**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Отчет о выполнении лабораторной работы №10**

**по дисциплине**

**«Технологии проектирования программного обеспечения»**

Направление подготовки 01.03.02 Прикладная математика и информатика

Работу выполнил студент группы 4ИТ/2 И.А. Молчанов

(подпись)

Работу проверил доц. каф. ИТ, к.т.н., доц. А.Н. Полетайкин

(подпись)

Краснодар

2023

**ВВЕДЕНИЕ**

**Тема**: Документирование и развертывание ПС

**Цель**: Освоение методики документирования ПС.

**Задание**

1. Собрать документированные материалы, разработанные при выполнении лабораторных работ №1–9 в единый документ с разбивкой на разделы. Темы лабораторных работ обозначить названиями разделов. В разделы не включать цель, задание и выводы из отчетов о выполнении лабораторных работ.

2. Дополнительно отдельным разделом 10 описать назначение, технические характеристики, принцип работы и меры безопасности при эксплуатации ПС.

3. Дополнительно отдельным разделом 11 составить руководство пользователя.

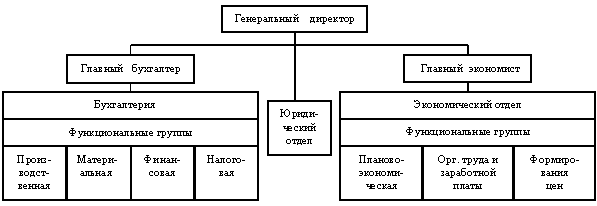
**Индивидуальная тема:** программное средство для анализа новостных сообщений на фейк.

1. **Характеристика объекта информатизации**

Объектом информатизации является система анализа новостных сообщений на признак фейка.

Данная система предназначена для автоматического сбора и анализа новостей на фейк. Этот процесс включает в себя сбор новостей, фильтрацию и предварительный анализ, анализ контекста, а также сбор соответствующей отчетности о нём.

Для примера рассмотрим компанию, предлагающую юридические услуги новостным агрегаторам «Ветров и партнёры». Компания занимается юридическим сопровождением новостных агрегаторов.



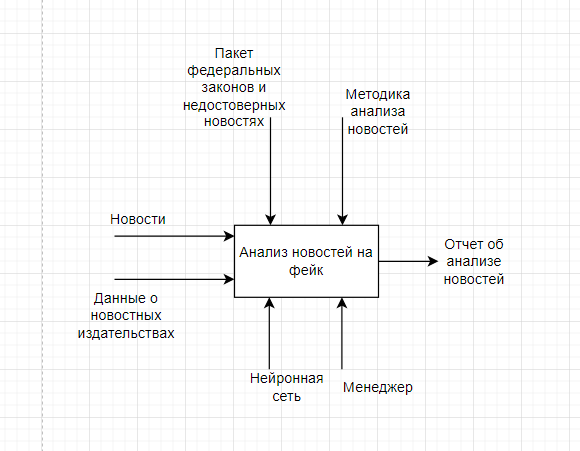
1. **Процесс информатизации**
   1. **Наименование процесса информатизации**

Процесс информатизации *–* процесс анализа новостных сообщений на признак фейка для заказчиков (новостных агрегаторов) в специализированной юридической компании «Ветров и партнеры». Он включает в себя сбор новостных сообщений, их анализ на признак фейка и составление отчета по проведенному анализу.

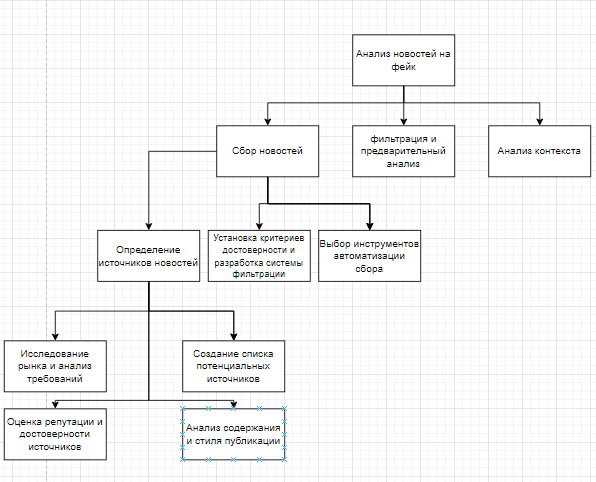
* 1. **Состав действующих лиц**

*Менеджер* *–* сотрудник, который использует систему для анализа новостных сообщений и определения, являются ли они фейком.

* 1. **«Черный ящик»**



* 1. **Декомпозиция бизнес-процесса**



* 1. **Основные задачи бизнес-процесса**

1. Обнаружение фейковых новостей.
2. Проверка достоверности информации.
3. Оценка рисков и потенциального влияния.
4. Разработка автоматизированной системы.
5. Обратная связь и обучение.
   1. **Диаграмма вариантов использования (use case diagram)**



* 1. **Описание входных и выходных потоков данных**

Вся информация хранится в документах.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Наименование и назначение потока | Форма представления | Обработчик (кто обрабатывает) | Корреспондент (откуда) | Характеристики обработки | |
| Трудозатраты | Периодичность |
| 1 | Новостные сообщения | Документ (таблица) | Менеджер | Клиент (новостной агрегатор) | 2 ч/ч | Ежедневно или по запросу |
| 2 | Данные о новостных издательствах | Документ (таблица) | Менеджер | Клиент (новостной агрегатор) | 2 ч/ч | Ежедневно или по запросу |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Наименование и назначение потока | Форма представления | Обработчик (кто обрабатывает) | Корреспондент (откуда) | Характеристики обработки | |
| Трудозатраты | Периодичность |
| 1 | Составление отчета | Документ | Менеджер | Архив | 1 ч/ч | Ежедневно или по запросу |

* 1. **Правила обработки информации и возможность ограничений**

1) Должно быть ограниченное количество записей в документе.

* 1. **Нормативно-справочная документация**

1) Пакет федеральных законов о недостоверных новостях *–* два закона, которыми в России в 2019 году был установлен запрет на публикацию недостоверной общественно значимой информации, распространяемой под видом правдивых сообщений.

2) Методика анализа новостей *–* это подробное описание всех условий и операций, которые обеспечивают правильность, воспроизводимость и другие регламентированные характеристики результатов анализа.

1. **Основные операции**
   1. **Схема решения задачи в ручном режиме**

По декомпозиции процесса видно, что при потребности проанализировать новостные тексты, менеджер компании должен в ручном режиме собрать все необходимые тексты, изучить пакет федеральных законов, составить методику анализа новостей, провести сам анализ и составить по нему отчет.

* 1. **Обоснование автоматизации схемы решения задачи**

Автоматизация задачи анализа фейковых новостей позволяет обрабатывать большие объемы информации за малое время. Это особенно важно для менеджеров, которые должны оперативно принимать решения на основе актуальных данных. Автоматизированная система может обработать и проанализировать большой поток новостей гораздо быстрее, чем это сделает человек. Человеческий фактор может быть подвержен предвзятости и субъективности при анализе новостей. Автоматизированная система основана на алгоритмах и моделях машинного обучения, которые работают на основе объективных критериев и правил, исключая субъективное влияние. Это позволяет получать более нейтральную и надежную оценку новостей.

* 1. **Процесс автоматизации**

В программу менеджер предоставляет файл с собранными новостными сообщениями, запускает анализ и в конце собирает отчет по выполненному анализу.

* 1. **Описание ранее разработанных систем (подсистем)**
  2. **RapidMiner**

RapidMiner – это инструмент для анализа данных с графическим интерфейсом, который также может быть использован для анализа новостных сообщений. Он предоставляет набор операторов и функциональности, которые позволяют создавать конвейеры обработки данных для классификации новостей и извлечения полезной информации из текстовых данных.

В RapidMiner есть возможность создать процесс анализа данных, состоящий из нескольких этапов. В контексте анализа новостей, первым шагом может быть импорт данных из источника новостных статей, такого как база данных или файл. RapidMiner поддерживает различные форматы данных, включая CSV, Excel, JSON и другие.

После импорта данных можно применить операторы предварительной обработки текста для очистки и подготовки новостных статей для дальнейшего анализа. Например, реализовано использование операторов для удаления стоп-слов, токенизации (разделения текста на отдельные слова или токены), стемминга (приведение слов к их основной форме) и удаления ненужных символов или пунктуации.

Затем можно использовать операторы (Рис 1.1) для извлечения признаков из текста новостей. Это может включать операторы для извлечения ключевых слов, фраз, н-грамм (последовательностей из нескольких слов) или других характеристик, которые могут быть полезны для классификации или анализа новостей.

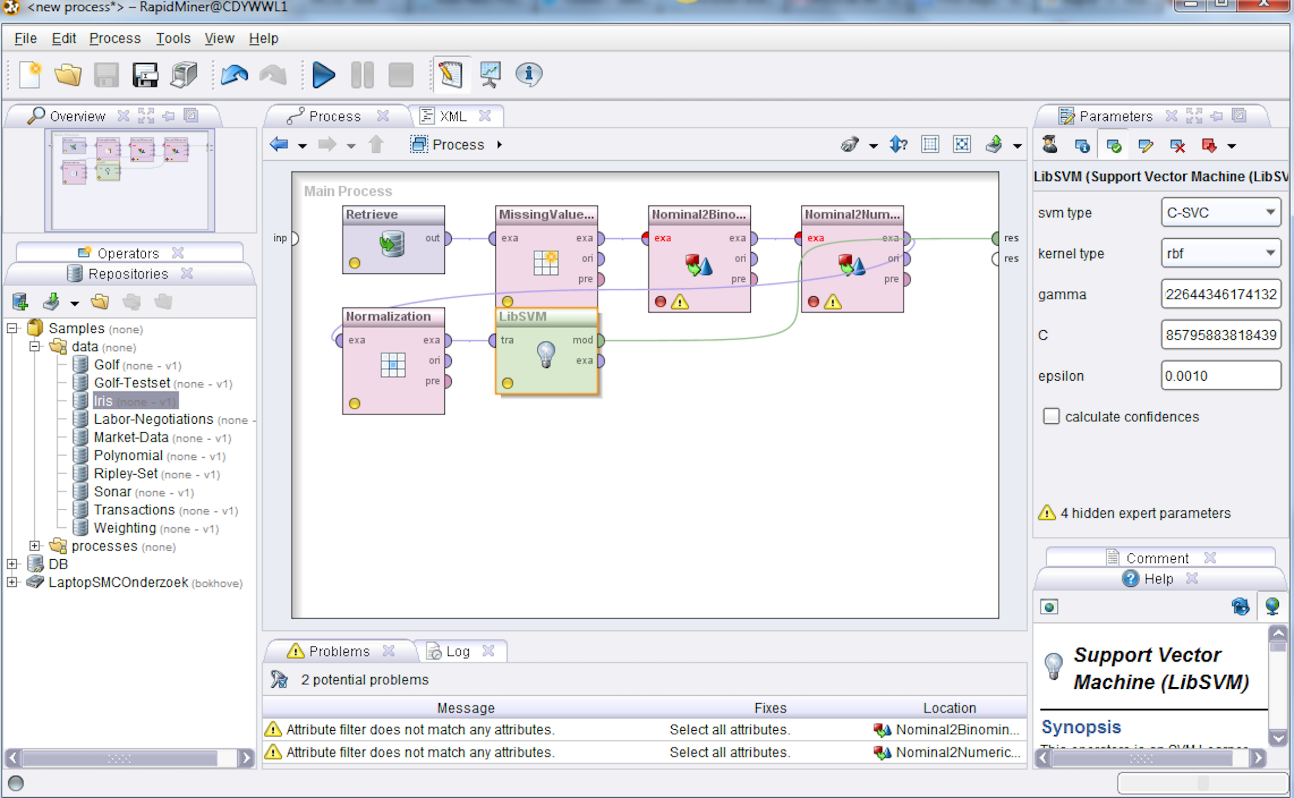


Рис 1.1 – Настройка операторов

После этого, если необходимо, можно применить алгоритмы классификации, доступные в RapidMiner, для классификации новостей. Это может быть метод наивного Байеса, машинное обучение на основе опорных векторов (SVM), решающие деревья или другие алгоритмы, которые предоставляются в RapidMiner. Доступна настройка параметров алгоритма и оценка его производительность с помощью операторов валидации и оценки модели.

Кроме классификации, RapidMiner также предоставляет возможности для анализа тональности текста, выявления тематик, кластеризации новостей и других видов анализа. Есть возможность использовать соответствующие операторы и методы, чтобы получить дополнительную информацию о ваших новостных данных.

Главное преимущество RapidMiner заключается в его графическом интерфейсе, который позволяет пользователю визуально создавать и настраивать процессы анализа данных без необходимости написания кода. Однако, RapidMiner также поддерживает использование скриптов на языке R или Python, если требуется более гибкий и расширенный анализ новостей. Интерфейс RapidMiner выглядит следующим образом (Рис. 1.2):

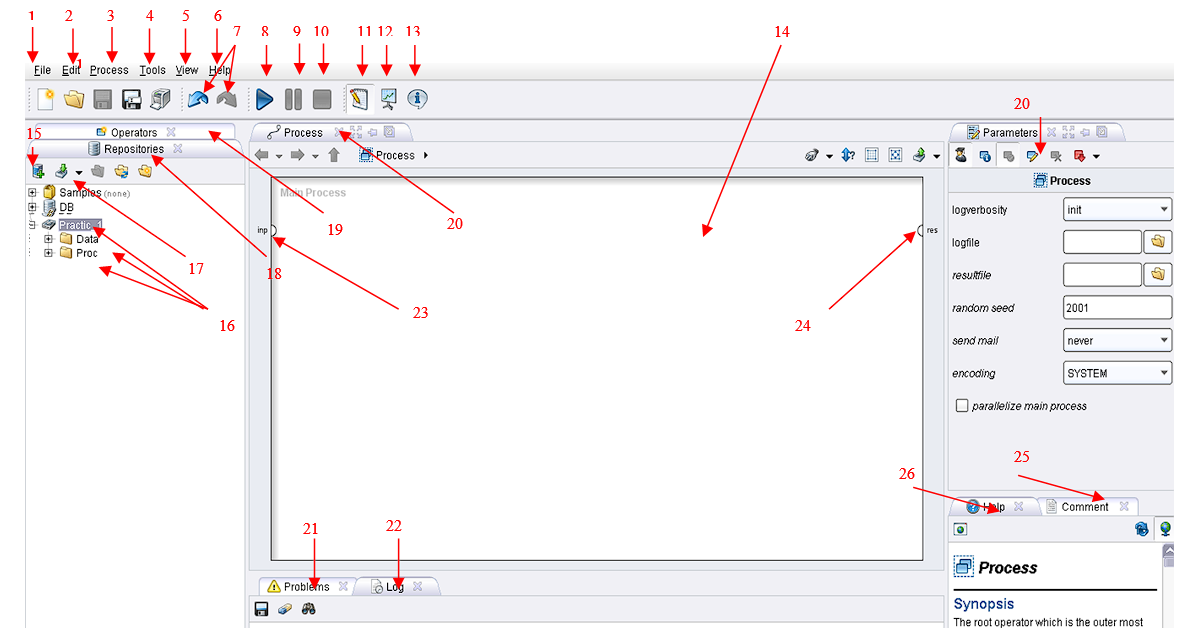


Рис 1.2 – Интерфейс RapidMiner

1. Пункт меню файл;
2. пункт меню вставка;
3. пункт меню процессы;
4. пункт меню инструменты;
5. пункт меню вид;
6. пункт меню помощь;
7. отмена и возврат действия;
8. запуск процесса;
9. кнопка приостановления процесса;
10. кнопка остановки процесса;
11. кнопка перехода на рабочий лист создания процесса;
12. кнопка перехода на рабочий лист результатов процесса;
13. кнопка перехода на приветственную страницу RapidMiner;
14. рабочее поле для создания процесса или модели;
15. кнопка для создания нового репозитория или загрузки уже созданного;
16. созданные папки для хранения данных (Data) и моделей, процессов (Proc) в репозитории (Practic\_1);
17. кнопка для импорта данных;
18. панель репозитория;
19. панель операторов;
20. панель параметры (меню будет видоизменятся в зависимости от активного элемента);
21. панель конфликтов. В нем будет появляться информация о возникших проблемах при выполнении процесса;
22. журнал загрузок;
23. узел входа данных (начало процесса);
24. узел выхода данных (завершение процесса);
25. меню комментариев;
26. меню помощи.

RapidMiner имеет множество графических представлений результата анализа текста (Рис 1.3-1.5):

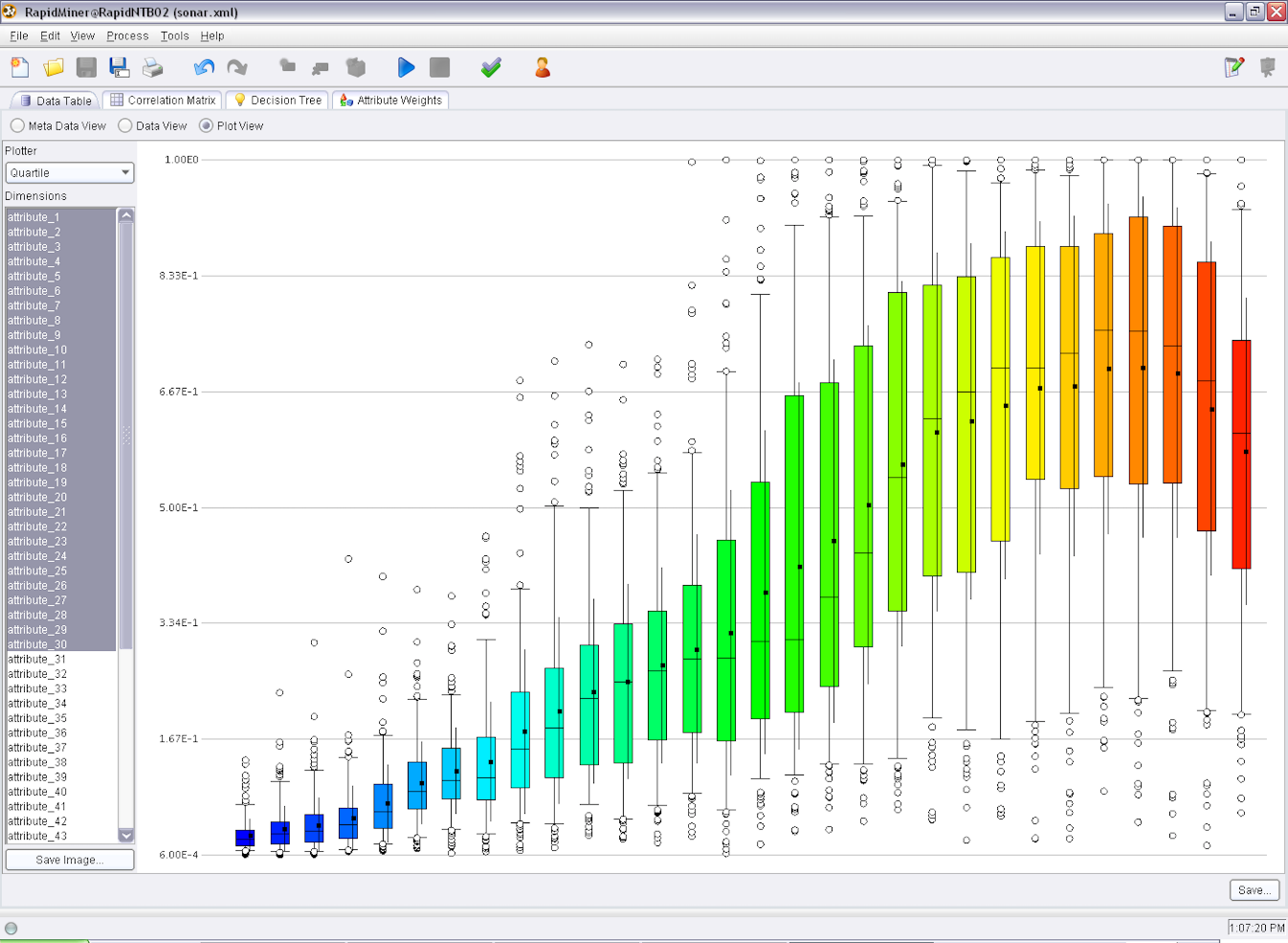


Рис 1.3 – Представление результата

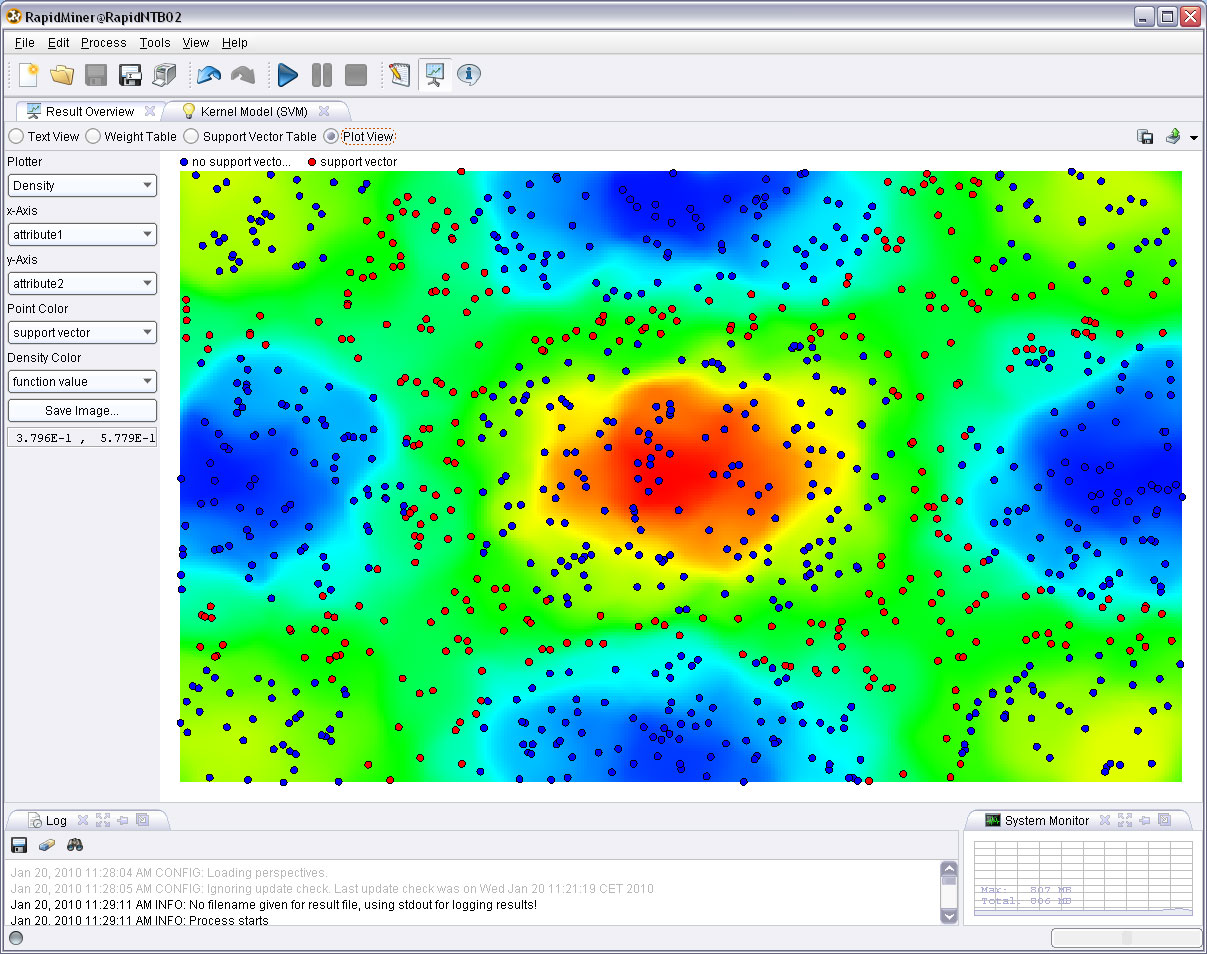


Рис 1.4 – Представление результата

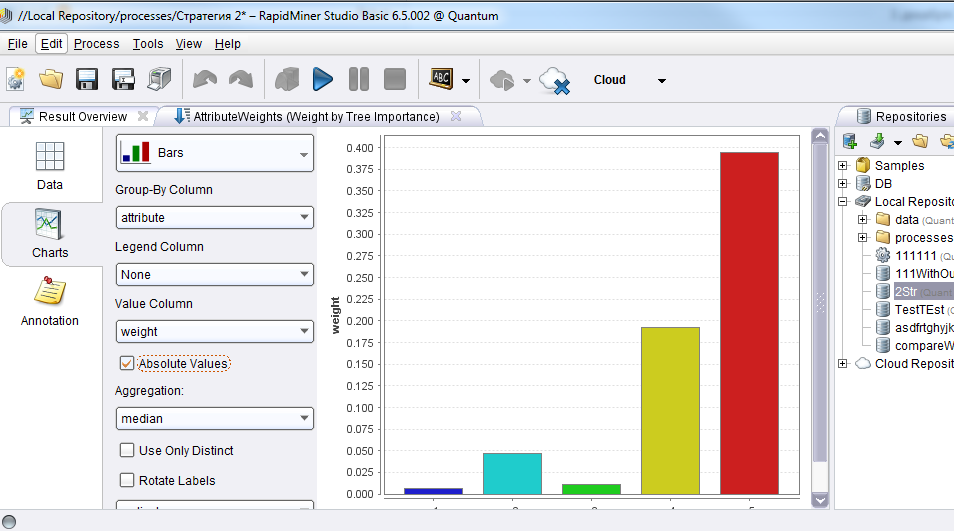


Рис 1.5 – Представление результата

В целом, RapidMiner предоставляет широкие возможности для анализа новостных сообщений и может быть полезным инструментом для создания и применения моделей классификации, извлечения признаков и других видов анализа в контексте новостных данных.

**1.2 MonkeyLearn**

MonkeyLearn – это платформа для автоматического обучения и классификации текста. Она предоставляет различные инструменты и API, которые позволяют пользователям создавать и развертывать модели машинного обучения для обработки текстовых данных, включая анализ новостных сообщений. Веб-интерфейс приложения MonkeyLearn (Рис 1.2.1):

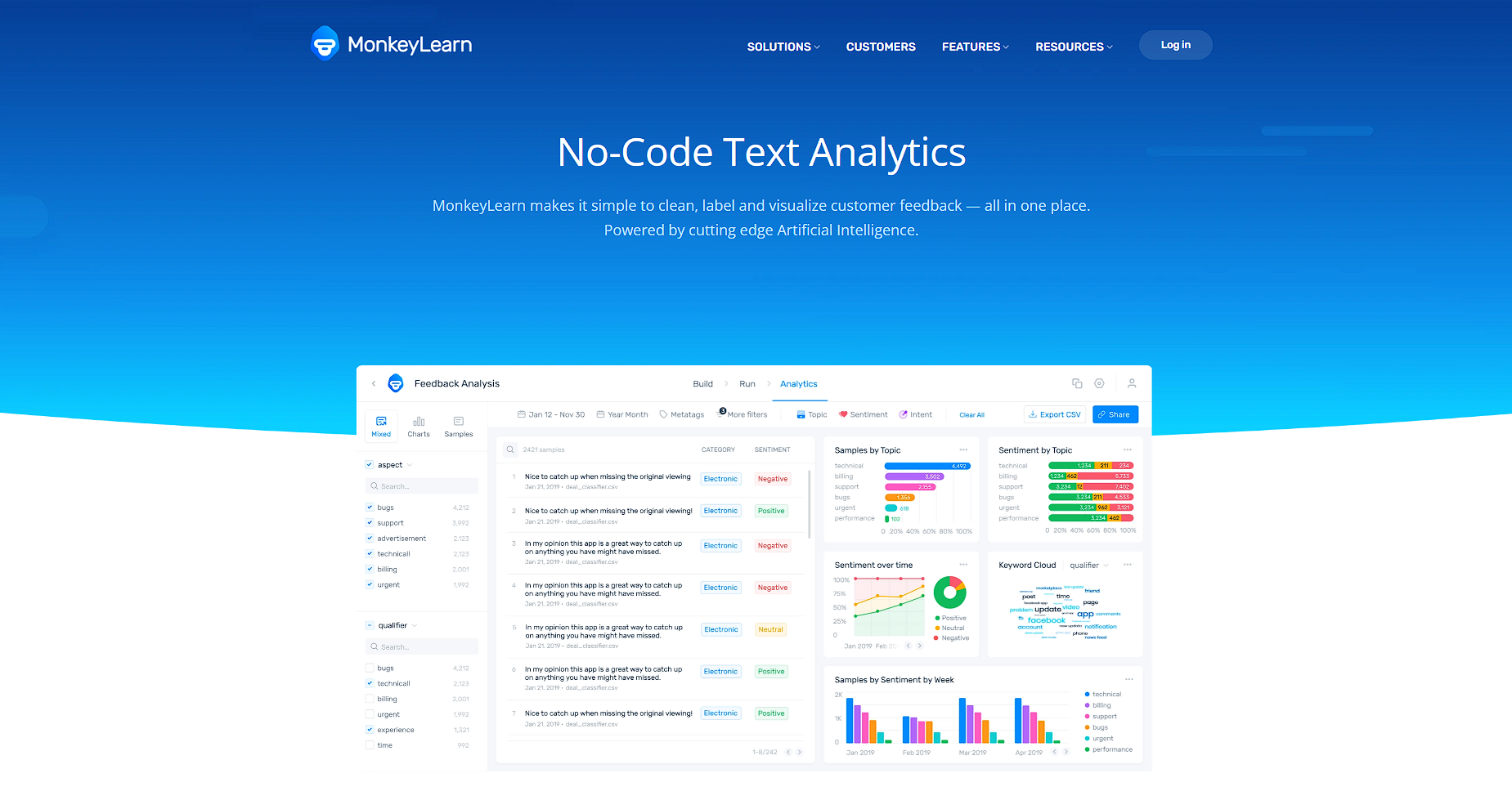


Рис 1.2.1 – Веб-интерфейс приложения MonkeyLearn

Основными функциональностями MonkeyLearn являются классификация текста, извлечение информации и обработка тональности. Эти функции позволяют обрабатывать текстовые данные, выделять ключевую информацию и анализировать эмоциональную окраску текста.

Для использования MonkeyLearn, вы можете создать модель классификации, которая будет принимать текстовые данные и присваивать им соответствующие метки или категории. В случае анализа новостей, вы можете создать модель, которая будет классифицировать новости по различным тематикам, например, спорт, финансы, политика и т.д. Для создания такой модели вам потребуются размеченные данные, то есть набор текстовых данных с уже известными метками категорий.

MonkeyLearn также предоставляет готовые модели и классификаторы, которые можно использовать без необходимости создания собственной модели. Вы можете использовать предварительно обученные модели, которые уже обучены на больших объемах данных и готовы к использованию в анализе новостей.

Помимо классификации, MonkeyLearn также предоставляет возможности для извлечения информации из текста. Это может включать извлечение ключевых фраз, сущностей, дат, местоположений и других типов информации, которая может быть полезна при анализе новостей.

Еще одна важная функция MonkeyLearn – это анализ тональности текста. С помощью этой функции вы можете определить эмоциональную окраску текста, например, положительную, отрицательную или нейтральную, что может быть полезно при анализе тональности новостных сообщений.

На сайте реализовано графическое представление результатов (Рис 1.2.2):

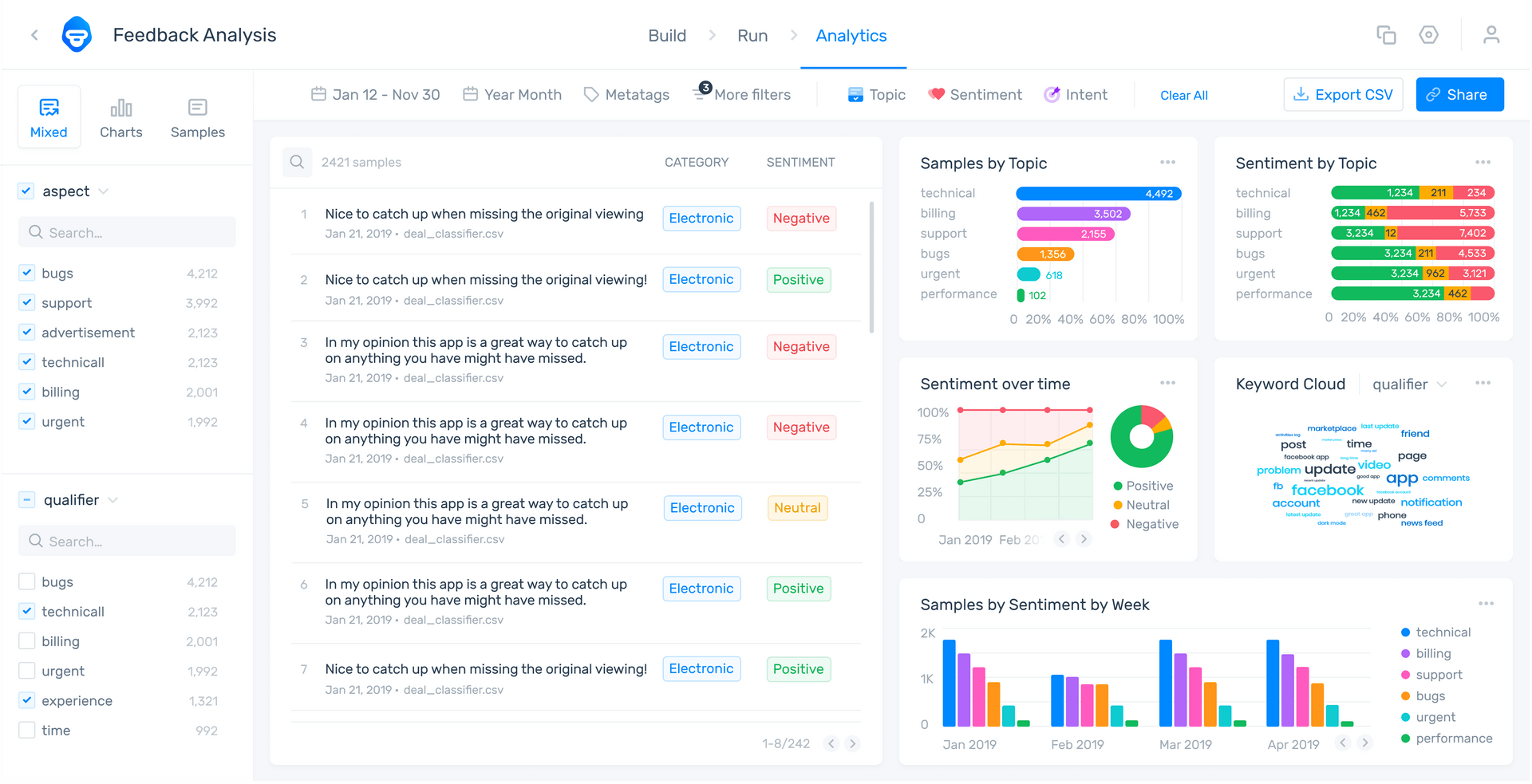


Рис. 1.2.2 – Графическое представление результатов анализа

MonkeyLearn предоставляет простой и интуитивно понятный интерфейс, который позволяет пользователям создавать и настраивать модели, загружать данные, обучать модели и получать результаты анализа. Кроме того, MonkeyLearn предлагает API, который позволяет интегрировать его функциональность в собственные приложения и системы.

* 1. **Сравнение систем**

Сравнение описанных ранее компьютерных разработок по одинаковым критериям представлено в таблице 1:

|  |  |  |
| --- | --- | --- |
| **Параметр** | **RapidMiner** | **MonkeyLearn** |
| Назначение | Инструмент для анализа данных и создания конвейеров обработки данных, включая анализ новостных сообщений. | Платформа для автоматического обучения и классификации текста, включая анализ новостных сообщений. |
| Гибкость | Предоставляет гибкие возможности для создания и настройки процессов анализа данных через графический интерфейс и скрипты на языке R/Python. | Предоставляет готовые модели и классификаторы, а также API для интеграции функциональности в приложения, с ограниченной возможностью настраивать модели. |
| Функ-ционал | Поддерживает различные операторы для предварительной обработки текста, извлечения признаков, классификации, анализа тональности и других видов анализа данных. | Предоставляет функционал для классификации текста, извлечения информации и анализа тональности, с возможностью использования предварительно обученных моделей или создания собственных. |
| Стоимость | Существует бесплатная версия с ограниченным функционалом. Платные планы доступны с различными функциями и уровнями поддержки. | Предлагает платные планы с различными функциями и уровнями использования. Бесплатный доступ ограничен по функциональности и количеству запросов. |
| Безопас-ность | Предоставляет возможность управления доступом и шифрования данных, а также соответствует нормам безопасности данных. | Обеспечивает безопасность данных с помощью шифрования и других мер безопасности, соответствует нормам GDPR и имеет меры контроля доступа. |
| Дизайн | Имеет графический интерфейс, который позволяет визуально создавать и настраивать процессы анализа данных без необходимости написания кода. | Имеет интуитивно понятный интерфейс, который удобен для использования и конфигурации моделей, а также предоставляет API для интеграции. |
| Пользовате-льский интерфейс | Имеет удобный и интуитивно понятный графический интерфейс, который облегчает создание и настройку процессов анализа данных. | Имеет простой и понятный интерфейс, который позволяет пользователям легко загружать данные, создавать и применять модели классификации. |
| Техничес-кая поддержка | Предлагает различные уровни технической поддержки для платных пользователей, включая документацию, форумы и обратную связь. | Предоставляет техническую поддержку в зависимости от выбранного плана, включая документацию, API-референции и техническую поддержку по электронной почте. |
| Системные требования | Работает на различных платформах (Windows, macOS, Linux) и требует установки на локальную машину или сервер. | Относительно менее требовательный продукт за счёт универсальной и удобной веб-версии. |

Таблица 1 – Сравнение «RapidMiner» и «MonkeyLearn»

* 1. **Назначение подсистемы и цели её создания**

Назначение системы анализа новостных сообщений на признак фейка заключается в определении достоверности информации, представленной в новостных статьях или сообщениях. Основная цель создания такой системы - борьба с распространением дезинформации и фейковых новостей.

Система анализа новостных сообщений на признак фейка имеет несколько целей:

1. Выявление фейковых новостей: система помогает автоматически обнаруживать новостные сообщения, содержащие ложную или искаженную информацию. Она анализирует различные аспекты сообщения, такие как источник, стиль письма, фактическую информацию и сравнивает ее с известными достоверными источниками данных. Это позволяет идентифицировать потенциальные фейковые новости и предупреждать пользователей о возможных манипуляциях информацией.
2. Предотвращение распространения дезинформации: система помогает предотвратить распространение фейковых новостей, предупреждая пользователей об их недостоверности. Это может быть особенно полезно в социальных сетях и онлайн-платформах, где фейковые новости могут быстро распространяться и влиять на общественное мнение. Путем предоставления пользователю информации о недостоверности определенной новости система помогает повысить осведомленность и критическое мышление.
3. Поддержка журналистов и факт-чекеров: система анализа новостных сообщений на признак фейка может быть полезной для журналистов и факт-чекеров, облегчая их работу по проверке достоверности информации. Автоматический анализ может помочь выявить потенциальные фейковые новости и предложить дополнительные ресурсы для проверки фактов, что помогает сократить время и усилия, затрачиваемые на ручную проверку каждого сообщения.
4. Улучшение информационной грамотности: система также может способствовать повышению информационной грамотности пользователей, помогая им развивать критическое мышление и умение анализировать достоверность новостей. Предоставление информации о фейковых новостях и методах их идентификации может помочь пользователям развить навыки отбора достоверной информации и более осознанно оценивать новости.

* 1. **Определение структуры программы и состава функциональных задач**

Задачи, программную реализацию которых предполагается осуществить, представлены в таблице 1.

|  |  |
| --- | --- |
| **Задача** | **Описание** |
| Сбор данных | Сбор новостных сообщений из различных источников, таких как новостные веб-сайты, социальные сети, блоги. |
| Предварительная обработка данных | Выполнение предварительной обработки данных, которая включает в себя очистку текста от лишних символов, удаление стоп-слов, приведение текста к нормализованному виду. |
| Анализ и классификация | Основной анализ новостных сообщений на признак фейка с помощью нейронной сети. |
| Проверка достоверности источников | Проверка достоверности источников новостей. Здесь может быть использована база данных достоверных и недостоверных источников. |
| Вывод результатов | Отображение результатов анализа пользователю. |

Таблица 1 – Структура программы и состав её функциональных задач

* 1. **Функциональные требования**

**3.1 Требования к входным и выходным данным**

**3.1.1. Требования к задаче «Сбор данных»**

Источники, из которых будут собираться новостные сообщения: новостные веб-сайты, социальные сети, блоги, форумы. Требуется составить список релевантных источников, которые будут использоваться для сбора данных.

**3.1.2. Требования к задаче «Предварительная обработка данных»**

Разработка процедур для удаления или корректировки неточностей, ошибок. Например, удаление дубликатов, заполнение пропущенных значений, исправление ошибок формата, проведение лемматизации и нормализации, чтобы гарантировать качество данных.

**3.1.3. Требования к задаче «Анализ и классификация»**

Необходимо определить значимые признаки, которые будут использоваться для классификации данных. Это может включать выбор и фиксацию признаков, создание новых признаков на основе имеющихся данных и другие методы инженерии признаков, чтобы максимизировать информативность и разделяющую способность признаков. Среди таких методов могут использоваться: Bag of Words (мешок слов), токенизация, метод наивного Байеса. Также произвести процесс обучения нейронной сети на обучающих данных. Необходимо провести: разделение данных на обучающую и тестовую выборки, настройку параметров модели, использование методов кросс-валидации для оценки качества модели.

**3.1.4. Требования к задаче «Проверка достоверности источников»**

Необходимо создать механизм для сбора информации об источниках данных. Это может включать сбор метаданных, таких как автор, организация, дата публикации и другие сведения, а также описание источника, его репутацию и достоверность. Требуется обеспечить возможность получения и хранения этой информации.

**3.1.5. Требования к задаче «Вывод результатов»**

Необходимо использовать один из способов представления результатов, например, в виде текстового отчета, графиков, таблиц, диаграмм, интерактивных визуализаций. Требуется выбрать наиболее подходящий формат, который будет удобен и понятен пользователям.

**3.2 Требования к программной реализации задач**

Для обеспечения корректной работы и сохранения целостности данных, информация о новостных сообщениях и их атрибутах должна быть храниться в базе данных. База данных должна обеспечивать эффективное хранение и доступ к данным. Пользовательский интерфейс программы должен быть интуитивно понятным и удобным в использовании. Код программы должен быть оптимизирован для обеспечения эффективной работы. Информационные объекты, используемые в программе, должны быть понятными для пользователей. Например, атрибуты новостных сообщений и факторы, используемые для анализа фейковости, должны иметь понятные и описательные наименования.

* 1. **Построение модели требований в нотации UML**

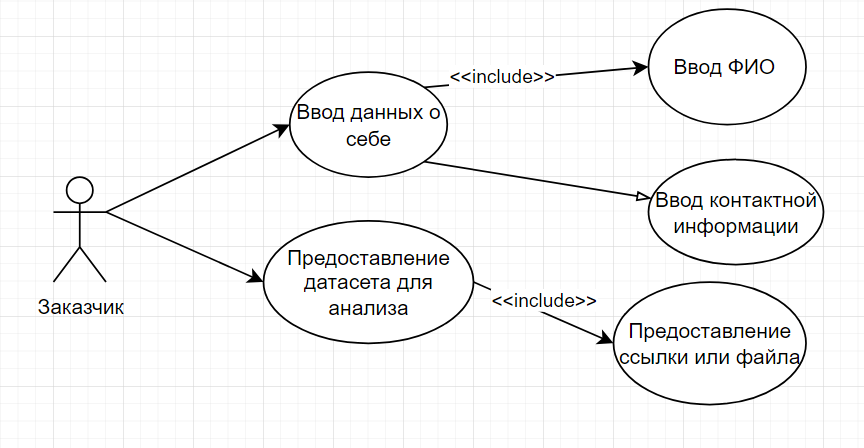


Рисунок 1 – UML-заказчик

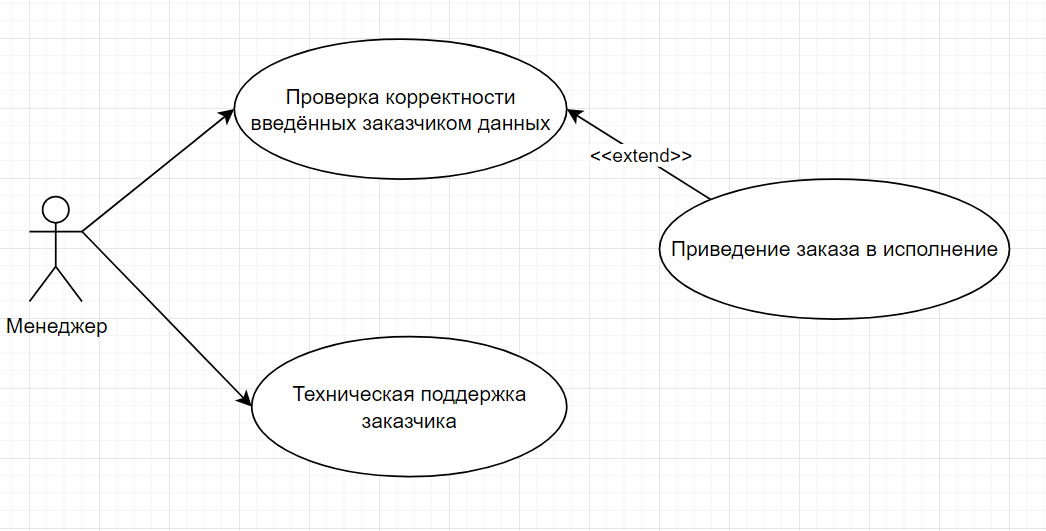


Рисунок 2 – UML-менеджер

* 1. **Требования к базе данных и к инструментальному программному обеспечению**

**5.1. Требования к информационному обеспечению**

• должна отсутствовать дублированная информация;

• должна быть обеспечена возможность ввода информации;

• должна быть защита от несанкционированного доступа к данным;

• должна быть обеспечена конфиденциальность информации;

• должна быть возможность получения данных с помощью языка запросов высокого уровня, без использования прикладных программ;

• БД должна иметь возможность неограниченно использоваться.

**5.2. Требования к инструментальному программному обеспечению**

Ниже будут представлены требования к системе управления базой данных (СУБД), к средству разработки программ (IDE), средствам автоматизированного проектирования ПО

• язык программирования: Python 3.10;

• выбрать среду разработки: PyCharm, JupyterNotebook, GoogleCollab;

• выбрать СУБД: MySQL, PostgreSQL.

* 1. **Нефункциональные требования**

1. *Защищенность –* система должна обеспечена защитой данных от вредоносных действий, таких как уничтожение, искажение или намеренное подделывание.
2. *Надежность –* система должна сохранять значения всех параметров в пределах установленных значений со временем. Важно, чтобы сбои в работе системы не приводили к опасным последствиям.
3. *Производительность* – система должна выполнять определенный объем работы за единицу времени.
4. *Удобный интерфейс –* интерфейс должен быть удобным, интуитивно понятным и продуманным, чтобы обеспечить простоту использования программного обеспечения. Дизайн не должен пугать клиентов излишней сложностью или изощренностью.
5. *Живучесть –* система должна сохранять или быстро восстанавливать способность к выполнению своих функций.
6. **Модель чёрный ящик IDEF0**

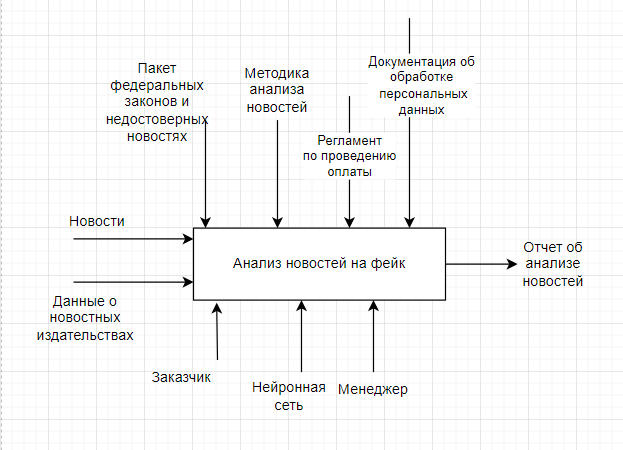


Рисунок 1 – Чёрный ящик IDEF0

1. **Диаграмма декомпозиции А0 на дочерние подпроцессы**

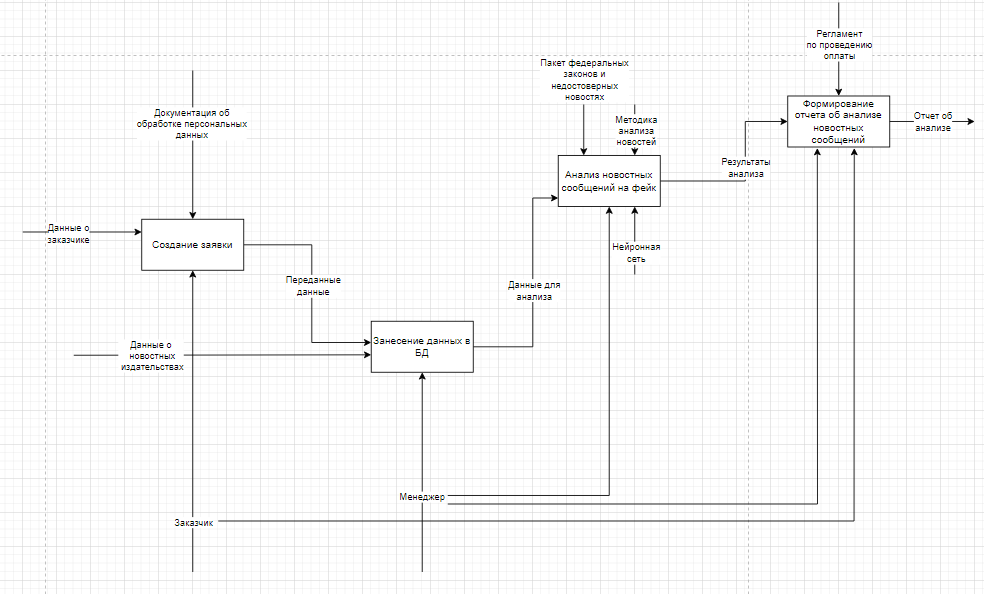


Рисунок 2 – Диаграмма декомпозиции А0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Источник** | **Тип источника** | **Приёмник** | **Тип приемника** |
| Данные заказчика | Внешняя граница | Input | Создание заявки | Mechanism |
| Документация об обработке перс. Данных | Внешняя граница | Control | Создание заявки | Mechanism |
| Заказчик | Внешняя граница | Control | Создание заявка, формирование отчёта об анализе | Mechanism |
| Переданные данные | Создание заявки | Mechanism | Занесение данных в БД | Mechanism |
| Менеджер | Внешняя граница | Control | Занесение данных в БД, формирование отчёта об анализе | Mechanism |
| Данные о новостных изд-х | Внешняя граница | Input | Занесение данных в БД | Mechanism |
| Данные для анализа | Занесение данных  в БД | Mechanism | Анализ новостных сообщений на фейк | Mechanism |
| Пакет фед. законов о недост. новостях | Внешняя граница | Control | Анализ новостных сообщений на фейк | Mechanism |
| Методика анализа новостей | Внешняя граница | Control | Анализ новостных сообщений на фейк | Mechanism |
| Нейронная сеть | Внешняя граница | Control | Анализ новостных сообщений на фейк | Mechanism |
| Результаты анализа | Формирование запроса на склад | Mechanism | Формирование отчёта об анализе | Mechanism |
| Регламент по проведению оплаты | Внешняя граница | Control | Формирование отчёта об анализе | Mechanism |
| Отчет об анализе | Формирование отчёта об анализе | Mechanism | Внешняя граница | Output |

Таблица 1 – Описание элементов функциональной модели

1. **Декомпозиция на подзадачи**
   1. **Создание заявки**

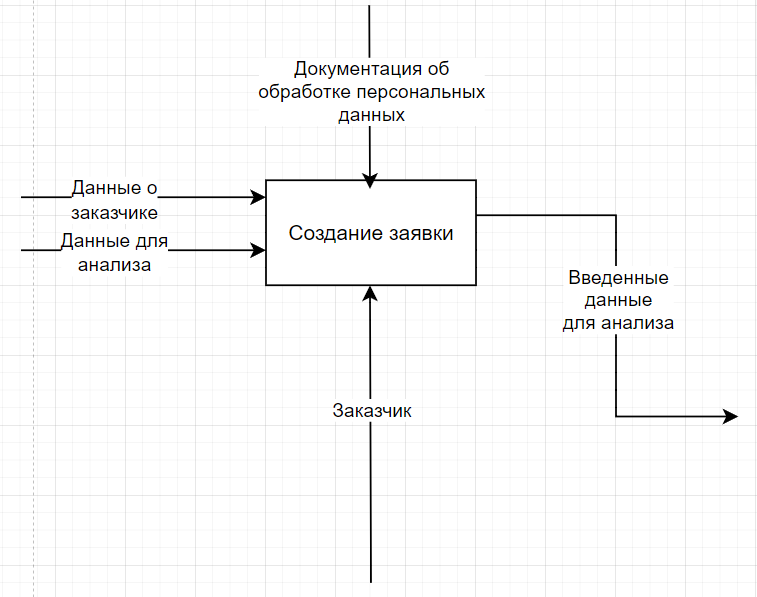
****

Рисунок 3 – Диаграмма декомпозиции «Создание заявки»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Источник** | **Тип источника** | **Приёмник** | **Тип приемника** |
| Данные о клиенте | Внешняя граница | Input | Создание заявки | Mechanism |
| Документация об обработке перс. Данных | Внешняя граница | Control | Создание заявки | Mechanism |
| Данные для анализа | Внешняя граница | Mechanism | Создание заявки | Mechanism |
| Заказчик | Внешняя граница | Control | Создание заявки | Mechanism |
| Введенные данные для анализа | Создание заявки | Mechanism | Внешняя граница | Output |

Таблица 2 – Описание элементов функциональной модели

* 1. **Занесение данных в БД**

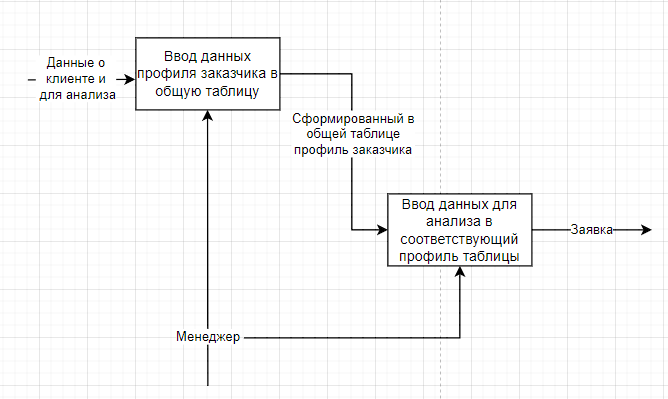


Рисунок 6 – Диаграмма декомпозиции «Обработка данных»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Источник** | **Тип источника** | **Приёмник** | **Тип приемника** |
| Данные о клиенте и для анализа | Внешняя граница | Input | Ввод данных профиля заказчика в общую таблицу | Mechanism |
| Менеджер | Внешняя граница | Control | Ввод данных профиля заказчика в таблицу, ввод данных для анализа в профиль таблицы | Mechanism |
| Сформированный в общей таблице профиль заказчика | Ввод данных профиля в общую таблицу | Mechanism | Ввод данных для анализа в профиль таблицы | Mechanism |
| Заявка | Ввод данных для анализа в профиль таблицы | Mechanism | Внешняя граница | Output |

Таблица 5 – Описание элементов функциональной модели

* 1. **Формирование отчёта о заказе**

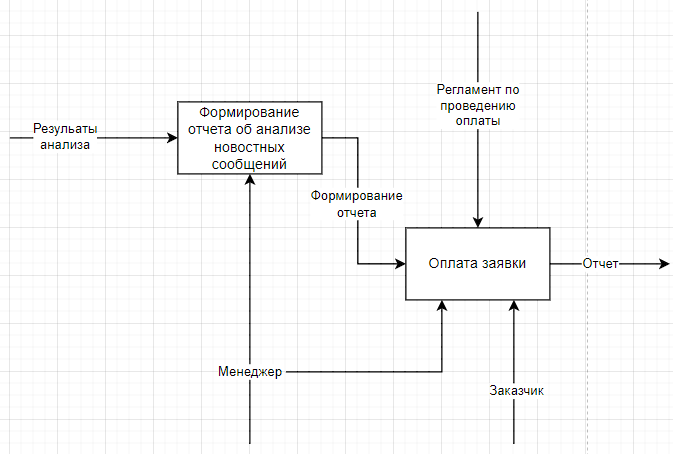
****

Рисунок 7– Диаграмма декомпозиции «Формирование отчёта о заказе»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование** | **Источник** | **Тип источника** | **Приёмник** | **Тип приемника** |
| Результаты анализа | Внешняя граница | Input | Формирование отчета об анализе новостных сообщений | Mechanism |
| Менеджер | Внешняя граница | Control | Формирование отчета об анализе новостных сообщений, оплата заявки | Mechanism |
| Формирование отчёта | Формирование отчета об анализе | Mechanism | Оплата заявки | Mechanism |
| Регламент по проведению оплаты | Внешняя граница | Control | Оплата заявки | Mechanism |
| Заказчик | Внешняя граница | Control | Оплата заявки | Mechanism |
| Отчёт | Оплата заявки | Mechanism | Внешняя граница | Output |

Таблица 6 – Описание элементов функциональной модели

**Абстракции подсистемы**

Выделим основные абстракции подсистемы, разделим их по типам, приведём описание и запишем это в следующую таблицу:

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Абстракция** | **Тип** | **Описание** |
| 1 | Менеджер | Сущность | Пользователь системы, проверяющий корректность заполненных клиентом данных, вносящий данные о клиенте и заказе в БД и занимающийся техподдержкой пользователей сайта |
| 2 | Заказчик | Сущность | Пользователь системы, заполняющий форму заявки, по итогу получающий отчет об анализе |
| 3 | Пакет федеральных законов о недостоверных новостях | Сущность | Положения в законодательстве о недостоверных новостях |
| 4 | Данные о заказчике | Сущность | ФИО, номер телефона и адрес электронной почты клиента |
| 5 | Данные заказа | Сущность | Необходимые клиенту товары, которые он хочет купить в интернет-магазине |
| 6 | Документация об обработке перс. данных | Сущность | Документ, описывающий работу со сведениями о клиенте, которые прямо или косвенно имеют к нему отношение. |
| 7 | Интерфейс менеджера | Интерфейс | Набор инструментов для взаимодействия менеджера с системой |
| 8 | Интерфейс заказчика | Интерфейс | Набор инструментов для взаимодействия заказчика с системой |
| 9 | Добавление  и изменение данных  о заказчиках и заявках в БД | Поведение | Способность добавлять и корректировать данные о заказчиках и их заказах |
| 10 | Добавление и изменение в БД данных для анализа | Поведение | Способность добавлять и корректировать данные для анализа |
| 11 | Формирование отчёта об анализе | Поведение | Способность создавать отчет об анализе новостных сообщений |

Таблица 1 – Абстракция подсистемы

**Классификация абстракций**

|  |  |  |
| --- | --- | --- |
| **№** | **Класс** | **Список абстракций** |
| 1 | Люди | Заказчик, менеджер |
| 2 | Предметы | Отчёт об анализе новостных сообщений. |
| 3 | Концепции | Документация об обработке персональных данных, пакет федеральных законов о недостоверных новостях |
| 4 | События | * Создание заявки; * Добавление/корректировки как данных о заказчиках и их заявках, так и о данных для анализа в БД; * Создание отчета об анализе. |
| 5 | Показатели | Данные о заказчике, данные для анализа |

Таблица 2 – Классификация абстракций

**Абстракции подсистемы и их поведение**

Проанализируем поведение выделенных абстракций. Выделим возможное поведение каждой абстракции в пределах рассматриваемой функции.

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Абстракция** | **Поведение** | **Описание поведения** |
| 1 | Менеджер | Сотрудник, участвующий в работе компании | Подтверждает заявку пользователя и ведёт с ним диалог, проверяет работу алгоритма на различных этапах |
| 2 | Заказчик | Лицо, пользующееся услугами компании | Оформляет заказ в интернет-магазине |
| 3 | Документация об обработке перс. данных | Регулирование отношений между покупателем и компанией | Менеджер вносит полученные данные о заказчике в БД и передает данные для анализа в систему |
| 4 | Добавление и изменение данных о заказчиках и заявках в БД | Работа с данными о заказчиках и заявках в БД | Часть системы, ответственная за добавление, изменение данных о заказчиках и заявках в БД |
| 5 | Добавление и изменение в БД данных для анализа | Работа с данными для анализа в БД | Часть системы, ответственная за добавление и изменение данных для анализа в БД |
| 6 | Формирование отчёта об анализе | Формирование отчёта об анализе | Часть системы, ответственная за составление отчёта |
| 7 | Создание заявки | Ввод всей нужной информации для предоставления услуг | Последовательность шагов, определяющая заполнения в базу данных входящих данных о заказчике и данных для анализа |
| 8 | Данные о заказчике | Подаются в форме заявки, отражают персональную информацию | Собираются менеджером для занесения данных о заказчике в БД |
| 9 | Данные для анализа | Подаются в форме заявки | Собираются менеджером для занесения данных для анализа в БД |

Таблица 3 – Абстракции подсистемы и их поведение

**Диаграмма классов**

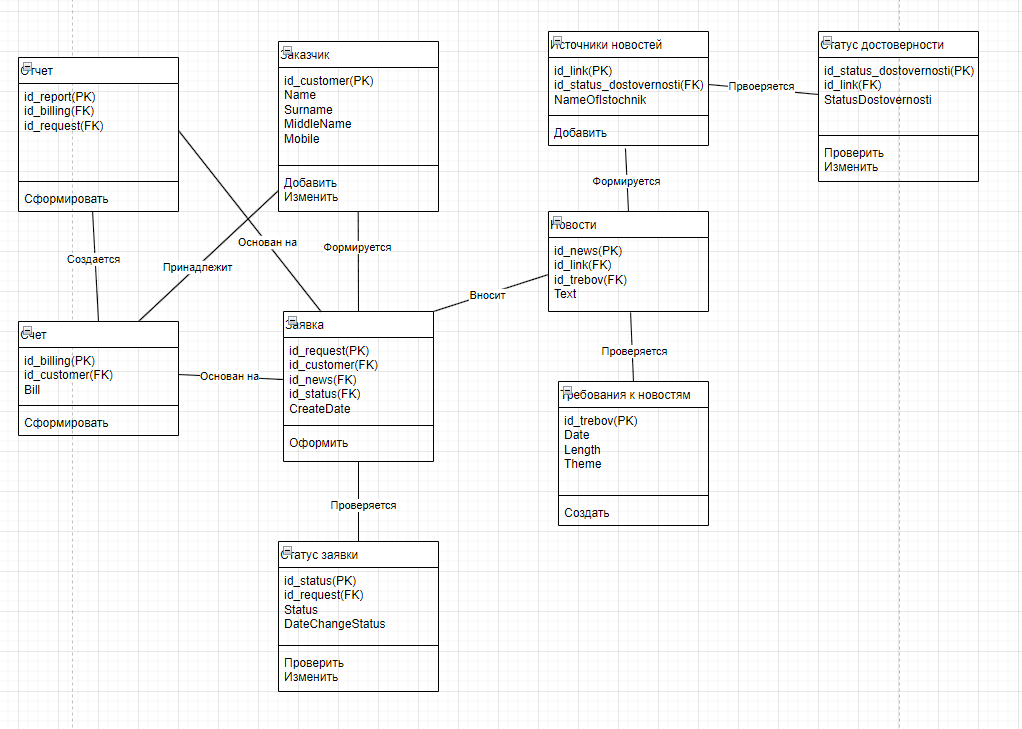


Рисунок 1 – Диаграмма классов

**Диаграмма состояний**

Диаграмма, описывающая состояние заявки на протяжении всего цикла работы системы:

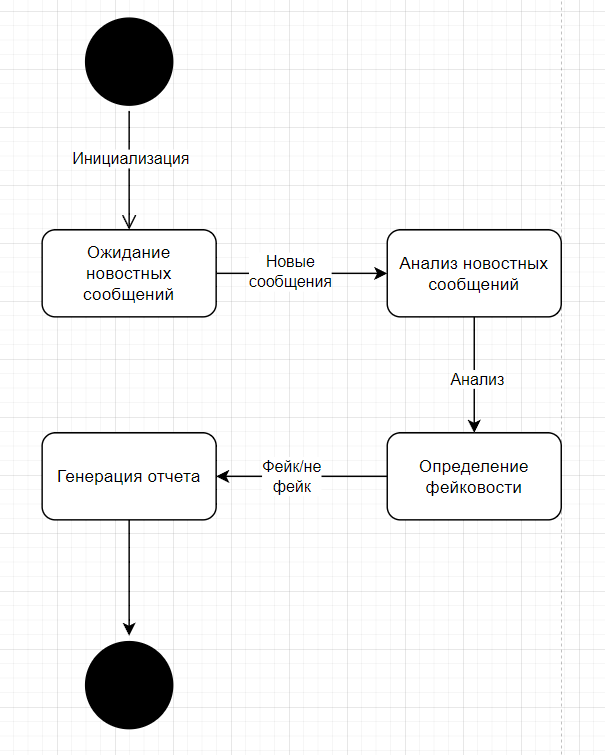
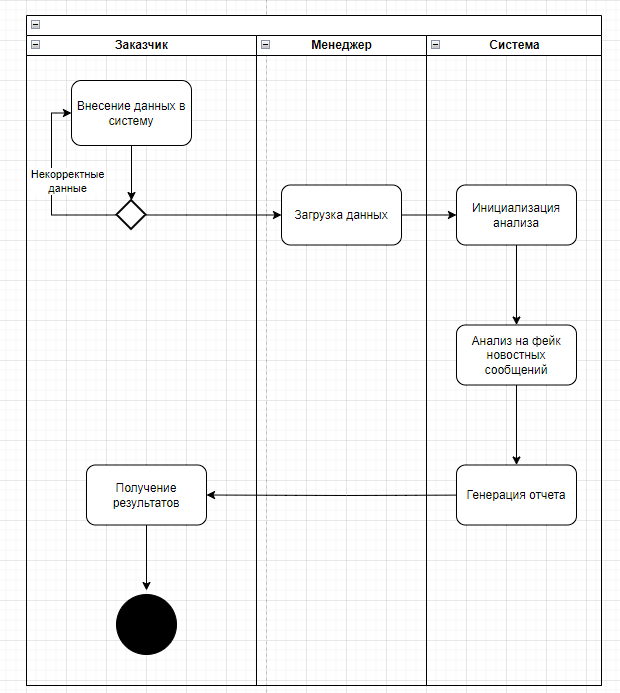


Рисунок 2 – Диаграмма состояний товара

**Activity diagram**

Диаграмма, описывающая процесс обработки заявки.



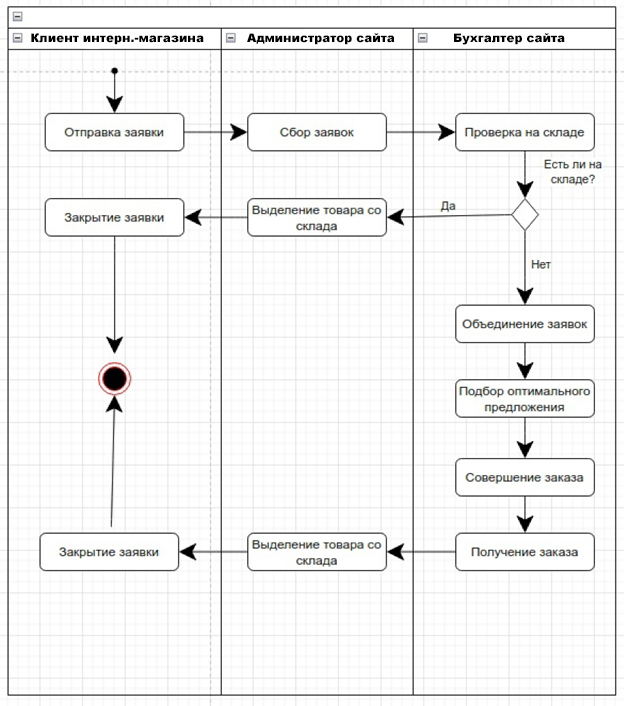


Рисунок 3 – Activity diagram

**6.1 Список объектов для потока событий**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Прецедент** | **Объект** | **Описание объекта** |
| 1 | Внесение данных в систему | Пользователь, заказчик, заявка | Пользователь отправляет заявку о необходимости в анализе новостей |
| 2 | Загрузка данных | Менеджер, заявка | Передача их в БД |
| 3 | Инициализация анализа | Система | Начало анализа |
| 4 | Анализ новостных сообщений | Система | Шаг анализа |
| 5 | Генерация отчета | Система | Сохранение полученных результатов |
| 6 | Получение результатов | Заказчик | Получение результатов заказчиком |

**Sequence diagram**

Диаграмма, описывающая жизненный цикл объекта в рамках прецедента.

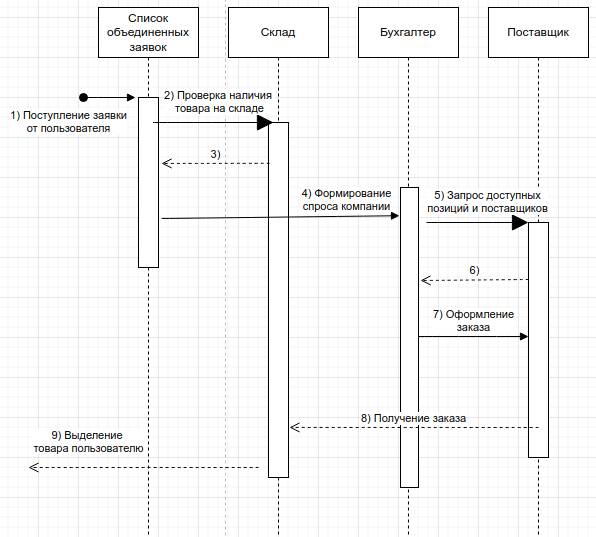


Рисунок 4 – Sequence diagram

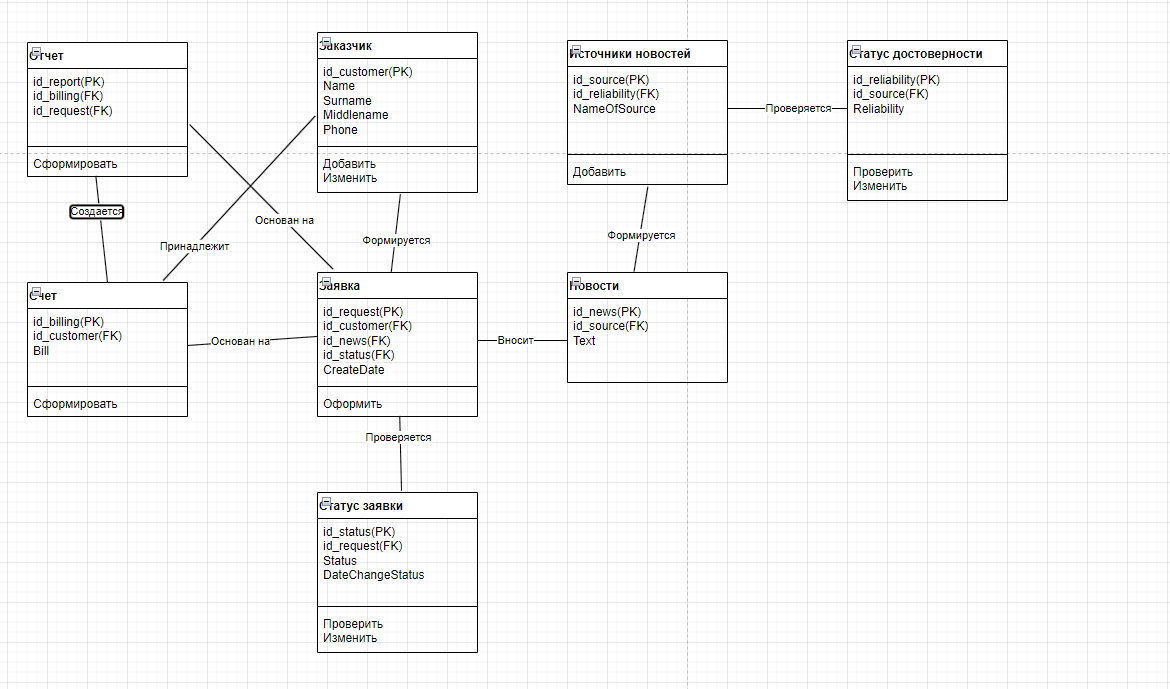
* + - 1. **Идентификация сущности**



Рисунок 1 – Формирование таблиц из сущностей

|  |  |
| --- | --- |
| Сущность | Название таблиц |
| Заказчик | Информация о заказчике (customer) |
| Заявка | Данные заявки (request) |
| Статус заявки | Информация о статусе заявки (request status) |
| Счет | Счет (bill) |
| Отчет | Информация об отчете (report) |
| Статус достоверности | Достоверность источника (reliability) |
| Источники новостей | Информация об источнике новостей (source) |
| Новости | Новостные сообщения (news) |

Таблица 1 – Соответствие название таблиц и сущностей

1. **ER-диаграмма логической модели данных**

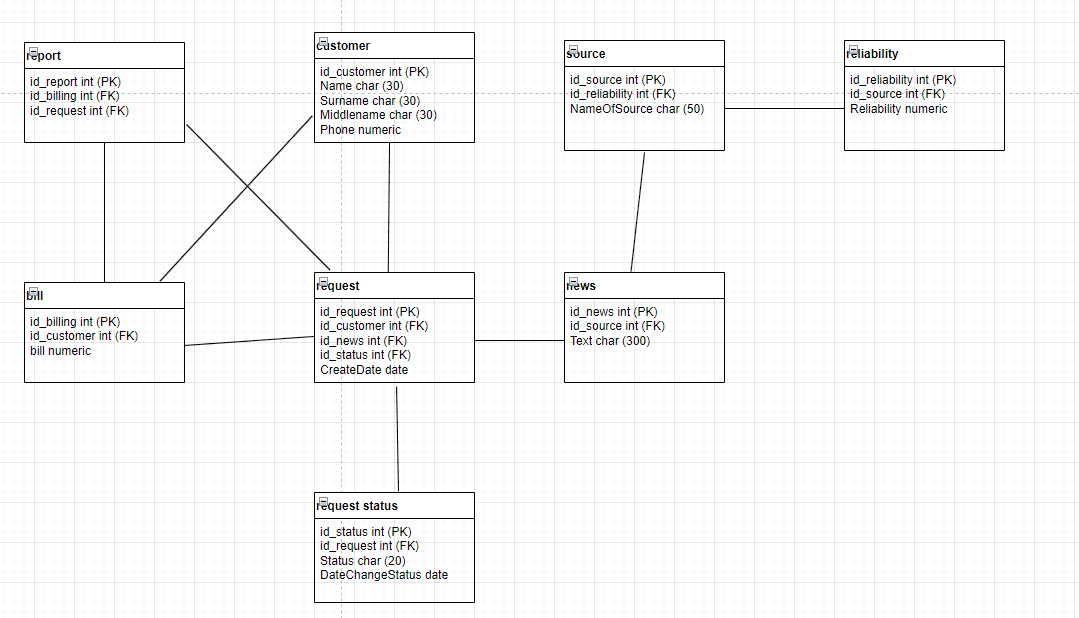


Рисунок 2 – Логическая модель данных

1. **Сравнительный анализ**

Сравнивая диаграмму классов с полученной логической схемой базы данных, можно отметить, что они идентичны. Сравнительный анализ с диаграммой классов показал на верную проектировку будущей БД. Также была проведена нормализация данных.

1. **ER-диаграмма БД**

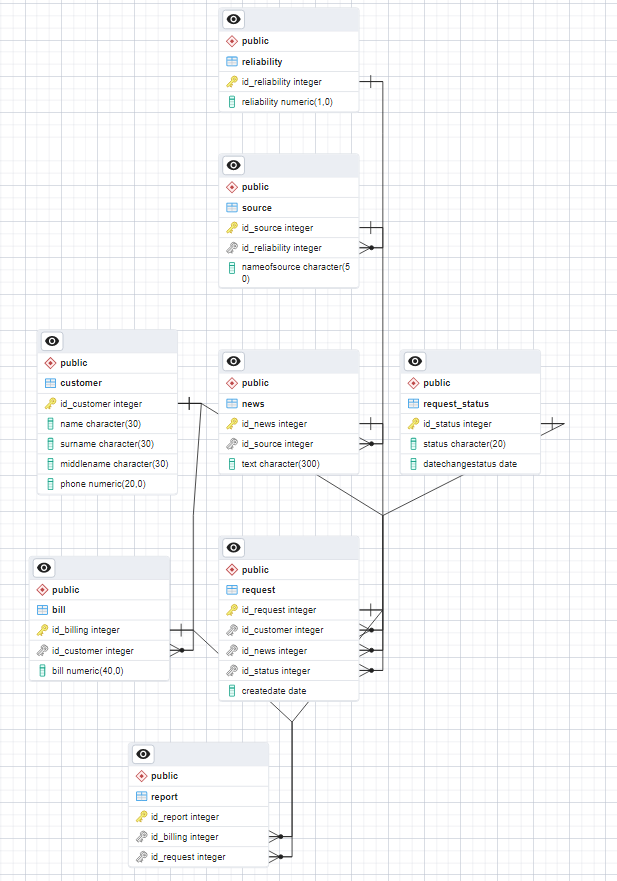


Рисунок 3 – ER-диаграмма

|  |  |  |
| --- | --- | --- |
| Сущность | Атрибуты | Описание |
| Справочные | | |
| 1 Заказчик (customer) | Идентификатор заказчика, имя, фамилия, отчество, номер телефона | Информация о заказчиках и соответствие им своего id |
| 2 Достоверность (reliability) | Идентификатор достоверности, статус достоверности | Информация о достоверности источника |
| 3 Статус заявки (request\_status) | Идентификатор статуса, статус заявки | Информация о статусе заявки |
| Оперативные | | |
| 4 Источники (source) | Идентификатор источника, идентификатор достоверности, название источника | Описывает источник новостных сообщений |
| 5. Новости (news) | Идентификатор новостей, идентификатор источника, текст новостей | Информация о новостях |
| 6 Заявка (request) | Идентификатор заявки, идентификатор новостей, идентификатор заказчика, идентификатор статуса, дата создания заявки | Общая информация о заявке |
| 7 Счет (bill) | Идентификатор счета, идентификатор заявки, номер счета | Информация о счете |
| 8 Отчет (report) | Идентификатор отчета, идентификатор счета, идентификатор заявки | Информация об отчете |

Таблица 3 – сущности физической модели

1. **SQL-код**

CREATE TABLE customer

(

id\_customer serial NOT NULL,

name char(30) NOT NULL,

surname char(30) NOT NULL,

middlename char(30) NOT NULL,

phone numeric(20) NOT NULL,

PRIMARY KEY (id\_customer)

);

CREATE TABLE bill

(

id\_billing serial NOT NULL,

id\_customer serial NOT NULL,

bill numeric(40) NOT NULL,

PRIMARY KEY (id\_billing),

FOREIGN KEY (id\_customer) REFERENCES customer (id\_customer)

);

CREATE TABLE reliability

(

id\_reliability serial NOT NULL,

reliability numeric(1) NOT NULL,

PRIMARY KEY (id\_reliability)

);

CREATE TABLE source

(

id\_source serial NOT NULL,

id\_reliability serial NOT NULL,

nameofsource char(50) NOT NULL,

PRIMARY KEY (id\_source),

FOREIGN KEY (id\_reliability) REFERENCES reliability (id\_reliability)

);

CREATE TABLE news

(

id\_news serial NOT NULL,

id\_source serial NOT NULL,

text char(300) NOT NULL,

PRIMARY KEY (id\_news),

FOREIGN KEY (id\_source) REFERENCES source (id\_source)

);

CREATE TABLE request\_status

(

id\_status serial NOT NULL,

status char(20) NOT NULL,

datechangestatus date NOT NULL,

PRIMARY KEY (id\_status)

);

CREATE TABLE request

(

id\_request serial NOT NULL,

id\_customer serial NOT NULL,

id\_news serial NOT NULL,

id\_status serial NOT NULL,

createdate date NOT NULL,

PRIMARY KEY (id\_request),

FOREIGN KEY (id\_customer) REFERENCES customer (id\_customer),

FOREIGN KEY (id\_news) REFERENCES news (id\_news),

FOREIGN KEY (id\_status) REFERENCES request\_status (id\_status)

);

CREATE TABLE report

(

id\_report serial NOT NULL,

id\_billing serial NOT NULL,

id\_request serial NOT NULL,

PRIMARY KEY (id\_report),

FOREIGN KEY (id\_billing) REFERENCES bill (id\_billing),

FOREIGN KEY (id\_request) REFERENCES request (id\_request)

);

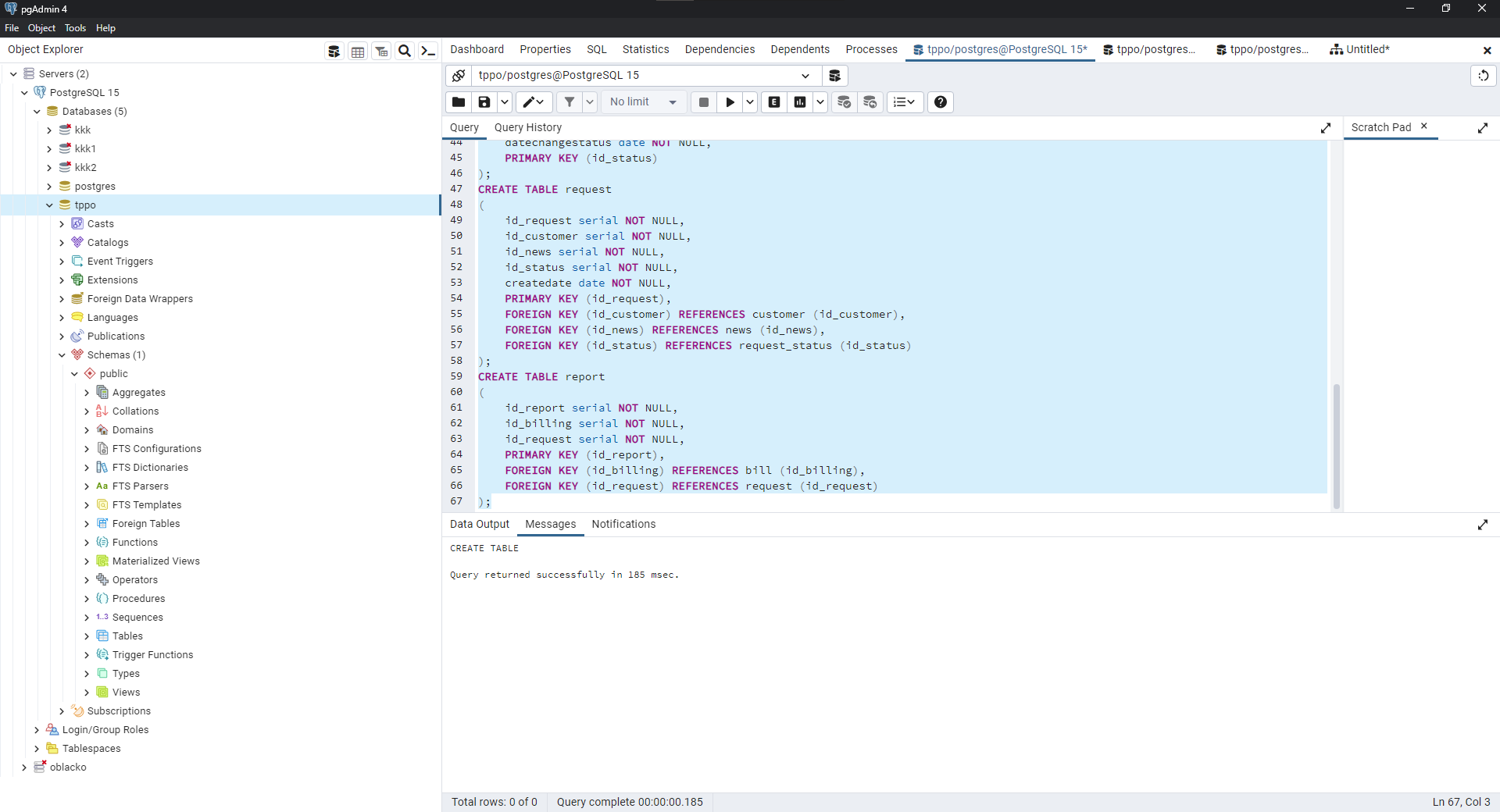


Рисунок 4 – Реализация Базы данных

1. **Выбор СУБД**

В качестве используемой для данной системы СУБД был выбран PostgreSQL — это популярная свободная объектно-реляционная система управления базами данных. Среди преимуществ: поддержка БД неограниченного размера, мощные и надёжные механизмы транзакций и репликации, легкая расширяемость, широкая функциональность.

Также стоит подчеркнуть, что это enterprise решение, которое используется во многих крупных проектах. Поэтому не возникнет проблем с поддержкой и помощью от сообщества. В том числе это хороший опыт для работы с ней в будущем.

1. **Связи между БД**

|  |  |  |
| --- | --- | --- |
| № п/п | Имя таблицы | Описание |
| 1 | Заказчик (customer) | Таблица с данными о заказчике. |
| 2 | Статус достоверности (reliability) | Таблица с данными о достоверности источника. |
| 3 | Источники новостей (source) | Таблица с данными об источниках новостей. |
| 4 | Новости (news) | Таблица с данными о новостях |
| 5 | Заявка (request) | Таблица с заявками |
| 6 | Статус заявки (request\_status) | Таблица с данными о статусе заявки |
| 7 | Счет (bill) | Таблица с информацией о счете |
| 8 | Отчет (report) | Таблица с информацией об отчете |

Таблица 4 Список разработанных таблиц

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Родительская таблица | | Дочерняя таблица | | Тип связи |
| Название | Атрибут | Название | Атрибут |
| reliability | id\_reliability | source | id\_reliability | 1:N |
| source | id\_sorce | news | id\_sorce | 1:N |
| news | id\_news | request | id\_news | 1:N |
| request\_status | id\_status | request | id\_status | 1:N |
| customer | id\_customer | request | id\_customer | 1:N |
| customer | id\_customer | bill | id\_customer | 1:N |
| bill | id\_billing | report | id\_billing | 1:N |
| requests | id\_request | report | id\_request | 1:N |

Таблица 5 - Связи между таблицами БД

* + - 1. **Используемое ПО**
  1. **Инструментальные средства**

Для разработки программной системы был выбран PyCharm, он обеспечивает удобную среду разработки с возможность. Отладки, авто дополнения кода и другими полезными функциями.

Основной язык программирования, на котором написано приложение: Python. Он предоставляет простой синтаксис, богатую стандартную библиотеку и широкие возможности для разработки GUI-приложений.

Были использованы библиотеки tensorflow, которая позволяет создавать структуру нейронной сети, numpy и pandas, позволяющие работать с математическими данными, также psycopg2, которая используется для взаимодействия с базой данных PostgreSQL из Python и фреймворк Flask, позволяющий запускать приложение в веб-пространстве.

Для работы с базами данных выбран PgAdmin4 – это программное обеспечение, которое отлично подходит для PostgreSQL и не имеет аналогов.

Также была использована система контроля версий Git, которая позволяет отслеживать изменения в коде, управлять версиями. Для этого была использована платформа GitHub.

* 1. **Общесистемное ПО**

Приложение может быть запущено на различных операционных системах, таких как Windows, macOS или Linux.

Ubuntu – самый безопасный среди дистрибутивов Линукс. Он менее требователен к ресурсам, более энергоэффективен при работе, бесплатен, доступен большинству ПО, для него выходят бесплатные обновления, у него исходный открытый код, а также более гибкая настраиваемость, возможность изменить любой элемент под себя, есть возможность управления настраиваемыми данными.

* 1. **Специальное ПO**

Модуль базы данных отвечает за хранение и организацию данных, обеспечивая эффективный доступ к ним.

Модуль обработки входных данных отвечает за обработку и анализ входных данных, поступающих в программу.

Модуль выходных данных отвечает за предоставление информации или результатов работы системы пользователю.

**1.4 Структурная схема общесистемного программного обеспечения**

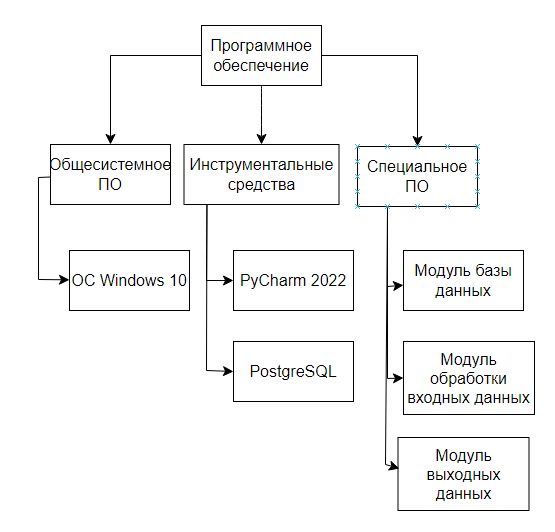


Рисунок 1 - Структурная схема

* + - 1. **Перечень разработанных компонентов приложения**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя | Тип | Описание |
| 1 | index.html | «source» | Реализация программного интерфейса. |
| 2 | form.html | «source» | Реализация интерфейса формы заявки. |
| 3 | app.py | «executable» | Исходный файл программы, из которой загружаются зависимости. |
| 4 | model.py | «source» | Реализация основных функциональных возможностей программы в классах модулей. |
| 5 | prediction\_function.py | «source» | Реализация функции для предсказания фейковости текста. |
| 6 | tensorflow | «library» | Библиотека для создания модели нейронной сети. |
| 7 | numpy | «library» | Библиотека для работы с математическими данными. |
| 8 | psycopg2 | «library» | Библиотека для работы с базой данных PostgreSQL. |
| 9 | customer | «table» | Таблица с данными заказчика |
| 10 | request | «table» | Таблица с данными о заявке |
| 11 | news | «table» | Таблица с данными о новостях |

Таблица 1 – Компоненты приложения

* + - 1. **Диаграмма компонентов**

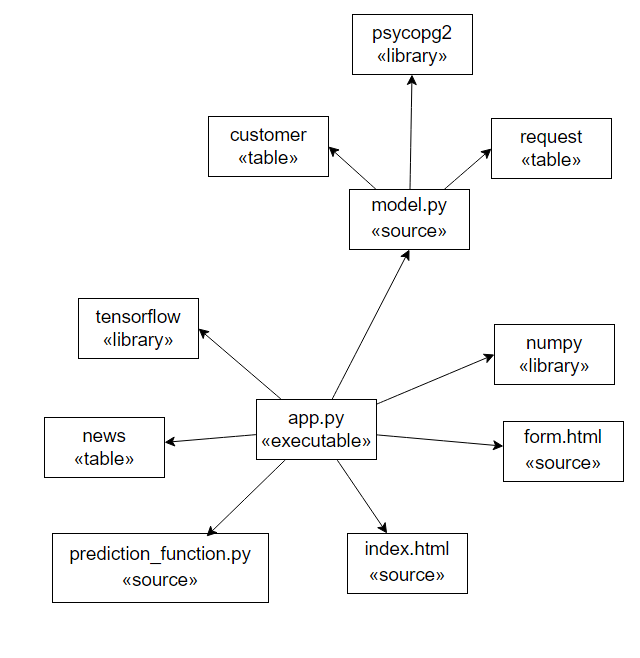


Рисунок 2 Диаграмма компонентов

* + - 1. **Физические элементы ПС и диаграмма развёртывания**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя | Тип | Описание |
| 1 | Компьютер | «processor» | Рабочая станция для взаимодействия с системой |

Таблица 2 - Физические элементы ПС

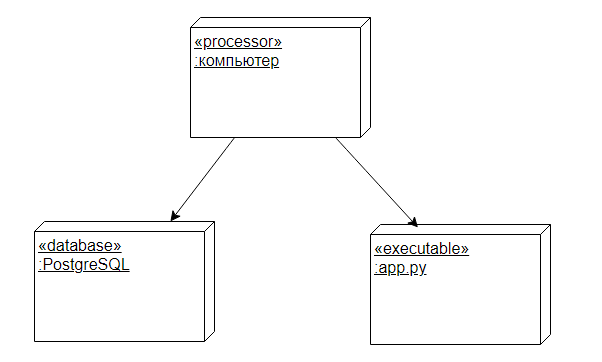


Рисунок 4 Диаграмма развёртывания

* + - 1. **Экранные формы компонентов приложения**

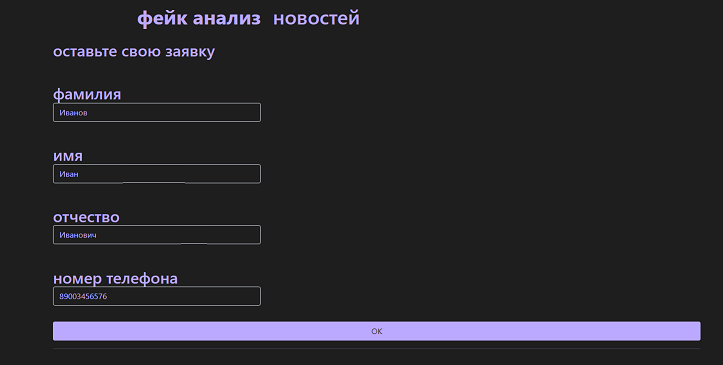


Рисунок 1 – окно формы заявки

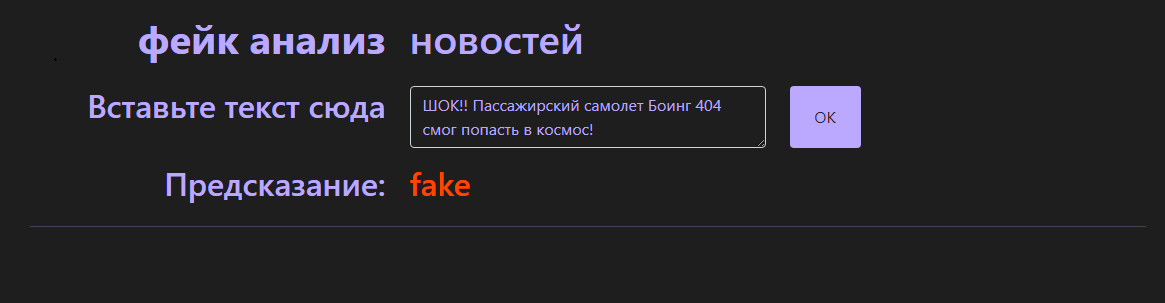


Рисунок 2 – окно при фейковой новости

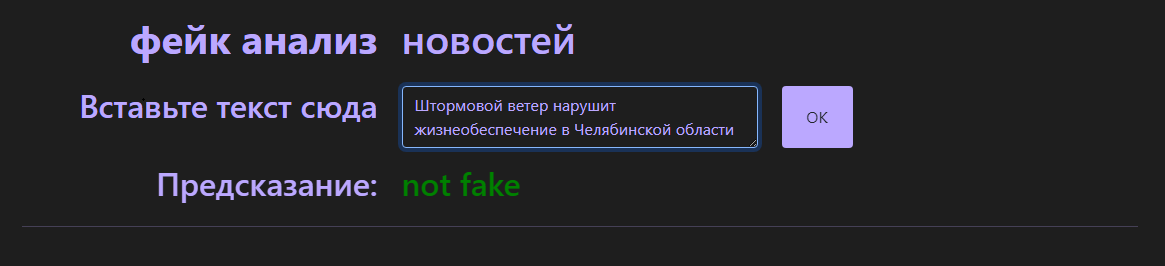


Рисунок 3 – окно при правдивой новости

* + - 1. **Сложность кода приложения**

В разработанном приложении:

* 38 функций и методов;
* 36 переменных;
* 33 оператора;

**1. Системное (ручное) пользовательское тестирование ПС.**

Системное тестирование – это тестирование всей системы в целом, как правило, через ее пользовательский интерфейс. Следует проверить соответствует ли разработанная ПС заявленным требованиям.

Система адаптивна под различные разрешения экрана, интерфейс простой и незапутанный.

Для корректного тестирования необходимо провести тест со стороны пользователя.

**1.1 Тест №1 (Пользователь)**

Состоит из проверки функционала пользователя в следующих действиях:

1. Зайти на страницу формы заявки
2. Заполнить поля некорректными данными
3. Проверить валидацию данных
4. Зайти на страницу формы ввода текста
5. Ввести текст для анализа
6. Проверить валидацию данных

Ожидается, что все элементы интерфейса корректно работают. Этот тест успешно пройден.

Скриншоты срабатывания некоторых пунктов проверки:

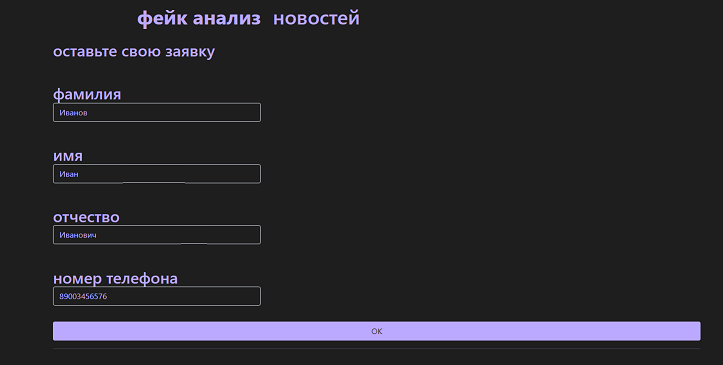


Рисунок 1 – окно формы заявки

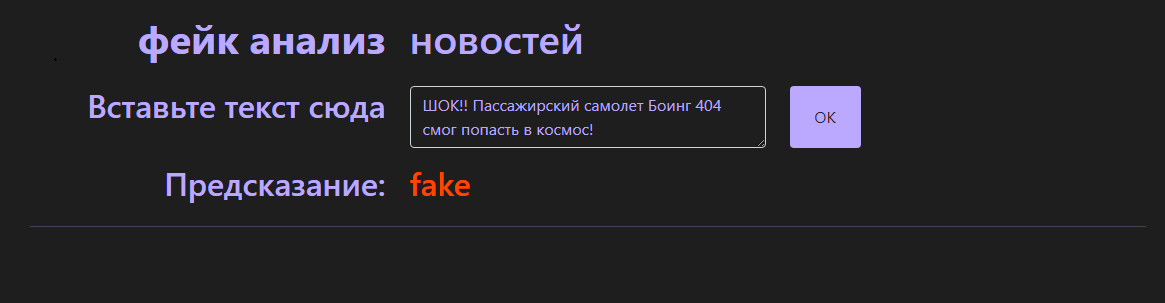


Рисунок 2 – окно при фейковой новости

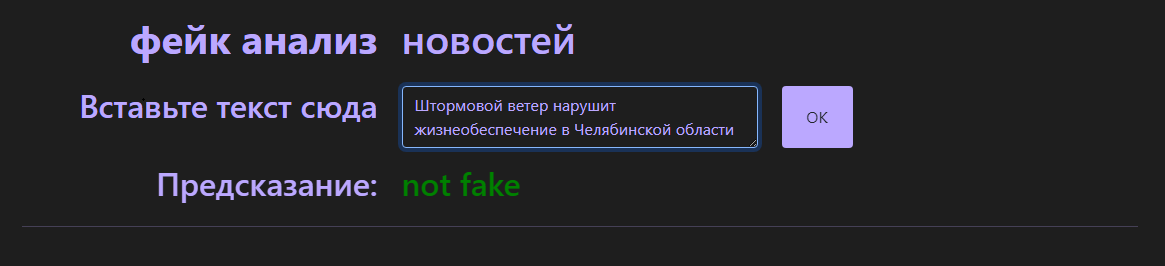


Рисунок 3 – окно при правдивой новости

**2. Нагрузочное тестирование**

Выполним нагрузочное тестирование программы и оценим эффективность автоматизированных функций ПС, запросов к БД. В качестве критерия эффективности используем время выполнения функции. Для тестирования подготовим 5 массивов исходных данных на 10, 50, 100, 500 и 1000. Тестирование проводилось на таблице обработки заявок всех клиентов. Результаты тестирования представлены в таблице 1.

Технические характеристики используемой машины:

* Оперативная память: 8 Гб
* Процессор: 4 ядра с тактовой частотой 2.333 ГГц, 8 потоков
* Разрядность Windows: 64-разрядная операционная система
* Размер жёсткого диска: 256 Гб

|  |  |  |  |
| --- | --- | --- | --- |
| № | Кол-во записей | Операция | Результат (сек.) |
| 1 | 10 | Вставка | 0,115 |
| 2 | 50 | Вставка | 0,212 |
| 3 | 100 | Вставка | 0,289 |
| 4 | 500 | Вставка | 0,422 |
| 5 | 1000 | Вставка | 0,619 |

Таблица 1 – Сравнительная таблица нагрузочного тестирования

При данном тестировании в базу данных загружается сначала 10 записей об объектах, далее 50, 100, 500 и 1000. Далее строится проект, тем самым проверяется скорость работы ПС с различными объёмами данных

После проведения теста мы получили следующие данные:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Количество исходных данных | 10 | 50 | 100 | 500 | 1000 |
| Время вычислений | 1550 мс | 2120 мс | 2890 мс | 4220 мс | 6190 мс |

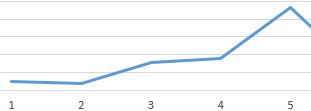


Рисунок 8 – График работы программы

**3. Стрессовое тестирование**

В этом виде тестирования необходимо проверить работу системы при чрезмерной нагрузке на неё. Стрессовое тестирование ПС будет проведено посредством отправки многократных запросов к БД PostgreSQL. Во время тестирования было обнаружено, что даже в стрессовых ситуациях приложение продолжает функционировать, БД быстро обрабатывает запросы. Могут возникать небольшие задержки в работе интерфейса, но после выполнения череды запросов задержки перестают себя проявлять.

**4. Модульное тестирование**

Для тестирования работы вставки была установлена библиотека jest и написан следующий тест:

module.exports = { presets: [['@babel/preset-env', {targets: {node: 'current'}}]], };

module.exports = { testEnvironment: 'jsdom',

moduleFileExtensions: ['js', 'json', 'vue'],

transform: { '^.+\\.js$': 'babel-jest',

'^.+\\.vue$': '@vue/vue3-jest' }

}

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9 - Результат модульного теста

**Баг-репорт 1.**

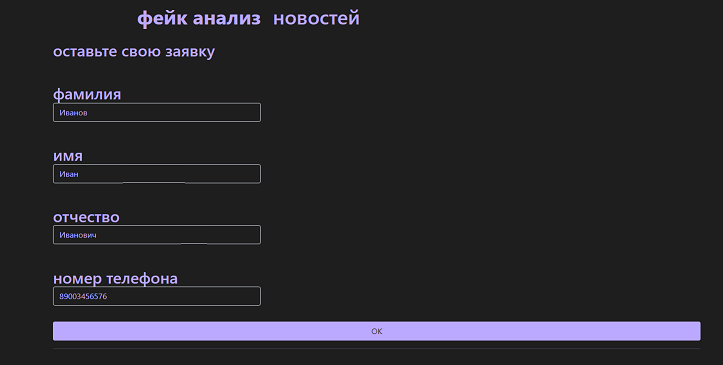
|  |  |
| --- | --- |
| Короткое описание (Summary) | Приложение позволяет вводить буквы в поле для номера телефона |
| Проект (Project) | Анализ новостей на фейк |
| Компонент приложения (Component) | form |
| Номер версии (Version) | 1.00 |
| Серьезность  (Severity) | S3 Значительный (Major) |
| Приоритет (Priority) | P2 Средний (Medium) |
| Статус (Status) | Новый |
| Автор (Author) | Молчанов Илья Александрович |
| Назначен на (Assigned To) | Молчанов Илья Александрович |
| Окружение | Windows 10 |
| Шаги воспроизведения (steps to Reproduce) | - перейти на страницу формы заявки;  - заполнить все поля;  - в поле «Номер телефона» вписать буквы;  - нажать на кнопку «ОК»; |
| Фактический результат (Result) | программа приняла номер телефона без ошибок |
| Ожидаемый результат (Expected Result) | программа должна вывести ошибку о невалидном значении |
| Прикрепленный файл (Attachment) |  |

**Баг-репорт 2.**

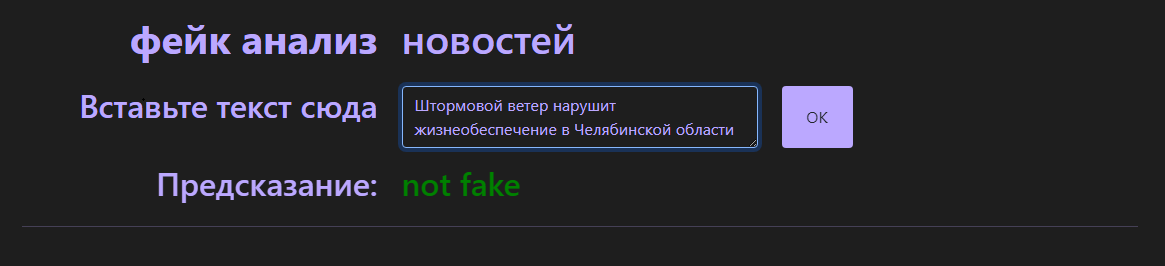
|  |  |
| --- | --- |
| Короткое описание (Summary) | Приложение позволяет вводить цифры в поле для имени |
| Проект (Project) | Анализ новостей на фейк |
| Компонент приложения (Component) | form |
| Номер версии (Version) | 1.00 |
| Серьезность  (Severity) | S3 Значительный (Major) |
| Приоритет (Priority) | P2 Средний (Medium) |
| Статус (Status) | Новый |
| Автор (Author) | Молчанов Илья Александрович |
| Назначен на (Assigned To) | Молчанов Илья Александрович |
| Окружение | Windows 10 |
| Шаги воспроизведения (steps to Reproduce) | - перейти на страницу формы заявки;  - заполнить все поля;  - в поле «Имя» вписать цифры;  - нажать на кнопку «ОК»; |
| Фактический результат (Result) | программа приняла имя без ошибок |
| Ожидаемый результат (Expected Result) | программа должна вывести ошибку о невалидном значении |
| Прикрепленный файл (Attachment) |  |

**Руководство пользователя**

1. Назначение приложения:
   * Данное программное средство разработано для анализа новостных сообщений на признаки фейка.
   * Оно предоставляет возможность проверки достоверности новостей и помогает пользователю оценить их надежность.
2. Установка приложения:
   * Скачайте все файлы из репозитория https://github.com/IlyaMolchanovv/ai-fake.
   * Для использования приложения откройте файл index.html в веб-браузере.
3. Концепция приложения:
   * Приложение состоит из двух основных окон: окна заполнения заявки и окна ввода данных для анализа.
   * Окно заполнения заявки предназначено для получения необходимых контактных данных пользователя.
   * Окно ввода данных для анализа предоставляет возможность ввести текст новостного сообщения, которое требуется проанализировать.
4. Функциональный состав приложения:
   * Окно заполнения заявки:
     + Пользователь должен заполнить все обязательные поля, такие как имя, фамилию, отчество и контактный номер телефона.



* + - Эти данные будут использоваться для связи с пользователем и предоставления результатов анализа.
  + Окно ввода данных для анализа:
    - В этом окне пользователь может ввести текст новостного сообщения, которое требуется проанализировать на признаки фейка.
    - После ввода текста пользователь может нажать кнопку "OK" или аналогичную кнопку для запуска процесса анализа.



* + Результаты анализа:
    - После завершения анализа приложение предоставит пользователю результаты, указывающие на степень достоверности новостного сообщения.
    - Результаты могут быть представлены в виде категорий, например "Достоверно", "Подозрительно" или "Недостоверно".
    - Пользователь может использовать эти результаты для принятия решения о доверии данной новости.