

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ  
Заведующий кафедрой ЭВМ  
\_\_\_\_\_ Д. В. Топольский  
« \_\_\_ » \_\_\_\_\_ 2022 г.

ПРОГРАММНЫЙ КОМПЛЕКС ПОИСКА НЕ СОДЕРЖАЩИХ  
ИНФОРМАЦИИ ОТЗЫВОВ ПОЛЬЗОВАТЕЛЕЙ ФИРМЫ  
ИНТЕРНЕТ-ПРОВАЙДЕРА

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  
ЮУРГУ-090301.2022.232 ПЗ ВКР

Руководитель работы,  
к.т.н., доцент кафедры ЭВМ  
\_\_\_\_\_ В. А. Парасич  
« \_\_\_ » \_\_\_\_\_ 2022 г.

Автор работы,  
студент группы КЭ-406  
\_\_\_\_\_ К. А. Забелина  
« \_\_\_ » \_\_\_\_\_ 2022 г.

Нормоконтролёр,  
к.п.н., доцент кафедры ЭВМ  
\_\_\_\_\_ М. А. Алтухова  
« \_\_\_ » \_\_\_\_\_ 2022 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_ Д. В. Топольский

«\_\_\_» \_\_\_\_\_ 2022 г.

### **ЗАДАНИЕ**

**на выпускную квалификационную работу бакалавра**  
студентке группы КЭ-406  
Забелиной Ксении Алексеевны,  
обучающейся по направлению  
09.03.01 «Информатика и вычислительная техника»

1. **Тема работы:** «Программный комплекс поиска не содержащих информации отзывов пользователей фирмы Интернет-провайдера» утверждена приказом по университету от 12 декабря 2021 г. №308/141

2. **Срок сдачи студентом законченной работы:** 1 июня 2022 г.

3. **Исходные данные к работе:**

- СУБД Oracle 12c;
- язык программирования баз данных PL/SQL;
- операционная система не ниже Windows 7;
- язык программирования сервера PL/SQL;
- система контроля версий SmartGit;
- среда разработки Toad for Oracle версии 12.1;
- программа поиска информации PowerGrep.

Обеспечить основной функционал программы:

– осуществить поиск неосмысленных слов и удаление сопутствующих комментариев;

– интегрирование программного комплекса в имеющуюся систему управления производством в данной фирме.

**4. Перечень подлежащих разработке вопросов:**

- импорт отзывов в данный комплекс;
- импорт базы данных существующих слов;
- анализ алгоритма поиска строк;
- разработка собственного программного продукта;
- тестирование в системе управления базами данных;
- интеграция в существующий интерфейс;
- оценка работоспособности на реальном продукте.

**5. Дата выдачи задания:** 1 декабря 2021 г.

Руководитель работы \_\_\_\_\_ /В. А. Парасич/

Студент \_\_\_\_\_ /К. А. Забелина/

## КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	10.03.2022	
Разработка модели, проектирование	21.03.2022	
Реализация системы	04.04.2022	
Тестирование, отладка, эксперименты	25.04.2022	
Компоновка текста работы и сдача на нормоконтроль	16.05.2022	
Подготовка презентации и доклада	24.05.2022	

Руководитель работы \_\_\_\_\_ /В. А. Парасич/

Студент \_\_\_\_\_ /К. А. Забелина/

## АННОТАЦИЯ

К. А. Забелина. Программный комплекс поиска не содержащих информации отзывов пользователей фирмы Интернет-провайдера. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2022, 60 с., 23 ил., библиогр. список – 19 наим.

В рамках выпускной квалификационной работы разрабатывается программный комплекс по поиску осмысленных слов в комментариях пользователей. Система учитывает процент корректных слов в одном комментарии. Впоследствии либо удаляет его, либо оставляет.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	7
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	9
1.1 Обзор аналогов .....	9
1.2 Анализ статистики корректных отзывов .....	11
1.3 Анализ основных технологических решений .....	14
1.3.1 Выбор системы управления базой данных .....	14
1.3.2 Выбор интегрированной среды разработки .....	17
2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ .....	20
2.1 Функциональные требования .....	20
2.2 Нефункциональные требования .....	21
3 ПРОЕКТИРОВАНИЕ .....	22
3.1 Архитектура предлагаемого решения .....	22
3.2 Алгоритмы поиска строк .....	27
3.3 Описание данных .....	29
4 РЕАЛИЗАЦИЯ .....	31
5 ТЕСТИРОВАНИЕ .....	39
5.1 Методология тестирования .....	39
5.2 Проведение процедуры тестирования .....	39
ЗАКЛЮЧЕНИЕ .....	46
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	47
ПРИЛОЖЕНИЕ А Исходный код анализа статистики .....	50
ПРИЛОЖЕНИЕ Б Исходный код обработки отзывов .....	54

## ВВЕДЕНИЕ

В выпускной квалификационной работе выполняется автоматизация документооборота службы Технической поддержки ООО Фирмы “Интерсвязь”, которая является объектом исследования. Предмет исследования – отзывы пользователей при обращении в техническую поддержку.

Назначение программного продукта: автоматизировать рутинную работу операторов, менеджеров службы технической поддержки, а так же сотрудников абонентского отдела по упрощению работы с абонентами.

В работе рассматривается проблема быстрого реагирования на просьбы о помощи пользователей.

Данная тема является актуальной, поскольку возросло количество звонивших пользователей (обращающихся), которые в действительности могли обойтись без общения с оператором. Живой контакт с оператором мог не понадобиться из-за того, что:

- 1) пользователь уже имел общение с консультантом и решил оставить отзыв о том, насколько эффективно было решение проблемы;
- 2) пользователь разрешил, например, ребенку поиграть на телефоне и тот в свою очередь отправил либо пустое сообщение, либо неразборчивое.

Случаев, когда пользователь писал неразборчиво, в связи с тем, что зол или мог быть в алкогольном опьянении, так же много. В пункте 1.1 будет приведена конкретная статистику.

В данном случае, действительно разбираться, из-за чего появилось то или иное сообщение пользователя, не является эффективным. Главная цель ВКР – это решить проблему таких неинформативных сообщений в работе службы Техподдержки. Ниже представлены задачи данной работы:

- 1) создание статистики корректных отзывов;

2) разработка процедуры по поиску осмысленных слов на основе имеющихся данных;

3) разработка документации по эксплуатации;

4) внедрение программного комплекса в интерфейс фирмы “Интерсвязь”.

В данном программном продукте будет производиться сортировка комментариев, а также последующее удаление лишних сообщений.



# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Обзор аналогов

В рамках исследования предметной области мы рассмотрели следующие аналоги разрабатываемого программного комплекса:

- 1) Total Commander;
- 2) Google – поиск;
- 3) Multi Text Funder.

Далее рассмотрим особенности каждого из них.

Total Commander [1] – файловый менеджер, который поможет найти информацию по ключевым словам в файлах или папках или найти, например, в каком файле находится обработчик кода.

Про Total Commander сказано очень многое. Те, кому приходится ежедневно работать с этим файловым менеджером, умеют настраивать его под свои задачи, знают комбинации клавиш быстрого доступа к различным функциям программы, а кое-кто, быть может, даже знаком и со скрытыми возможностями этой утилиты. Для таких пользователей преимущества Total Commander перед стандартным файловым менеджером очевидны.

Преимуществами Total Commander являются:

- 1) поиск текста в файлах;
- 2) поиск по количеству вложенных папок;
- 3) поиск по примерному времени создания файла;
- 4) поиск по размеру файла;
- 5) поиск в архиве.

Поисковая система Google [2] – всемирно-известный интернет ресурс, по поиску необходимых пользователю данных.

Преимуществами Google поиска являются:

- 1) простота в использовании;
- 2) отсутствие необходимости в особых знаках;
- 3) огромная база данных;
- 4) высокоскоростной поиск данных;
- 5) максимальная точность информации;
- 6) учет качества каждой страницы.

Multi Text Funder [3] – это программа, осуществляющая множественный поиск текста в документах.

Преимуществами Multi Text Funder являются:

- 1) простота и удобство;
- 2) предварительный просмотр найденной информации;
- 3) одновременный поиск нескольких слов;
- 4) высокая скорость поиска.

**Вывод.** Из всего вышесказанного следует, что имеются несколько программ осуществляющих поиск информации. Есть также программные продукты, которые не указаны, они отслеживают нецензурные выражения в социальных сетях, которые сложно найти в публичном доступе. В каждом из этих продуктов, есть свои недостатки. А также нельзя применить код ни одной из выше перечисленных программ под уже существующую логику в данной фирме, в связи с тем, что данная информация так же недоступна и не полноценна. Поэтому разработка программного продукта по удалению некорректных комментариев, является актуальной задачей.

## 1.2 Анализ статистики корректных отзывов

Прежде чем начинать программировать процедуру по удалению отзывов, необходимо понять, что именно стоит удалять и стоит ли вообще их удалять.

Несколько месяцев проводился анализ отзывов. Приведем несколько примеров:

- 1) “”;
- 2) “Плохо”;
- 3) “Спасибо за помощь”;
- 4) “10”;
- 5) “Проблему не решили”;
- 6) “Нормально”;
- 7) “Не чего не ипредпринимаюь”;
- 8) “Бещ комментариев, нт слов”;
- 9) “Вабще не решают проблемы”.

**Выводы.** Во-первых, если в комментарии не более двух слов, то такие комментарии стоит удалить.

Во-вторых, встречаются пустые комментарии, либо содержащие только цифры и знаки препинания. Тоже будут удаляться.

В-третьих, теперь рассмотрим комментарии семь-девять.

В комментарии восемь, когда два слова корректны, а два нет, еще возможно понять смысл. Но представим, что слов будет больше и количество корректных слов будет равно количеству некорректных. Смысл будет понятен лишь на половину или же совсем не понятен. Поэтому было принято решение, что количество осмысленных слов должно превышать 75%, а количество неосмысленных – должно быть менее 25%. Тогда такие комментарии как семь и девять будут корректны. И это будет правильно – ведь их смысл понятен.

Чтобы удостовериться, какие комментарии нужны предприятию, рассмотрим статистику ниже:

1) количество комментариев, в которых меньше двух слов – равно 813 из 5421 сообщения, что является 15% от всех отзывов (рисунок 1);

2) количество пустых комментариев – равно 24 из 5421 сообщения, что является 0,44% от всех отзывов (рисунок 2);

3) количество комментариев, в которых три четверти слов корректны, а остальные нет – равно 272 из 5421 сообщения, что является 5% от всех отзывов (рисунок 3);

4) количество всех некорректных комментариев из пунктов 1, 2 и 3 равно 1109 из 5421 сообщения, что является 20,5% от всех отзывов (рисунок 4).

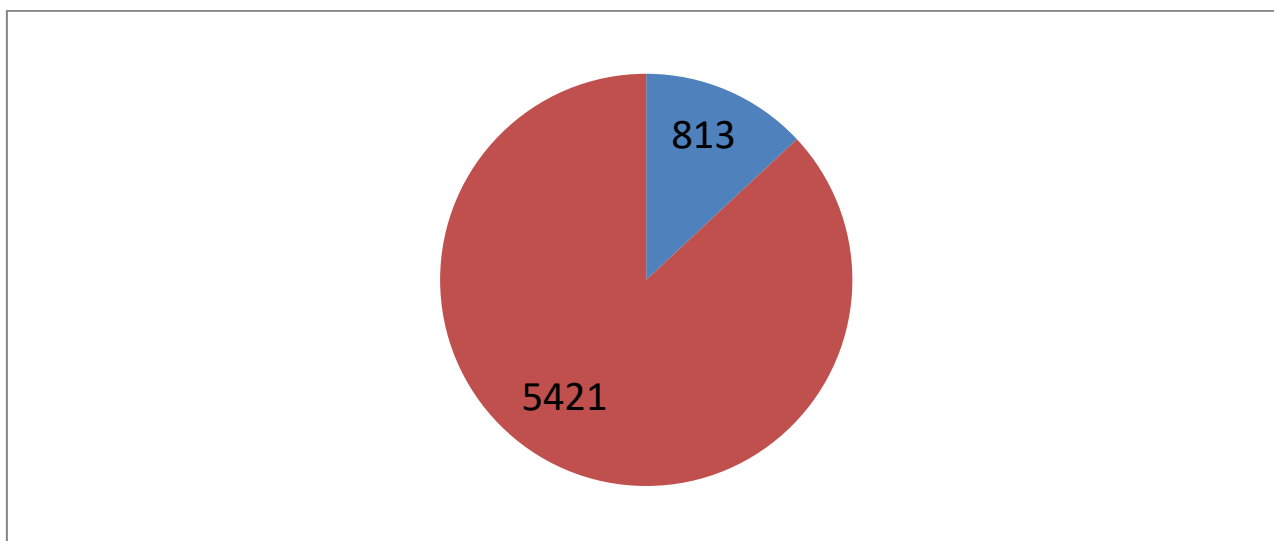


Рисунок 1 – Количество комментариев, в которых не более двух слов на все комментарии

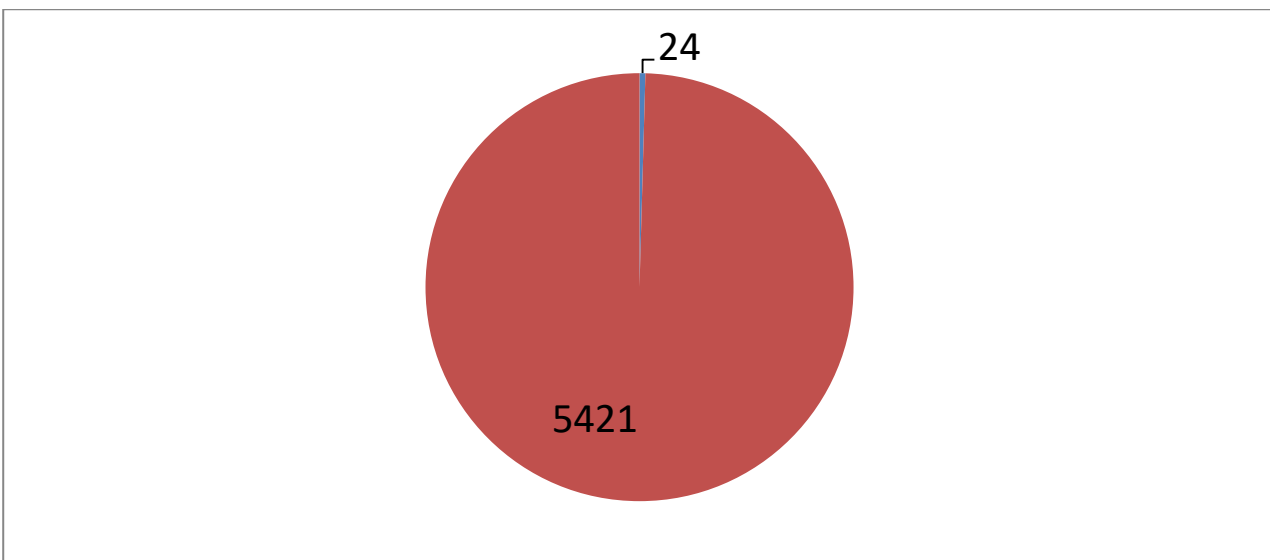


Рисунок 2 – Количество пустых комментариев на все комментарии

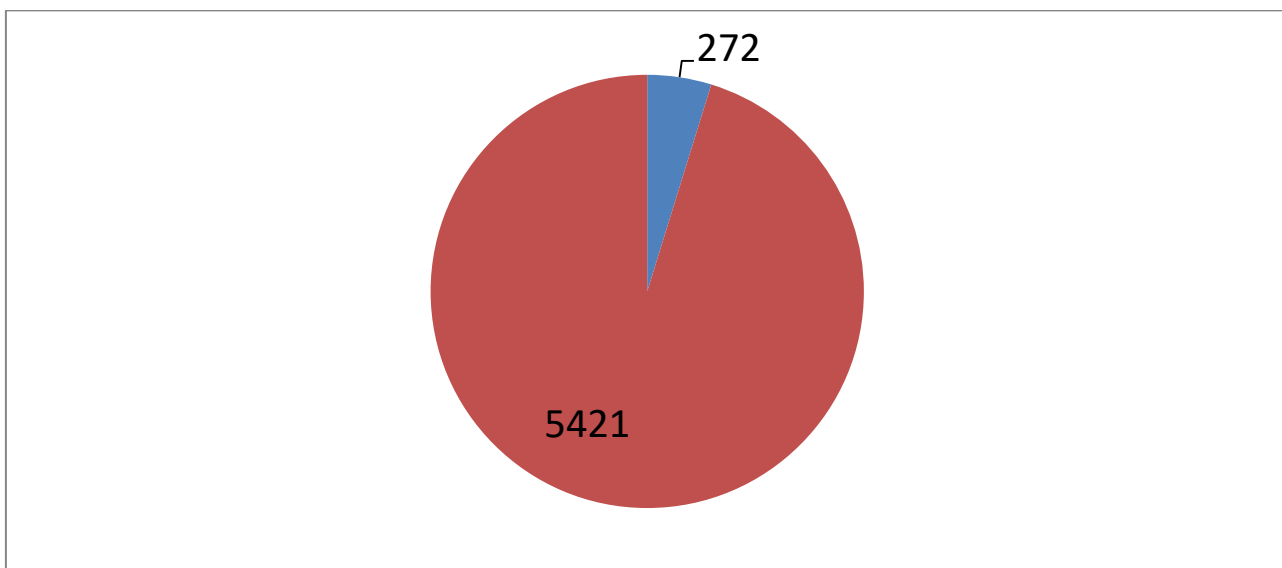


Рисунок 3 – Количество комментариев, в которых три четверти слов корректны, а остальные нет на все комментарии

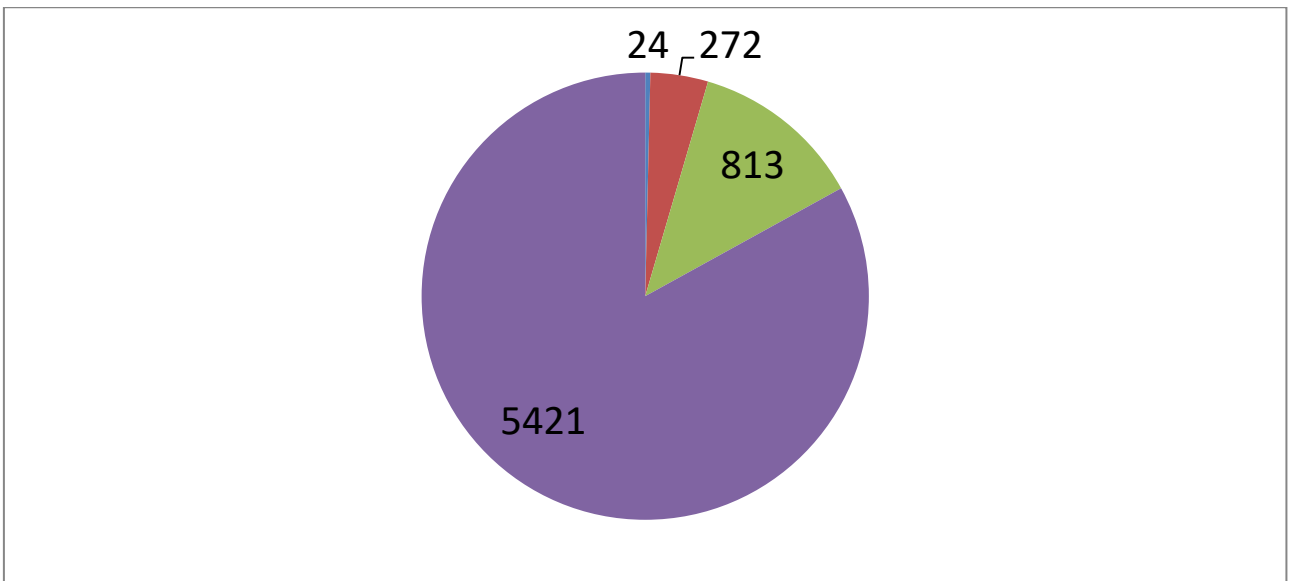


Рисунок 4 – Количество всех некорректных комментариев из пунктов один – три на количество всех комментариев

Код для анализа статистики можно увидеть в приложении А.

### 1.3 Анализ основных технологических решений

#### 1.3.1 Выбор системы управления базой данных

Определим, какая СУБД больше подойдет для разработки программы:

- 1) колоночная;
- 2) графовая;
- 3) документная;
- 4) ключ-значение;
- 5) реляционная.

Далее рассмотрим особенности каждой из них:

Колоночные СУБД – это NoSQL СУБД, где данные хранятся "поколоночно", т.е. колонка – это как отдельная таблица. Соответственно чтение будет происходить из конкретного столбца сразу.

Колоночные СУБД неэффективны, если:

- 1) выборка простая;
- 2) выборка осуществляется по ключевым словам;
- 3) параметры статичны;
- 4) количество данных меньше нескольких сотен миллионов [4];
- 5) отсутствие транзакций;
- 6) поддержка не всех операций DML;
- 7) ограничение в операциях JOIN;
- 8) строгая типизация;
- 9) зависимость от оперативной памяти;
- 10) отсутствие оконных функций и зависимых подзапросов [5].

Графовые СУБД [4] – это NoSQL СУБД, предназначенные для работы с графами, с их узлами, свойствами, и произвольными отношениями между узлами.

Графовые СУБД неэффективны, если строить какое-то подобие социальной сети, или реализовывать систему оценок и рекомендаций.

Документные СУБД [4] – это NoSQL СУБД, где основной единицей логической модели данных является документ – структурированный текст, с определенным синтаксисом.

Документные СУБД не эффективны, если:

- 1) используется транзакционная модель;
- 2) необходимо формирование отчетности;
- 3) структура данных статична;
- 4) связность практически отсутствует [6].

Ключ-значение [6] – это NoSQL СУБД, где есть уникальный ключ и его значение в упрощенном виде. Чаще всего такие СУБД используют для кэширования.

Ключ-значение не эффективна, если:

1) в базе данных много сущностей (таблиц), а у сущностей будут сложные структуры с разными типами данных;

2) делать из этой таблицы сложные запросы, которые возвращают множества строк [4];

3) объем данных не должен превышать объем ОЗУ;

4) слабая сохранность данных;

5) нет транзакций и связанных данных.

Реляционные СУБД работает с небольшими по размерам транзакциями, но идущими большим потоком, и при этом от системы требуется минимальное время отклика, а так же возможность, при определенных условиях, отменить любые изменения выполняемых в рамках транзакции [4].

Преимущества реляционных СУБД:

1) целостность данных;

2) фиксация изменений;

3) хранимые процедуры;

4) блокировки базы данных и параллельный доступ.

**Вывод.** В данном случае стоит выбрать реляционную СУБД, в связи с тем, что задачей является создание хранимой процедуры и использование в ней транзакций, операций DML и DDL и сохранение данных.

Определим, какие есть реляционные СУБД:

1) MySQL;

2) PostgreSQL;

3) Oracle;

4) Microsoft SQL Server.

Рассмотрим каждую систему из них более подробно:

MySQL [6] – это система управления реляционными базами данных.

Минусами MySQL являются:



- 1) низкая производительность;
- 2) возможная потеря данных из-за нестабильности сервера.

PostgreSQL – это объектно-реляционная система управления базами данных.

Минусами PostgreSQL являются:

- 1) строгая типизация;
- 2) сложность написания;
- 3) система автоматизации может вызвать затруднения.

Oracle 12c [7] – это объектно-реляционная система управления базами данных, предназначена для облачных сред и может быть размещена на одном или нескольких серверах, что позволяет управлять базами данных, которые содержат миллиарды записей.

Достоинствами Oracle 12c являются:

- 1) инновационный функционал;
- 2) надежность хранения данных [8].

Microsoft SQL Server [9] – это реляционная система управления базами данных.

Минусами Microsoft SQL Server являются:

- 1) взаимодействие только с Windows;
- 2) высокая стоимость.

**Вывод.** Все варианты могут подойти для разработки приложения.

Руководством было принято решение использовать СУБД Oracle 12c, уже используемую в компании.

### **1.3.2 Выбор интегрированной среды разработки**

Рассмотрим основные популярные среды разработки:

- 1) Oracle SQL Developer;
- 2) SQL Navigator;

- 3) PL/SQL Developer;
- 4) Toad for Oracle;
- 5) DataGrip.

Далее рассмотрим особенности каждого из этих инструментов:

Oracle SQL Developer [10] – это программное средство от корпорации Oracle, для работы с запросами SQL и программными единицами PL/SQL.

Минусами Oracle SQL Developer являются:

- 1) не сохраненные настройки;
- 2) отсутствие контекстно-зависимых подсказок встроенных в интерфейс;
- 3) большая ресурсоемкость.

SQL Navigator – программное обеспечение, представляющее собой платформу для визуального редактирования данных и метаданных и генерации скриптов.

Минусами SQL Navigator являются:

- 1) отладчик, не являющийся интуитивно понятным;
- 2) ошибки в интерфейсе.

PL/SQL Developer – интегрированная среда разработки на языках SQL и PL/SQL, с возможностью администрирования баз данных, ориентирована на применение в среде Oracle database [11].

Минусами PL/SQL Developer являются:

- 1) отсутствие экспорта/импорта данных;
- 2) не слишком удобен и разнообразен интерфейс;
- 3) мало функций для анализа данных.

Toad for Oracle – программное обеспечение, используемое для работы с реляционными системами управления базами данных.

Плюсами Toad for Oracle являются:

- 1) форматирование кода;

- 2) экспорт/импорт данных;
- 3) удобство для анализа данных;
- 4) удобный и понятный интерфейс.

DataGrip [12] – это IDE, для написания SQL-запросов и работы с базами данных на платформах Linux, MacOS и Windows.

Минусами DataGrip являются:

- 1) отсутствие читабельного интерфейса;
- 2) плохо развитый браузер объектов;
- 3) не корректное форматирование текста.

## 2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

### 2.1 Функциональные требования

Система должна обеспечить возможность выполнения следующих функций:

- 1) обеспечить поиск некорректных слов;
- 2) обеспечить удаление некорректных слов;
- 3) обеспечить закрытие тикета (задачи) с некорректным комментарием;
- 4) обеспечить закрытый тикет сообщением о том, что комментарий удален;
- 5) обеспечить совместное использование несколькими разработчиками пакетов базы данных;
- 6) обеспечить просмотр закрытого тикета;
- 7) обеспечить возможность сохранения данных в таблице, в течение времени жизни самих тикетов с такими данными;
- 8) обеспечить возможность сохранения содержания удаленного комментария в закрытом тикете;
- 9) обеспечить возможность не корректировать осмысленный комментарий;
- 10) обеспечить возможность изменение процедуры программистом базы данных данной фирмы по требованию заказчика;
- 11) обеспечить возможность удаление процедуры программистом базы данных данной фирмы по требованию заказчика.

## 2.2 Нефункциональные требования

Из нефункциональных требований следует выделить следующие:

- 1) обеспечить интегрирование программного комплекса в систему управления производством в данной фирме;
- 2) обеспечить сохранение содержания отзыва, когда число осмысленных слов больше двух;
- 3) будет считаться за осмысленное слово – название в кавычках;
- 4) обеспечить сохранение содержания отзыва, когда в кавычках не менее двух осмысленных слов;
- 5) если в кавычках не менее двух осмысленных слов, то обеспечить удаление кавычек;
- 6) обеспечить сохранение содержания отзыва, когда количество осмысленных слов не менее 75% включительно, соответственно, удаление при количестве неосмысленных слов менее 25%;
- 7) обеспечить методические указания разработчику;
- 8) обеспечить методические указания менеджеру.

## 3 ПРОЕКТИРОВАНИЕ

### 3.1 Архитектура предлагаемого решения

Прежде чем приступать к рассмотрению архитектуры предлагаемого решения по корректированию отзывов пользователей, первым делом необходимо изучить существующие на фирме этапы работы с отзывами.

Первый этап работы операторов отдела службы технической поддержки: пользователь оставляет какой-то отзыв в мобильном приложении. Отзыв загружается во внутреннюю систему (service-desc) в виде задания (тикета). Оператор открывает в service-desc свои задания и просматривает каждое. Если в отзыве действительно есть какая-то проблема, он звонит пользователю, в противном случае закрывает тикет. Ознакомиться с процессом можно на рисунке 5.

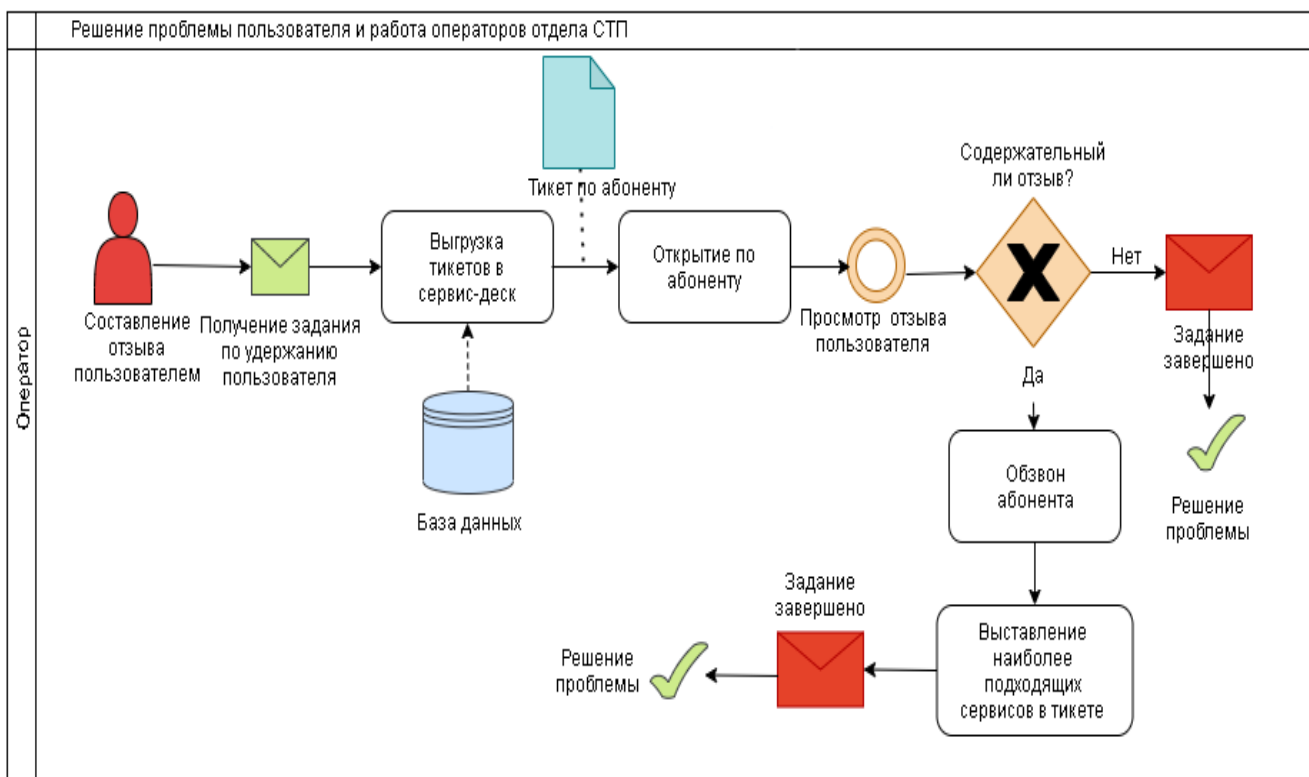


Рисунок 5 – Как на данный момент работает система в интернет-провайдере.  
Модель AS-IS

Как можно заметить, из-за того, что комментарии на начальном этапе не обрабатываются, оператору приходится делать намного больше действий. Из-за этого в период большого наплыва отзывов возникает нехватка времени. Не говоря уже о том, что вся рутинная работа лежит на человеке, что влечет за собой много недочетов или грубых ошибок. Например, оператор может не загрузить все свои задания во внутреннюю систему, таким образом, некоторые проблемы абонентов останутся нерешенными, что приведет к оттоку пользователей. Более того, каждый оператор определяет проблему абонента по своему и выставляет в карточки заданий наиболее подходящие с его точки зрения сервисы.

После рассмотрения архитектуры существующего решения, вернемся к предлагаемому решению. В нашем случае применяется паттерн проектирования – MVC (model-view-controller), как показано на рисунке 6.

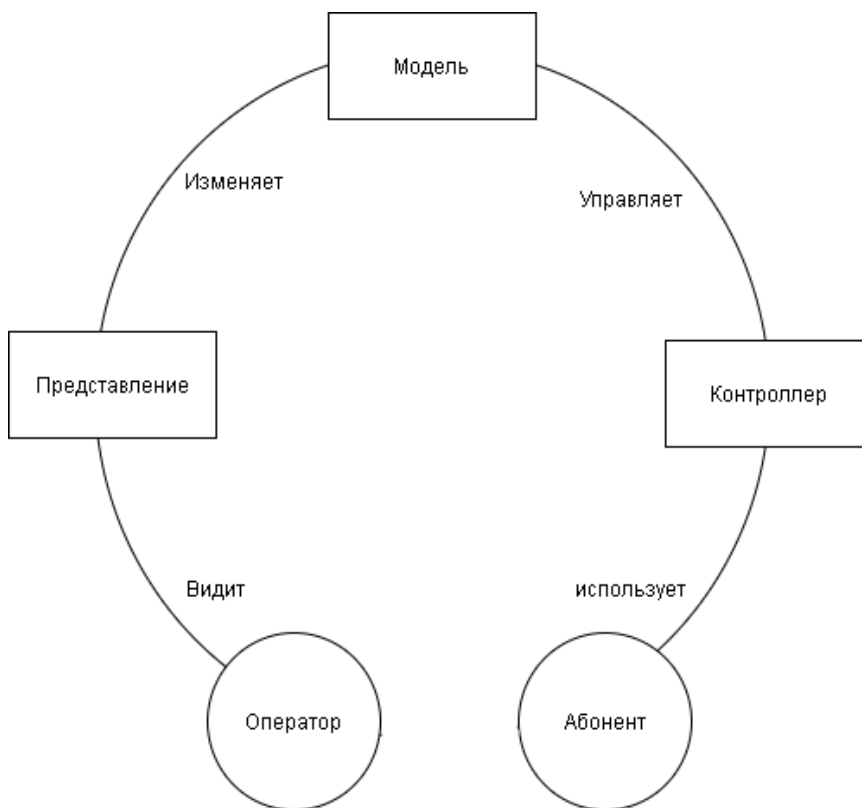


Рисунок 6 – Модель MVC

Абонент отправляет комментарий – назовем это событием. Когда произошло событие, контроллер запускает выполнение действий модели и одновременно создается задание.

Рассмотрим модель ТО-ВЕ как более наглядное представление данной архитектуры на рисунке 7.

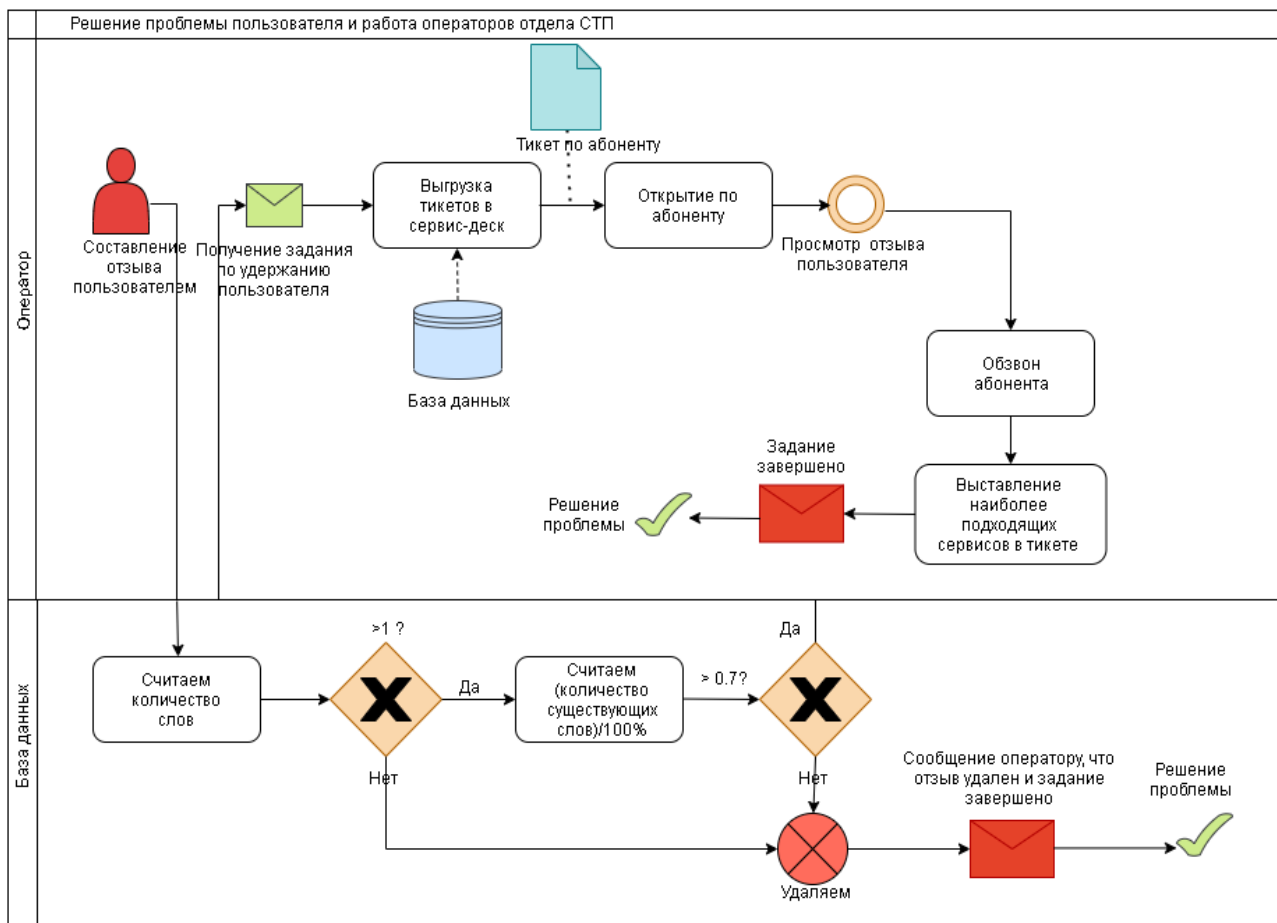


Рисунок 7 – Модель ТО-ВЕ

Как можно видеть, теперь оператору не надо будет тратить время на просмотр всех комментариев, а только тех, в которых действительно содержится проблема.

Рассмотрим более детальное выполнение процедуры.

Абонент отправляет отзыв. В этот момент отзыв попадает в нашу базу данных в таблицу `sd_ticket_comment` (таблица будет описана дальше), а так же создается тикет на обработку комментария. После попадания комментария в



таблицу, процедура начинает считывать количество букв [а..я] в последовательности между пробелами – назовем их словами. Если их количество не больше двух или комментарий содержит знаки препинания, цифры или является пустым, то модель будет автоматически закрывать задание (это все будет происходить настолько быстро, чтобы оно не распределилось на диспетчеров и поэтому очень важно, чтобы процедура была максимально оптимизирована). Если задание было закрыто, то действие процедуры переходит к следующему комментарию или прекращает обработку отзывов до следующей отправки комментария пользователем. Если же задание не было закрыто, то происходит выполнение запроса и с помощью регулярных выражений необходимо извлекать каждое слово по отдельности и заносить его в массив `word_all_arr`. Далее через цикл модель обращается к внешней библиотеке и сравнивает каждый элемент массива со словами, которые нашли в библиотеке. Словарь будет храниться в виде таблицы `dictionary_correct_word` (описание можно найти в п. 3.3). Слова в ней находятся склоняемые во всех падежах, а так же во всех существующих в русском языке частях речи. Так данная таблица по необходимости заказчика может пополняться или некоторые слова могут удаляться. После сравнения слов, считаем количество корректных.

Определим критерий корректных слов в отзыве по формуле (1).

$$\frac{\text{количество совпадающих слов}}{\text{количество всех слов}} \cdot 100\% \geq 75\% \quad (1)$$

Если критерий выполняется, то задание попадет диспетчеру в очередь на обработку, в ином случае тикет закроется.

Впоследствии оператор может посмотреть результат события.

Для более наглядного отображения вышесказанного рассмотрим алгоритм на рисунке 8 ниже.

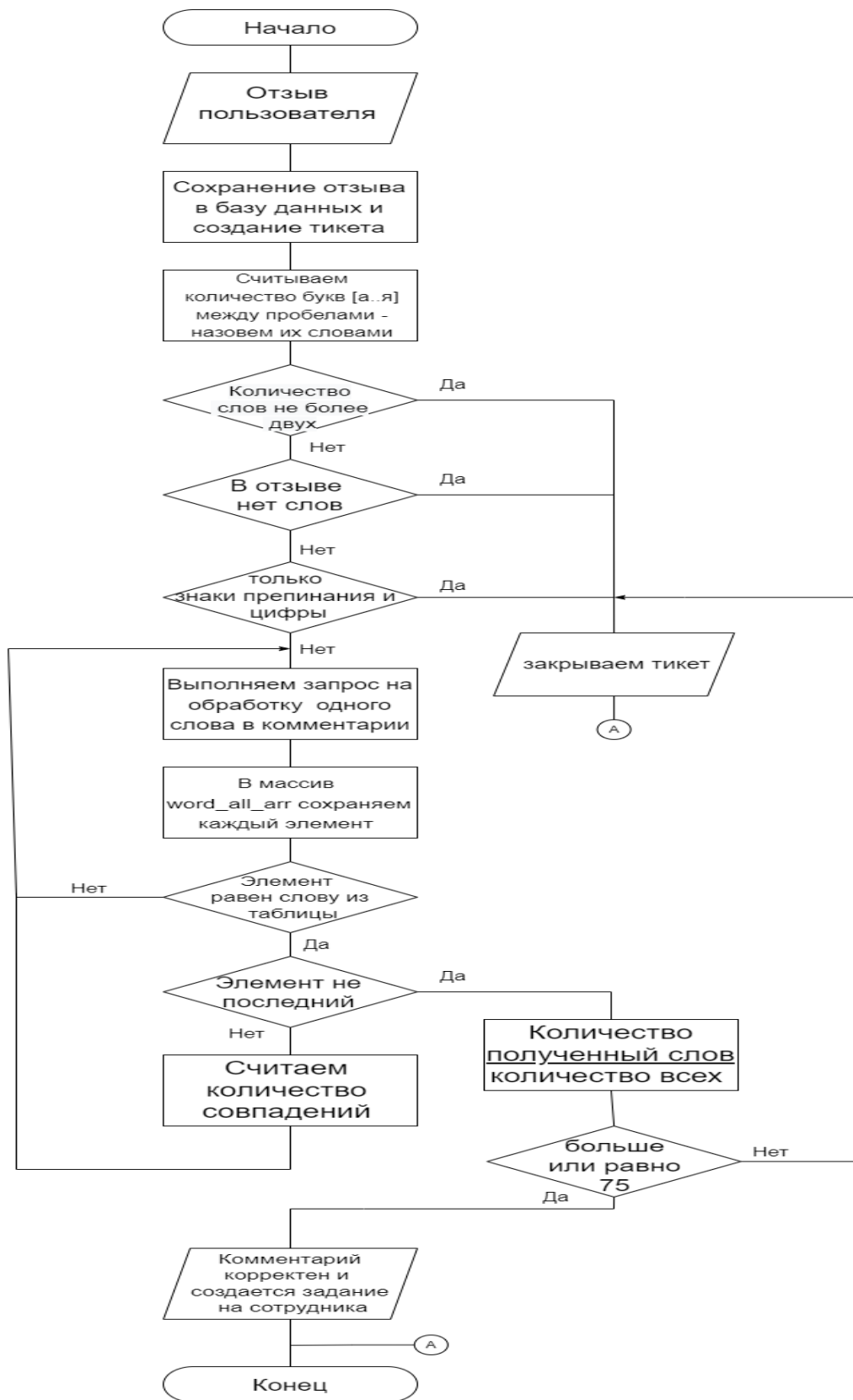


Рисунок 8 – Алгоритм выполнения процедуры

## 3.2 Алгоритмы поиска строк

Компьютерные алгоритмы поиска строк очень важный класс строковых алгоритмов, в которых одно или несколько предложений находятся в более крупном абзаце [13]. Строка, по которой осуществляется поиск, называется шаблоном.

В качестве примера можно привести азбуку Морзе, когда целые слова составляются из точек и длинных и коротких тире просто сопоставлением.

“••••• —••• —•• — — —” – слово на азбуке Морзе сопоставляется со словом “HELLO”.

Есть два вида поиска строки:

- 1) поиск, которому задают параметр и определяют, как именно данный параметр должен обрабатываться в тексте;
- 2) поиск по регулярному выражению или по шаблону.

Для примера первого вида поиска, возьмем предложение “Права одного человека заканчиваются, когда начинаются права другого человека”, в качестве параметра зададим слово “права”, его задачей является количество вхождений в предложение, то есть два.

К минусу такого поиска можно отнести то, что изменение по падежам слова, приведет к тому, что поиск не сможет его найти.

Для примера второго вида поиска, в качестве шаблона, можно привести “прав?”. В качестве знака вопроса можно подставить любую букву, и тогда данное слово можно склонять по падежам.

К минусу такого поиска можно отнести его сложность относительно первого.

Рассмотрим некоторые алгоритмы поиска строки.

«Наивный» строковый поиск [14] – это алгоритм, который в цикле сравнивает шаблон и каждый элемент строки и в конце цикла выводит результат.

Основной минус данного алгоритма – скорость. Из-за прохождения циклом каждого элемента, скорость выполнения сильно замедляется.

Алгоритм Кнута-Морриса-Пратта [15] – это алгоритм, который вычисляет совпадение по шаблону до тех пор, пока два символа не будут различаться с шаблоном. В таком случае, цикл полностью завершается.

Основной минус данного алгоритма – это то, что если текст будет множество раз совпадать с шаблоном, то скорость выполнения цикла будет такая же медленная, как и в предыдущем.

Алгоритм Bitap [16] – это алгоритм, который приблизительно сопоставляет строки шаблона со строками текста. Алгоритм разбивает шаблон на единицу и ноль и дальше работает булева алгебра.

Основным минусом данного алгоритма является то, что он работает лучше с небольшим набором символов.

Алгоритм двустороннего сопоставления строк [17] – это алгоритм, который говорит сам за себя. Для начала он сравнивает начало строки, затем ее конец и если они совпадают, то цикл заканчивается.

Основным минусом данного алгоритма является ситуация, когда совпадает начало и конец, но не совпадает середина и тогда данный алгоритм выведет неверный результат.

Алгоритм обратного недетерминированного сопоставления строк [18] – это алгоритм, который также как и Bitap используется битовую последовательность для сравнения. Сопоставления идут с конца и если они доходят до конца и битовая последовательность равна единице, то цикл прошел успешно, и он завершается.

Основной минус такой же, как у Bitap – он работает лучше с маленьким набором символов.

FM-индекс [19] – это алгоритм, который работает на сжатие текста и индексирование его. Его главное преимущество – это уменьшение размера текста.

Основным минусом данного алгоритма является то, что он очень хорошо работает только для вычисления длины.

Алгоритм Р. Боуера и Д. Мура (БМ-поиск) [15] – это алгоритм, который работает справа налево. Он сравнивает не по одному символу, а до тех пор, пока не закончится совпадение символов текста и шаблона. Данный алгоритм за счет сравнения нескольких символов работает значительно быстрее, относительно предыдущих алгоритмов.

### 3.3 Описание данных

Опишем две основные таблицы данных, которые будут применяться в программном продукте:

1) `dictionary_correct_word` – таблица, которая хранит словарь корректных слов;

2) `sd_ticket_comment` – таблица, отвечающая за комментарии в тикетах.

Опишем столбцы `dictionary_correct_word` в таблице 1.

Таблица 1 – Описание столбцов таблицы `dictionary_correct_word`

Наименование	Описание
<code>word_id</code>	Идентификатор корректных слов
<code>word</code>	Корректное слово

Опишем столбцы `sd_ticket_comment` в таблице 2.

Таблица 2 – Описание столбцов таблицы sd\_ticket\_comment

<b>Наименование</b>	<b>Описание</b>
ID	Идентификатор комментария
TICKET_ID	Идентификатор тикета
INS_USER_ID	Идентификатор пользователя, создавшего комментарий
INS_DATE DATE	Дата создания комментария
COMMENTS	Текст комментария

На вход поступает комментарий, он добавляется в таблицу sd\_ticket\_comment в качестве новой записи. Программный продукт разбивает комментарий на отдельные слова и сравнивает их с таблицей dictionary\_correct\_word. Затем по условиям выдвинутым в статистике в п 1.1, на выходе либо ничего не изменяется в тикете, либо он закрывается, так как комментарий некорректен.

## 4 РЕАЛИЗАЦИЯ

Для этапа тестирования, необходимо проанализировать, что будет отображаться в интерфейсе, чтобы было понятно, что было, а что поменялось. Так же запустим программный продукт и выведем результаты.

Для начала рассмотрим интерфейс, в котором располагается список тикетов на рисунках 9 и 10.

На рисунке 9 можно видеть:

- 1) «незакрытые», «на исполнение», «дан срок» и «сохранить новый» – созданы для тех, для кого данные задания распределяются;
- 2) «Номер» – это ID тикета;
- 3) «Состояние»: новый (только созданный), открытый (если сотрудник его взял в работу), закрытый (задание выполнено);
- 4) дата регистрации или же дата создания тикета;
- 5) во вкладке сервисы отображаются критерии, по которым применялся фильтр данных, в данном случае, поиск происходит по отзывам в мобильном приложении;
- 6) в поле «Ответственность» указывается, ответственный отдел и фамилия ответственного сотрудника;
- 7) в поле «Исполнение» указывается, исполнительный отдел и фамилия сотрудника взявшего тикет;
- 8) за любыми действиями тикета можно наблюдать, для чего в тикете в поле наблюдатель можно указать свою фамилию;
- 9) так же можно осуществить фильтр по адресу пользователя, его фамилии или комментария;
- 10) в поле «источник» вводится то, откуда пришел данный тикет.

«Не закрытые» – это количество тикетов, которые еще не успел закрыть сотрудник. «На исполнении» – это количество тикетов, которые нужно взять в работу. «Дан срок» – это тикеты, у которых выставлен срок закрытия. «Сохранить новый» – это создание тикета, например, нужно создать задание сотруднику на то, чтобы позвонить абоненту в процессе обработки тикета.

В поле «Наблюдатель» происходит фильтр по фамилии наблюдающего.

**Фильтры**

[Незакрытые \(0\)](#)  
[На исполнение \(0\)](#)  
[Дан срок \(0\)](#)  
[Сохранить новый](#)  
Номер

Номер

Состояние

Новый  
 Открыт  
 Закрыт

Когда зарегистрирован

Дата регистрации

01.03.2022 00:00

31.03.2022 00:00

▼ Сервисы

Физ  Юр

Сервис  НЕ

Отзыв  НЕ

Админ.вопрос / Признак  НЕ

Причина / Возражение  НЕ

Результат  НЕ

Пусто

▶ Ответственность

▶ Исполнение

▶ Наблюдатель

▶ Адрес

▶ Абонент

▶ Комментарий

▼ Источник

Мобильное приложение

Рисунок 9 – Список тикетов (фильтр)



Новый 30.03.2022 20:55 30.03.2022 20:55	Оценка: Плохо. Комментарий: не продуктивно Тикет №253246184
Новый 30.03.2022 20:46 30.03.2022 20:46	Оценка: Плохо. Комментарий: часто пропадает трансляция Тикет №252359601
Новый 30.03.2022 20:15 30.03.2022 20:15	Оценка: Плохо. Комментарий: удовлетворена ответом Тикет №252612230

Рисунок 10 – Список тикетов

На рисунке 10 можно видеть:

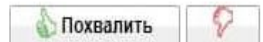
- 1) в первом столбце состояние тикета и дату регистрации;
- 2) во втором столбце есть оценка, которая делится на хорошо и плохо (можно видеть на рисунках 11 и 12) и указывается пользователем при отправке отзыва;
- 3) дальше непосредственно сам комментарий (интерфейс для отправки пользователем комментарий представлен на рисунке 13), который мы вытаскиваем в коде и ID тикета.

- Баланс**  
0,00 руб.
- Оплата**  
пополните счёт
- Бонусы**  
экономия 0 руб./мес.
- Услуги**  
...
- История**  
оплата и детализация
- Личные данные**  
[Redacted]
- Обратная связь**



Оцените качество ответов техподдержки "Интерсвязь"

14.10.2021 15:22:21 Консультация в call центре



14.10.2021 15:16:25 Консультация Голосового Ассистента

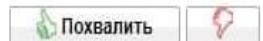


Рисунок 11 – Оценка пользователя (часть 1)

## ← Качество обслуживания

Консультация чат-бота 03.11.2021 17:16 Вам удалось оперативно решить свой вопрос?	 
Консультация чат-бота 03.11.2021 11:29 Вам удалось оперативно решить свой вопрос?	 
Консультация чат-бота 03.11.2021 11:08 Вам удалось оперативно решить свой вопрос?	 
Консультация чат-бота 02.11.2021 15:08 Вам удалось оперативно решить свой вопрос?	 
Консультация чат-бота 02.11.2021 13:06 Вам удалось оперативно решить свой вопрос?	 
Консультация чат-бота 02.11.2021 12:37 Вам удалось оперативно решить свой вопрос?	 

Рисунок 12 – Оценка пользователя (часть 1)

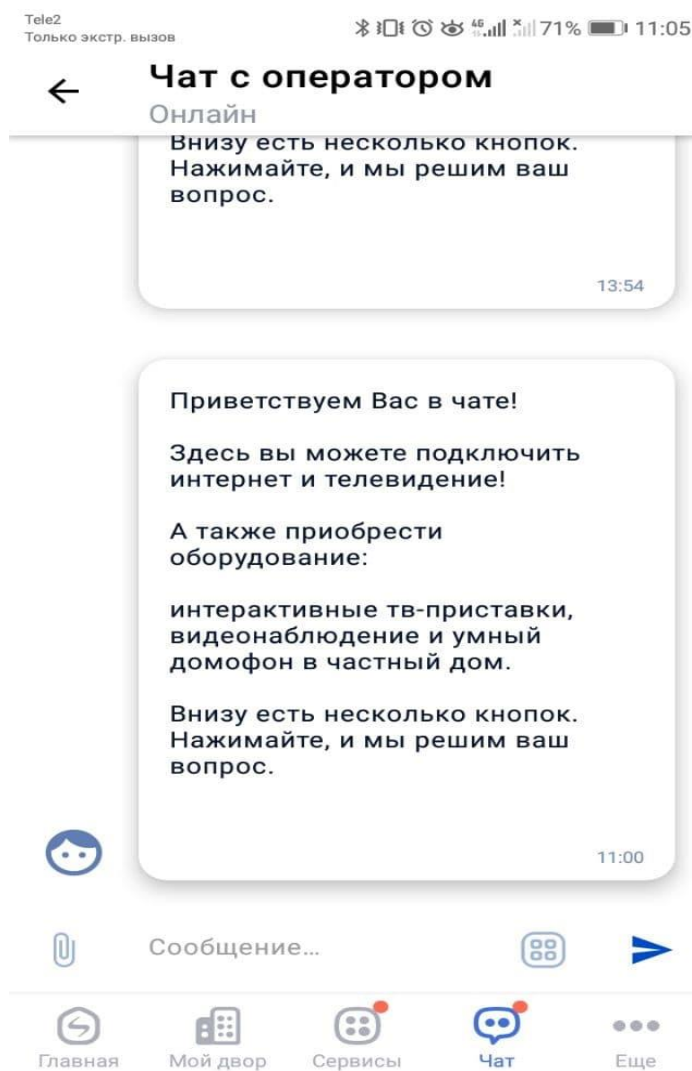


Рисунок 13 – Интерфейс отправки комментария

На рисунке 14 можно видеть:

- 1) в «зарегистрировано» указывается, дата создание и кем создается. В данном случае тикет был создан через job (job – это процедура, выполняющаяся либо по графику, либо по определенным условиям, таким как, например, создание комментария);
- 2) состояние, которое указывается разной цветовой гаммой;
- 3) дальше критерии, по которым был найден комментарий и приоритет;
- 4) в комментариях к тикету, можно видеть все комментарии, которые были созданы пользователем за время существования тикета;

5) самый первый комментарий: «Работа с анлайком из мобильного приложения» создается из базы, если первый отзыв от пользователя пришел с оценкой: «плохо».

Если состояние серого цвета, значит тикет – новый. Если состояние оранжевое, значит тикет – открытый. Если состояние красного цвета, значит тикет – закрытый.

Проблема	
Зарегистрировано	30.03.2022 20:15:09 <a href="#">JOB</a>
Состояние	<span>Новый</span>
Источник	Мобильное приложение
Сервис	Отзыв
Админ.вопрос / Признак	-
Причина / Возражение	-
Результат	-
Приоритет	0
Закрето	

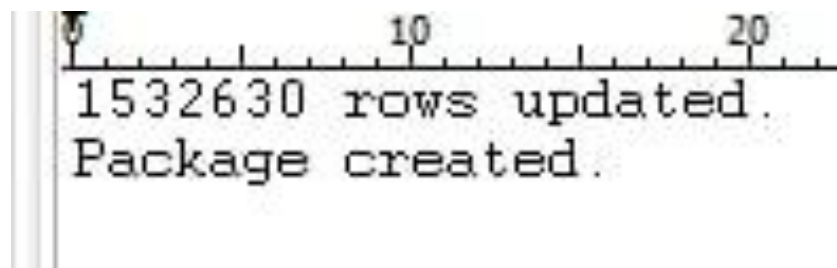
Статистика по тикетам	
-----------------------	--

Комментарии к тикету (2)	
Добавлено	Комментарий
30.03.2022 20:15:09 <a href="#">JOB</a>	Работа с анлайком из мобильного приложения.
30.03.2022 20:15:09 <a href="#">JOB</a>	Оценка: Плохо. Комментарий: удовлетворена ответом Тикет №252612230

Рисунок 14 – Тикет

Результаты выполнения программного продукта отображены на рисунке 15. Код программного комплекса вынесен в приложение В. Так как среда разработки TOAD не все отображает в выводе, поэтому мы можем видеть, что новая таблица создана и в нее добавлены данные из словаря, то есть произошел update в новую таблицу. А так же создан пакет, в котором находится наш код.



0 10 20  
1532630 rows updated.  
Package created.

Рисунок 15 – Результаты выполнения

## **5 ТЕСТИРОВАНИЕ**

### **5.1 Методология тестирования**

В рамках выпускной квалификационной работы было проведено альфа-тестирование. В дальнейшем в рамках задачи планируется провести бетта-тестирование и внедрение на фирме «Интерсвязь». При тестировании учитывались такие параметры, как:

- 1) быстроедействие;
- 2) работоспособность;
- 3) выполнение процедуры только в рамках отзыва в мобильном приложении;
- 4) закрытие некорректных тикетов, а так же не закрытие корректных;
- 5) корректное отображение в интерфейсе.

### **5.2 Проведение процедуры тестирования**

Для проведения тестирования рассмотрим данные на тестовом сервере с 01.03.2022 и приведем их на рисунках 16–19, чтобы впоследствии проанализировать изменения.

Новый

30.03.2022 20:55  
30.03.2022 20:55

Оценка: Плохо. Комментарий: не продуктивно Тикет №253246184

Новый

30.03.2022 20:46  
30.03.2022 20:46

Оценка: Плохо. Комментарий: часто пропадает трансляция Тикет №252359601

Новый

30.03.2022 20:15  
30.03.2022 20:15

Оценка: Плохо. Комментарий: удовлетворена ответом Тикет №252612230

Рисунок 16 – Список тикетов до выполнения процедуры (часть 1)



<p><b>Закрыт</b> 30.03.2022 20:55 27.05.2022 16:06</p>	<p>Закрыт автоматическі т.к. коментарый некорректен</p>
<p><b>Новый</b> 30.03.2022 20:46 30.03.2022 20:46</p>	<p>Оценка: Плохо. Комментарий: часто пропадает трансляция Тикет №252359601</p>
<p><b>Закрыт</b> 30.03.2022 20:15 27.05.2022 16:07</p>	<p>Закрыт автоматическі т.к. коментарый некорректен</p>

Рисунок 17 – Список тикетов после выполнения процедуры (часть 1)

<p><b>Новый</b> 23.03.2022 18:51 23.03.2022 18:51</p>	<p>Оценка: Плохо. Комментарий: пошло заонить Тикет №252917268</p>
---	---

Рисунок 18 – Список тикетов до выполнения процедуры (часть 2)

<p><b>Закрыт</b>  23.03.2022 18:51  27.05.2022 16:22</p>	<p>Закрыт автоматически т.к. комментарий некорректен</p>
--	--

Рисунок 19 – Список тикетов после выполнения процедуры (часть 2)

Как можно заметить на картинках выше, если комментарий действительно некорректен, то тикет закрывается с комментарием «Закрыт автоматически, т. к. комментарий не корректен».

Теперь перейдем к карточкам тикетов, выполним процедуру уже по новым данным и посмотрим, что изменится. На рисунках 20 и 21 будут отображены карточка №1 и карточка №2 до выполнения процедуры.

Проблема	
Зарегистрировано	30.03.2022 20:15:09 <a href="#">JOB</a>
Состояние	<b>Новый</b>
Источник	Мобильное приложение
Сервис	Отзыв
Админ.вопрос / Признак	-
Причина / Возражение	-
Результат	-
Приоритет	0
Закрыто	

#### Статистика по тикетам

#### Комментарии к тикету (2)

Добавлено	Комментарий
30.03.2022 20:15:09 <a href="#">JOB</a>	Работа с анлайком из мобильного приложения.
30.03.2022 20:15:09 <a href="#">JOB</a>	Оценка: Плохо. Комментарий: удовлетворена ответом Тикет №252612230

Рисунок 20 – Карточка №1 до выполнения процедуры

Проблема	
Зарегистрировано	23.03.2022 18:51:45 <a href="#">JOB</a>
Состояние	<b>Новый</b>
Источник	Мобильное приложение
Сервис	Отзыв
Админ.вопрос / Признак	-
Причина / Возражение	-
Результат	-
Приоритет	0
Закрывается	

### Статистика по тикетам

Комментарии к тикету (2)	
Добавлено	Комментарий
23.03.2022 18:51:45 <a href="#">JOB</a>	Работа с анлайком из мобильного приложения.
23.03.2022 18:51:45 <a href="#">JOB</a>	Оценка: Плохо. Комментарий: пошло заонить Тикет №252917268

Рисунок 21 – Карточка №2 до выполнения процедуры

На рисунках 22 и 23 будут отображены карточка №1 и карточка №2 после выполнения процедуры.

Проблема	
Зарегистрировано	30.03.2022 20:15:09 <a href="#">JOB</a>
Состояние	<b>Закрыт</b>
Источник	Мобильное приложение
Сервис	Отзыв
Админ.вопрос / Признак	-
Причина / Возражение	-
Результат	-
Приоритет	0
Закрыто	27.05.2022 16:07:00 <a href="#">JOB</a>

### Статистика по тикетам

Комментарии к тикету (3)	
Добавлено	Комментарий
30.03.2022 20:15:09 <a href="#">JOB</a>	Работа с анлайком из мобильного приложения.
30.03.2022 20:15:09 <a href="#">JOB</a>	Оценка: Плохо. Комментарий: удовлетворена ответом Тикет №252612230
27.05.2022 16:07:00 <a href="#">JOB</a>	Закрыт автоматически т.к. комментарий некорректен

Рисунок 22 – Карточка №1 после выполнения процедуры

Проблема	
Зарегистрировано	23.03.2022 18:51:45 <a href="#">JOB</a>
Состояние	<span style="background-color: green; color: white; padding: 2px;">Закрыт</span>
Источник	Мобильное приложение
Сервис	Отзыв
Админ.вопрос / Признак	-
Причина / Возражение	-
Результат	-
Приоритет	0
Закрыто	27.05.2022 16:22:30 <a href="#">JOB</a>

#### Статистика по тикетам

#### Комментарии к тикету (3)

Добавлено	Комментарий
23.03.2022 18:51:45 <a href="#">JOB</a>	Работа с анлайком из мобильного приложения.
23.03.2022 18:51:45 <a href="#">JOB</a>	Оценка: Плохо. Комментарий: поишлось заонить Тикет №252917268
27.05.2022 16:22:30 <a href="#">JOB</a>	Закрыт автоматически т.к. комментарий некорректен

Рисунок 23 – Карточка №2 после выполнения процедуры

**Вывод.** В ходе тестирования, программный комплекс работал верно. Все требования были соблюдены. Ошибок в работе не выявлено.

## ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы был реализован программный комплекс поиска не содержащих информации отзывов пользователей, а так же протестирован в интерфейсе фирмы Интерсвязь.

Для достижения данной цели, были выполнены следующие задачи:

- 1) создана статистика корректных отзывов;
- 2) разработана процедура по поиску осмысленных слов на основе имеющихся данных;
- 3) разработана документация по эксплуатации.

Так как программный комплекс разработан полностью, то теперь отзывы не будут распределяться сотрудникам службы технической поддержки до тех пор, пока он не пройдет проверку в процедуре и только при успешном прохождении проверки, попадет на исполнение к сотруднику. Данный программный комплекс закрывает примерно 20 процентов некорректных отзывов относительно всех отзывов, что существенно упростит работу отдела СТП (это можно увидеть на этапе тестирования), а при большом наплыве обращаемых процент будет значительно увеличиваться.

Была разработана документация, которая будет выдана всем сотрудникам.

В дальнейшем планируется проведение бета-тестирования и внедрение в фирме.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бондаренко, С. Десять причин перейти в Total Commander / Сергей Бондаренко, Мария Бондаренко. – Текст. Изображение (неподвижное ; двухмерное) : электронные // 3D news : [сайт]. – Март 2010. – URL: <https://3dnews.ru/588141> (дата обращения: 16.06.2022).

2. Google: особенности индексации и продвижения в системе, ее достоинства и недостатки. – Текст. Изображение (неподвижное ; двухмерное) : электронные // PR CY : [сайт]. – URL: <https://pr-cy.ru/lib/seo/Google-osobennosti-indeksatsii-i-prodvizheniya-v-sisteme-ee-dostoinstva-i-nedostatki> (дата обращения: 16.06.2022).

3. Multi Text Funder. – Текст. Изображение (неподвижное ; двухмерное) : электронные // Интересные решения : [сайт]. – URL: <https://interestingsolutions.ru/MultiTextFinder.aspx> (дата обращения: 16.06.2022).

4. @YevSam, Е. Какую СУБД выбрать и почему / Евгений @YevSam. – Текст. Изображение (неподвижное ; двухмерное) : электронные // habr : [сайт]. – URL: <https://habr.com/ru/post/579248/> (дата обращения: 16.06.2022).

5. Вичугова, А. Что не так с clickhouse: 10 главных недостатков / Анна Вичугова. – Текст. Изображение (неподвижное ; двухмерное) : электронные // bigdata : [сайт]. – URL: <https://www.bigdataschool.ru/blog/clickhouse-limitations.html> (дата обращения: 16.06.2022).

6. Колосов, А. Рассуждение на тему какую базу данных выбирать / Алексей Колосов. – Текст. Изображение (неподвижное ; двухмерное) : электронные // habr : [сайт]. – URL: <https://habr.com/ru/post/348220/> (дата обращения: 16.06.2022).

7. Oracle database 12c. – Текст. Изображение : электронные // Oracle : [официальный сайт]. – URL: <https://www.oracle.com/cis/database/12c-database/> (дата обращения: 16.06.2022).

8. Драч, В. Е. Сравнение современных СУБД / В. Е. Драч. – Текст : электронный // Молодой ученый : журнал. – 2017. – №1. – С. 2. – URL: <https://drach.pro/blog/hi-tech/item/145-db-comparison> (дата обращения: 16.06.2022).

9. Microsoft sql server: преимущества и недостатки. – Текст. Изображение (неподвижное ; двумерное) : электронные // Новости : [сайт]. – Октябрь 2021. – URL: <https://astv.ru/news/materials/microsoft-sql-server-preimushestva-i-nedostatki> (дата обращения: 16.06.2022).

10. Sql developer как пользоваться ПК Знаток. – Текст. Изображение (неподвижное ; двумерное) : электронные // Sql developer : [сайт]. – URL: <https://pcznatok.ru/kompjutery/sql-developer-kak-polzovatsja.html> (дата обращения: 16.06.2022).

11. Oracle SQL Developer. – Текст. Изображение (неподвижное ; двумерное) : электронные // Википедия : [сайт]. – Май 2022. – URL: [https://ru.wikipedia.org/wiki/Oracle\\_SQL\\_Developer](https://ru.wikipedia.org/wiki/Oracle_SQL_Developer) (дата обращения: 16.06.2022).

12. DataGrip. – Текст. Изображение : электронные // Jet Brains : [официальный сайт]. – URL: <https://www.jetbrains.com/ru-ru/datagrip/> (дата обращения: 16.06.2022).

13. String-searching algorithm. – Текст. Изображение (неподвижное ; двумерное) : электронные // Wikipedia : [сайт]. – URL: [https://en.wikipedia.org/wiki/String-searching\\_algorithm#Algorithms\\_using\\_a\\_finite\\_set\\_of\\_patterns](https://en.wikipedia.org/wiki/String-searching_algorithm#Algorithms_using_a_finite_set_of_patterns) (дата обращения: 16.06.2022).

14. Наивный алгоритм поиска подстроки в строке. – Текст. Изображение (неподвижное ; двумерное) : электронные // Университет ИТМО : [сайт]. –



Январь 2022. – URL:  
[https://neerc.ifmo.ru/wiki/index.php?title=Наивный\\_алгоритм\\_поиска\\_подстроки\\_в\\_строке](https://neerc.ifmo.ru/wiki/index.php?title=Наивный_алгоритм_поиска_подстроки_в_строке) (дата обращения: 16.06.2022).

15. Иванов, Ю. Алгоритмы поиска в строке / Юрий Иванов. – Текст. Изображение (неподвижное ; двухмерное) : электронные // Хабр : [сайт]. – Январь 2011. – URL: <https://habr.com/ru/post/111449/> (дата обращения: 16.06.2022).

16. Bitap algorithm. – Текст. Изображение (неподвижное ; двухмерное) : электронные // Wikipedia : [сайт]. – Январь 2022. – URL: [https://en.wikipedia.org/wiki/Bitap\\_algorithm](https://en.wikipedia.org/wiki/Bitap_algorithm) (дата обращения: 16.06.2022).

17. Двусторонний алгоритм сопоставления строк. – Текст. Изображение (неподвижное ; двухмерное) : электронные // Wikipedia : [сайт]. – Май 2022. – URL: [https://en.wikipedia.org/wiki/Two-way\\_string-matching\\_algorithm](https://en.wikipedia.org/wiki/Two-way_string-matching_algorithm) (дата обращения: 16.06.2022).

18. Алгоритм обратного недетерминированного сопоставления Dawg. – Текст. Изображение (неподвижное ; двухмерное) : электронные // igm : [сайт]. – URL: <https://www-igm.univ-mlv.fr/~lecroq/string/bndm.html> (дата обращения: 16.06.2022).

19. FM-index. – Текст. Изображение (неподвижное ; двухмерное) : электронные // Wikipedia : [сайт]. – Март 2022. – URL: <https://en.wikipedia.org/wiki/FM-index> (дата обращения: 16.06.2022).

## ПРИЛОЖЕНИЕ А

### Исходный код анализа статистики

#### Листинг А – Анализ статистики

```
DECLARE
    word_all_v          VARCHAR2 (4000);
    cnt_word_v          NUMBER;
    cnt_all_v           NUMBER;
    cnt_uncorrect_word_v NUMBER;
    cnt_correct_word_v  NUMBER;
    itog_less_75_v     NUMBER;
    n                   NUMBER;
    cnt_less_2_v       NUMBER;
    cnt_is_null_v      NUMBER;
    itog_less_2_v      NUMBER;
    itog_is_null_v     NUMBER;
BEGIN
    -- Создадим цикл
    -- Входим в цикл если нашли комметарий из мобильного
приложения,
    -- который был создан за последний месяц
    -- Так же считаем количество слов в самом комментари и
ВЫВОДИМ
    -- его длину
    FOR rec
        IN (SELECT regexp_count (
                lower (
                    substr (e.comments, instr (e.comments,
                        'Комментарий: '), length (e.comments))
                ),
                '[^[a-я]+[^\ ]+')

```

```

    )
  - 2
    AS wordallcount,
    e.id AS comments,
    length (e.comments) AS cnt
FROM regsysref.sd_ticket_comment e
WHERE   e.comments LIKE
        '%Оценка%'
        || '%Комментарий%'
        AND e.ins_date >= trunc (sysdate) - 30) LOOP
-- установим счетчик в 1, просто чтобы сравнение было
-- корректное
n := 1;

WHILE n != rec.cnt LOOP
  SELECT DISTINCT
    regexp_substr (lower (substr (e.comments, instr
      (e.comments, 'Комментарий:'), length (e.comments) -
      56)),
    '[^[a-я]+[^\ ]+', n)
  INTO word_all_v
  FROM regsysref.sd_ticket_comment e
  WHERE e.id = rec.comments AND rec.wordallcount > 2;

  n := n + 1;

IF word_all_v IS NOT NULL THEN
  -- Затем сравниваем это последовательность символов -
  -- называемое словом со словом из словаря
  -- Если в словаре, такое слово имеется, то увеличиваем
  -- количество на единицу

```

```

SELECT count (*)
  INTO cnt_word_v
  FROM kseniyaz.dictionary_correct_word@testmain dcw
  WHERE dcw.word = word_all_v;
END IF;

-- Здесь считаем проент корректных слов
cnt_all_v := (cnt_word_v / rec.cnt) * 100;

      IF cnt_all_v < 75 THEN
          cnt_uncorrect_word_v :=
              cnt_uncorrect_word_v + 1;
      ELSE
          cnt_correct_word_v := cnt_correct_word_v + 1;
      END IF;

-- считаем процент всех некорректных слов, относительно
-- всех слов
itog_less_75_v := (cnt_uncorrect_word_v /
  (cnt_uncorrect_word_v + cnt_correct_word_v)) * 100;
END LOOP;
END LOOP;

-- Здесь сразу считаем комментарии, где количество слов не
более
-- двух, а так же количество пустых комментариев
-- И количество всех комментариев
SELECT count (
  CASE
    WHEN t.wordallcount <= 2 AND wordallcount > 0 THEN
      comments
  END

```

```

),
count (
    CASE
        WHEN t.wordallcount = 0 THEN comments
    END
),
count (*)
INTO cnt_less_2_v,
    cnt_is_null_v,
    cnt_all_v
FROM (SELECT regexp_count (
        lower (
            substr (e.comments, instr (e.comments,
                'Комментарий: '), length (e.comments))
        ),
        '^[^а-я][^ ]+'
    )
    - 2
    AS wordallcount,
    e.id AS comments
FROM regsysref.sd_ticket_comment e
WHERE e.comments LIKE
        '%Оценка%'
        || '%Комментарий%'
    AND e.ins_date >= trunc (sysdate) - 30) t;
-- считаем процент всех некорректных слов, относительно всех
слов
itog_less_2_v := (cnt_less_2_v / cnt_all_v) * 100;
itog_is_null_v := (cnt_is_null_v / cnt_all_v) * 100;
COMMIT;
END;

```

## ПРИЛОЖЕНИЕ Б

### Исходный код обработки отзывов

#### Листинг В – Код программы

```
--Создадим таблицу слов из словаря
CREATE TABLE kseniyaz.dictionary_correct_word (word VARCHAR2
(400));

--Выдадим права на таблицу
GRANT SELECT ON kseniyaz.dictionary_correct_word TO
user_for_db_link_to_testdb;

ALTER TABLE kseniyaz.dictionary_correct_word
  ADD word_id NUMBER NOT NULL;

--Создадим последовательность
CREATE SEQUENCE kseniyaz.dictionary_correct_word_1_seq;

UPDATE kseniyaz.dictionary_correct_word
  SET word_id =
kseniyaz.dictionary_correct_word_1_seq.nextval;
CREATE OR REPLACE PACKAGE BODY regsysref.service_desk_pkg
/*****
*****
NAME:   service_desk_pkg
PURPOSE: Все для работы Service Desk
SET DEFINE OFF;

...
*****
*****/
IS
...
```

## Продолжение приложения Б

```
PROCEDURE merge_sd_ticket_comment_prc (  
    oper_in          IN   VARCHAR2,  
    id_inout        IN OUT NUMBER,  
    change_user_id_in IN   NUMBER DEFAULT  
    const.default_number_ct,  
    ticket_id_in    IN   NUMBER,  
    comments_in     IN   VARCHAR2,  
    visible_type_in IN   VARCHAR2 DEFAULT NULL,  
    result_out      OUT NUMBER,  
    message_out     OUT VARCHAR2)  
  
-----  
-----  
-- Добавление комментариев к тикетам. Таблица sd_ticket_comment  
-----  
-----  
-- oper_in          - Производимая операция: INS - вставка  
-- id_inout        - ид записи в таблице комментариев  
-- change_user_id_in - кто делает изменения (иногда явно  
приходит  
-- из интерфейса)  
-- ticket_id_in    - ид тикета  
-- comments_in     - комментарий  
-- visible_type_in - тип видимости коммента (null - видно для  
-- всех, ms - только для управляющих компаний, ..)  
-----  
-----  
-- result_out - 0 если все ок, иначе код ошибки  
-- message_out - описание ошибки  
-----  
-----IS  
    change_user_id_v  NUMBER;
```

```
sysdate_v      DATE := sysdate;
reg_date_v     DATE;
word_all_v     VARCHAR2 (4000);
  cnt_word_v   NUMBER;
  n            NUMBER;
  result_v     NUMBER;
message_v      VARCHAR2 (4000);
BEGIN
  result_out := 0;
  IF oper_in <> 'INS' THEN
    result_out := 1;
    message_out := 'Неверная операция';
    RETURN;
  END IF;
  IF comments_in IS NULL OR ticket_id_in IS NULL THEN
    result_out := 2;
    message_out := 'Не заданы обязательные параметры';
    RETURN;
  END IF;
  IF change_user_id_in = -1 THEN
    change_user_id_v := a_dba.env_pkg.get_user_id_func ();
  ELSE
    change_user_id_v := change_user_id_in;
  END IF;
  SELECT max (reg_date)
  INTO reg_date_v
  FROM sd_ticket t
 WHERE t.id = ticket_id_in;

  IF reg_date_v IS NULL THEN
    result_out := 3;
```



```

message_out := 'Нет такого тикета';
RETURN;
END IF;
CASE oper_in
WHEN 'INS' THEN
  id_inout := sd_ticket_comment_seq.nextval;
  INSERT INTO sd_ticket_comment (id,
                                ticket_id,
                                reg_date,
                                ins_user_id,
                                ins_date,
                                comments,
                                visible_type)
    VALUES (id_inout,
            ticket_id_in,
            reg_date_v,
            change_user_id_v,
            sysdate_v,
            comments_in,
            visible_type_in);
  UPDATE sd_ticket t
    SET t.change_date = sysdate_v
    WHERE id = ticket_id_in;
END CASE;
--Закроем тикеты с некорректными комментариями
-- Создадим цикл
-- Входим в цикл, если нашли комментарий из мобильного
-- приложения, который был создан за последние 6 месяцев
-- Так же считаем количество слов в самом комментарии,
ВЫВОДИМ
-- его длину, id тикета и id самого комментария

```

```

FOR rec
  IN (SELECT regexp_count (
        lower (
          substr (
            e.comments,
            instr (e.comments, 'Комментарий: '),
            length (e.comments)
          )
        ),
        '[^[a-я]+[^\ ]+'
      )
    - 2
    AS wordallcount,
    e.id AS comments,
    length (e.comments) AS cnt,
    e.ticket_id
  FROM regsysref.sd_ticket_comment e
  WHERE e.comments LIKE
        '%Оценка%'
        || '%Комментарий%'
    AND e.ins_date >= add_months (trunc (sysdate,
        'mm'), -6)) LOOP
-- Если количество слов в комментарии больше двух, найдем
-- процент корректных слов
IF rec.wordallcount > 2 THEN
  n := 1;
  -- Для начала выбираем символы, пока не дойдем до
  -- пустого значения (пробела)
  WHILE n != rec.cnt LOOP
    SELECT DISTINCT
      regexp_substr (lower (substr (e.comments,

```

```

instr
(e.comments, 'Комментарий:'), length
(e.comments) - 56)), '[^[a-я]+[ ^ ]+', n)
INTO word_all_v
FROM regsysref.sd_ticket_comment e
WHERE e.id = rec.comments;
n := n + 1;
IF word_all_v IS NOT NULL THEN
-- Затем сравниваем это последовательность
-- символов - называемое словом со словом из
-- словаря. Если в словаре, такое слово имеется,
-- то увеличиваем количество на единицу
SELECT count (*)
INTO cnt_word_v
FROM kseniyaz.dictionary_correct_word@testmain
dcw
WHERE dcw.word = word_all_v;
END IF;
-- Если процент корректных слов, меньше 75,
-- закрываем тикет с комментарием
IF ( (cnt_word_v / rec.cnt) * 100) < 75 THEN
regsysref.service_desk_pkg.merge_ticket_prc (
oper_in      => 'UPD',
id_inout     => rec.ticket_id,
state_in     => 3, -- закрыт
comments_in  => 'Закрыт автоматически
т.к. комментарий некорректен',
change_user_id_in => const.user_job_ct, --JOB
result_out   => result_v,
message_out  => message_v
);

```

## Окончание приложения Б

```
        END IF;
    END LOOP;
ELSE
    -- Если количество слов в отзыве меньше двух, либо слов
    -- нет, либо там только знаки препинания, либо цифры
    -- В таких случаях тикет будет закрыт
    regsysref.service_desk_pkg.merge_ticket_prc (
        oper_in          => 'UPD',
        id_inout         => rec.ticket_id,
        state_in         => 3, -- закрыт
        comments_in      => 'Закрыт автоматически т.к.
        комментарий некорректен',
        change_user_id_in => const.user_job_ct, --JOB
        result_out       => result_v,
        message_out      => message_v
    );
    END IF;
    END LOOP;
END merge_sd_ticket_comment_prc;
...
END service_desk_pkg;
/
```