором возникла авария, а также геокодирование. В данном разделе описывается реализация контроллеров, выполняющие соответствующие задачи.

Разработка серверной части системы. В рамках разрабатываемой системы описанные ранее API являются самостоятельными, тогда как серверная ее часть представляет собой главный сервис, использующий эти API для сбора и обработки данных, взаимодействующий с базой данных для осуществления манипуляций с ними и отвечающий за обновление представлений этих данных в виде HTML-страниц и доступ к ним.

База данных и ORM. В качестве базы данных была выбрана PostgreSQL. Для разрабатываемой системы в базе данных используется сущность для сообщений, — messages. Они являются центральными единицами в системе, так как отражают жалобу на некоторый объект и всю необходимую информацию о ней. В данном разделе приводятся скрипты инициализации таблиц, описываются их поля, представляется настройка взаимодействия с базой данных при помощи ORM SQLAlchemy.

Инициализация сервиса. В данном разделе приводится реализация главного сервиса и обновления данных. Для основного сервиса описывается структура и инициализация. Ежедневное обновление данных осуществляется с помощью планировщика из библиотеки APScheduler и асинхронного HTTP-клиента httpx. В следующих подразделах рассматриваются группы контроллеров сервиса и их вспомогательные функции.

Получение и обработка данных. В данной части работы рассматривается реализация обработчика для эндпоинта, который отвечает за получение новых и старых сообщений. Обработчик настроен на выполнение при GET-запросах к определенному URL и возвращает HTML-ответ с результатами. Функция обрабатывает запрос, извлекает сообщения на основе переданных параметров, обрабатывает их и записывает в базу данных.

Представление данных. В этой части работы описываются обработчики, отвечающие за представление данных, имеющихся в базе, дополнительная фильтрация и структура возвращаемых HTML-шаблонов. Обработчики прописаны для списка записей, конкретных записей и изображений, привязанных к конкретным записям. На рисунках 2, 3 представлены примеры возвращаемых страниц. Фильтрация данных направлена на удаление дубликатов записей и исключения сообщений от организаций.

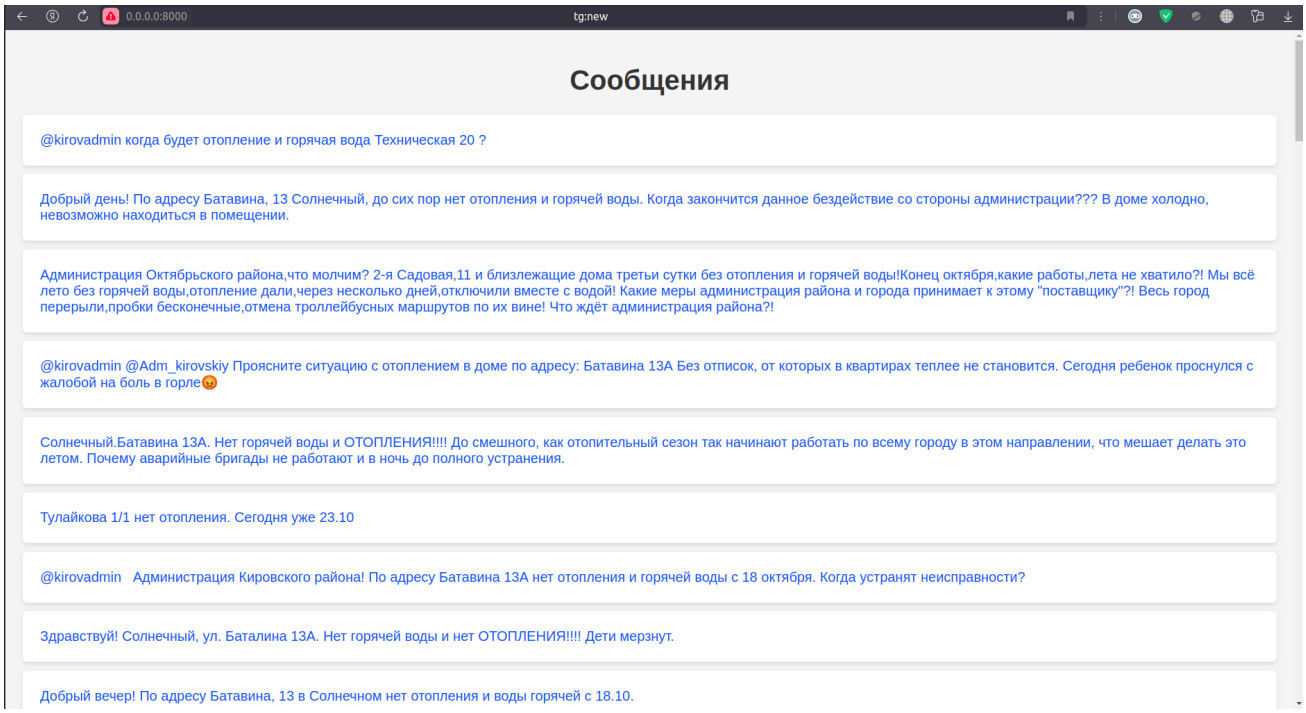

Рисунок 2 – Пример результата запроса новых данных из Telegram

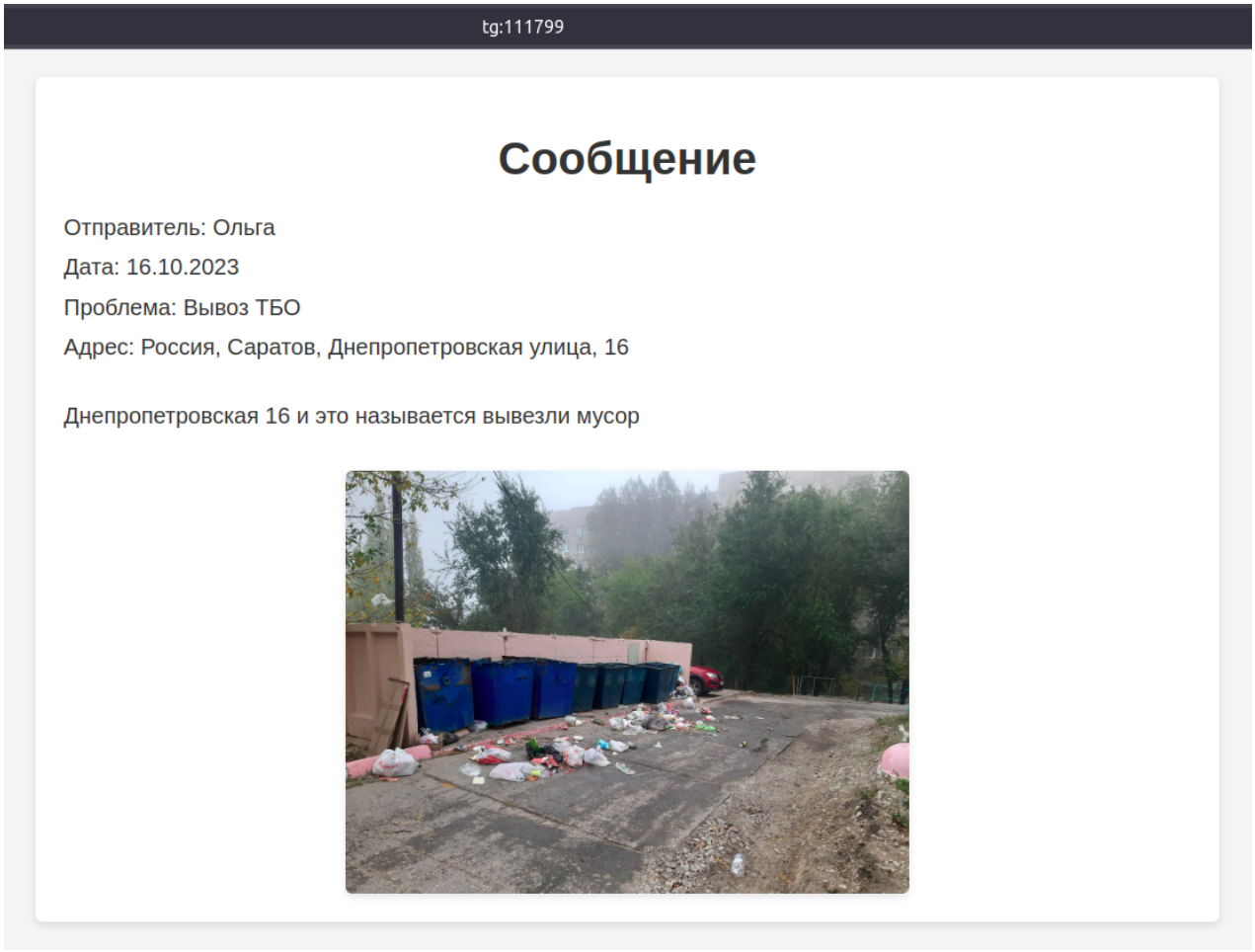

Рисунок 3 – Пример результата запроса конкретного сообщения с изображением

Обновление данных. В данном подразделе представлено описание логи-

ки обработчика основного хоста сервиса. Назначение обработчика заключается в запросе на получение новых записей в базу данных при необходимости и перенаправлении на страницу карты. Он используется для обработки следующих ситуаций: при посещении клиентом главной страницы веб-приложения (в том числе первое посещение сайта), а также при запросе планировщика для ежедневного сбора свежих данных.

Отображение карты. Конечной целью в разработке данной системы является реализация карты города, на которой отображаются все собранные и обработанные данные о жалобах в виде меток. В данном подразделе описываются подготовка данных для карты, обработчик, отвечающий за отображение карты, и скрипт, инициализирующий карту и управляющий метками на ней.

На рисунке 4 приведен пример общего вида карты. Нажатие на карточки на боковой панели фильтрует метки по соответствующей категории. Нажатие на метку раскрывает балун с информацией о конкретной аварии, как представлено на рисунке 5.

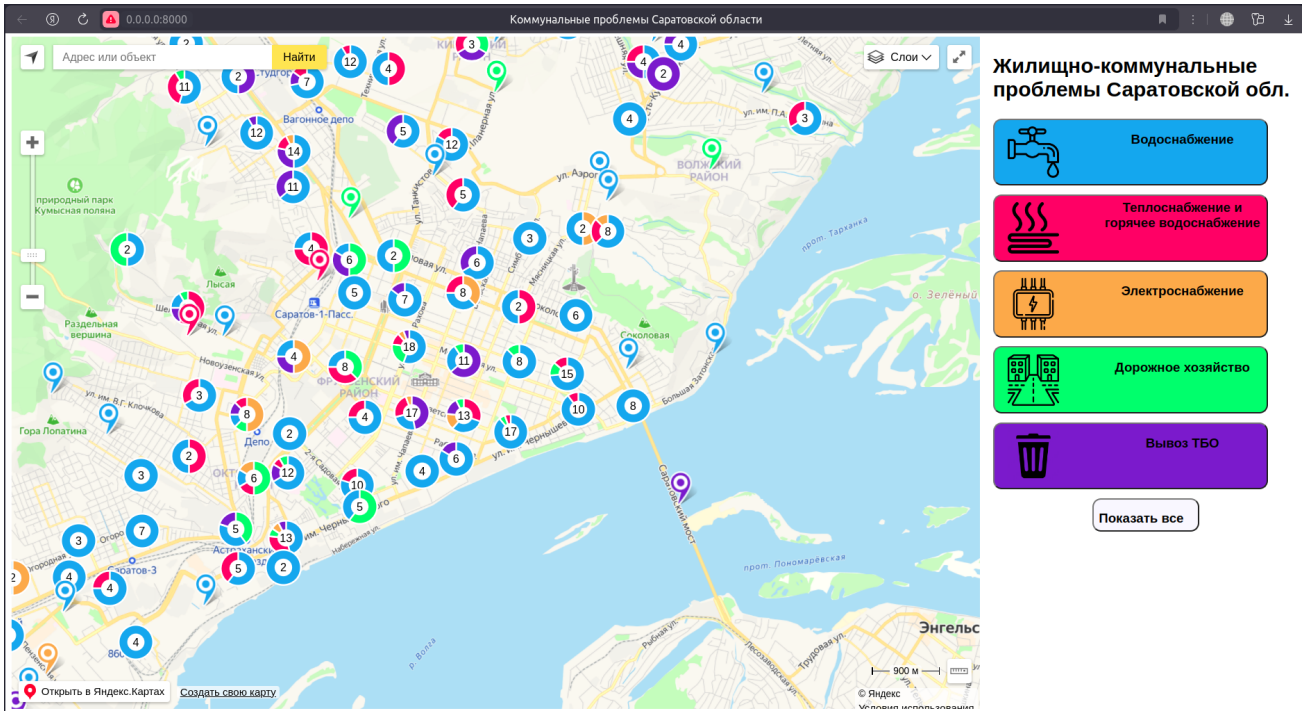

Рисунок 4 – Общий вид карты

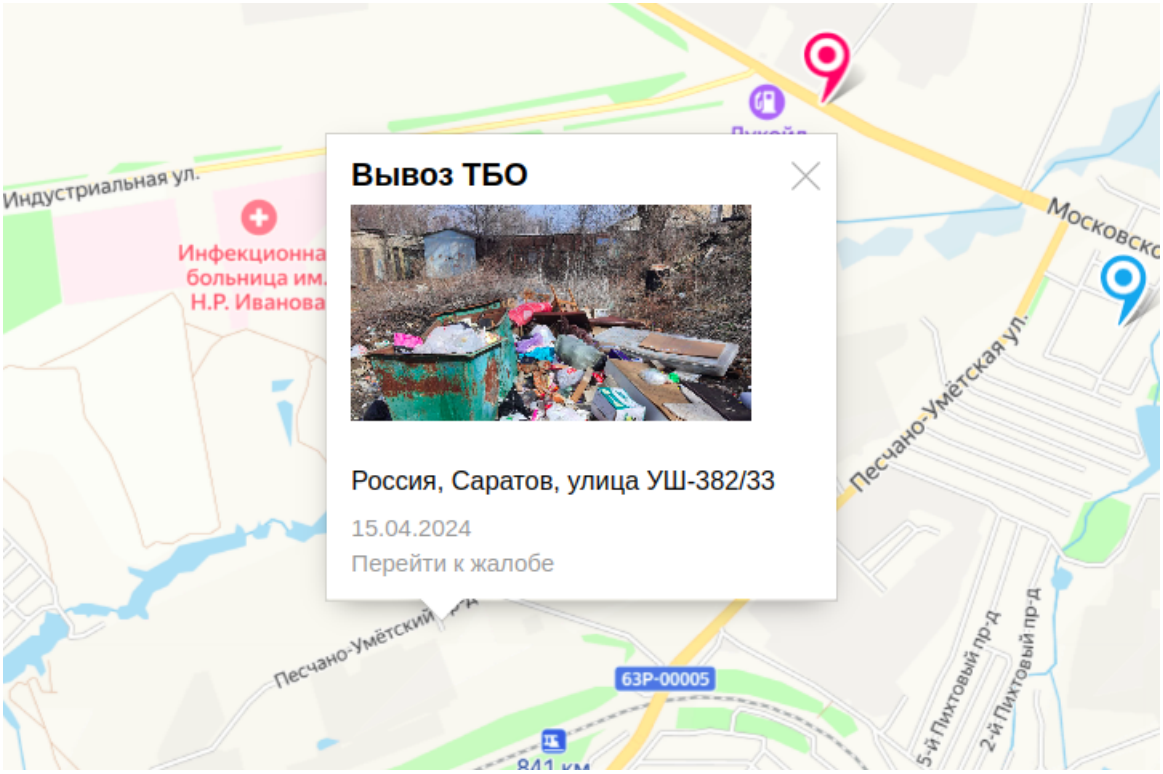

Рисунок 5 – Балун одиночной метки

Контейнеризация системы. В данном разделе представлено описание шагов для контейнеризации разрабатываемой системы. В проекте есть четыре подсистемы, для каждой из которых предназначен собственный контейнер: сервисы для парсинга, обработки текста и серверная часть системы, а также база данных. Здесь представлено описание Dockerfile для каждой подсистемы, Dockerfile-шаблона для FastAPI-сервисов и конфигурационного файла docker-compose.yaml для запуска приложения.

ЗАКЛЮЧЕНИЕ

В данной работе был рассмотрен процесс автоматизированного сбора данных при помощи парсинга, изучены методы обработки естественного языка и средства работы с геоданными. Была разработана и подготовлена для развертывания система, состоящая из нескольких сервисов. Микросервисная структура приложения, а также использование модели MVC позволили сделать систему легко расширяемой. Она включает в себя два независимых API для парсинга и для обработки текста, а также сервис, взаимодействующий с этими API и базой данных.

Разработанная система предоставляет веб-сайт с интерактивной картой, отображающий состояние городских объектов, отдельные страницы для каждой жалобы, а также методы для автоматического сбора новых данных и обработки жалоб населения, получаемых из мессенджера Telegram и сайтов водоснабжающих организаций.

Данный сайт может использоваться людьми, планирующими покупать или снимать жилье, для выбора комфортных жилищных условий, а также органами государственной и муниципальной власти, которые осуществляют контроль над городским хозяйством для повышения качества и эффективности своей работы.