Федеральное государственное бюджетное образовательное учреждение высшего образования

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**Кафедра информационных технологий**

Пояснительная записка к курсовой работе

по дисциплине «Современные технологии программирования»

на тему:

«Информационная система учета обращений пользователей в техническую поддержку»

Выполнил(а):

студент(ка) группы ДПИ22-1с институт открытого образования Берникова В.С.

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

Научный руководитель:

Кандидат технических наук Пальчевский Е.В.

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

2024 г

# **СОДЕРЖАНИЕ**

[СОДЕРЖАНИЕ 2](#_Toc194758461)

[ВВЕДЕНИЕ 3](#_Toc194758462)

[ОПИСАНИЕ ПРОГРАММЫ 4](#_Toc194758463)

[1.1 Алгоритмические решения 4](#_Toc194758464)

[1.1.1 Безопасность 4](#_Toc194758465)

[1.1.2 Клиент 5](#_Toc194758466)

[1.1.3 Сервер 6](#_Toc194758467)

[1.1.4 База данных 7](#_Toc194758468)

[1.1.5 Зависимости программы 8](#_Toc194758469)

[1.2 Описание интерфейса программы 9](#_Toc194758470)

[1.2.1 Авторизация (Регистрация) 9](#_Toc194758471)

[1.2.2 Личный кабинет 13](#_Toc194758472)

[1.2.3 Панель администратора 18](#_Toc194758473)

[ЗАКЛЮЧЕНИЕ 20](#_Toc194758474)

[СПИСОК ИСТОЧНИКОВ 21](#_Toc194758475)

[ПРИЛОЖЕНИЯ 23](#_Toc194758476)

# **ВВЕДЕНИЕ**

Безусловно, темпы развития технологий, задаваемые нынешним временем - не только быстры, но и растут с каждым днём. Никакая компания сегодня не может существовать без IT-решения будь то сайт, приложение или веб-приложение. И конечно, при взаимодействии пользователей с таким IT-решением, будут появляться технические проблемы, из-за которых пользователи будут обращаться за помощью. Хорошо организованная система технической поддержки пользователей – залог успешного взаимодействия организации с клиентами, потому что все любят, когда их проблемы решают быстро и четко.

Таким образом, было решено разработать информационно-справочную систему типа ServiceDesk для регулирования коммуникации клиентов организации с технической поддержкой. Система подобного типа предоставляет инструменты для эффективной организации и приоритезации заявок. Более того, включает в себя оперативную и управленческую отчетность. [11]

Следуя из всего выше сказанного, целью текущей курсовой работы является реализация системы учёта обращений пользователей в техническую поддержку.

Достижение цели планируется по средствам выполнения нескольких этапов:

* создание серверной части,
* создание клиентской части,
* создание базы данных.

# **ОПИСАНИЕ ПРОГРАММЫ**

## **Алгоритмические решения**

В текущем разделе будут представлены описания использованных в процессе разработке решений.

### **1.1.1** **Безопасность**

Безопасность – неотъемлемая часть любого IT-решения. Поэтому ей было уделено отдельное внимание в рамках работы над данной информационно-справочной системой.

Было решено использовать Spring Security, являющуюся надежной платформой, поднимающей безопасность веб-приложений. Особенно на базе Spring. Функции типа аутентификация, авторизация и защита от распространенных атак предоставляются. [5]

Три основные концепции, на базе которых работает Spring Security, можно представить в виде списка:

* аутентификация (установление личности).
* авторизация (контроль доступа).
* защита (от веб-уязвимостей). [5]

Spring Security добавляет цепочки фильтров в приложение. Эти фильтры перехватывают HTTP-запросы и применяют правила. Современный подход к определению правил безопасности в Spring Security заключается в использовании SecurityFilterChain компонента. [5]

### **1.1.2 Клиент**

Клиентская часть приложения – это та часть приложения, которая видна клиенту, и с которой клиент, собственно, взаимодействует. Поэтому для создания клиентской части были использованы инструменты разработки, подходящие к выбранному веб-модулю Spring, а именно:

* HTML,
* CSS,
* JavaScpirt,
* Bootstrap,
* Thymeleaf.

HTML самый популярный среди языков гипертекстовой разметки, к которому чаще всего прибегают разработчики. Потому что он состоит из элементов или тегов, которые определяют различные части веб-страницы. [7]

Каскадные таблицы стилей CSS будут нужны для внешнего оформления и настройки стилей, составляющих экран HTML-объектов. [8]

И хотя основная часть текущий работы будет выполнена с помощью Java, при работе с HTML так или иначе затрагивается и JavaScript, скрипты которого можно удобно вставлять прямо в гипертекстовую разметку. [2]

Для большей оптимизации и облегчения работы над дизайнерской частью веб-приложения было решено использовать Bootstrap, который содержит все необходимые компоненты и шаблоны веб-интерфейса. [9]

### **1.1.3 Сервер**

Серверную часть было решено разрабатывать, используя веб-модуль Spring Boot. Потому что, обладая большим функционалом, он позволяет создавать web-приложения, не требуя от программиста отдельных усилий для его настройки, что позволяет сконцентрироваться на самом главном. [1]

### **1.1.4 База данных**

В последнее время база данных PostgreSQL обогнала MySql по популярности и частоте выбора среди разработчиков. Обусловлено это явление расширяемостью, масштабируемостью и кроссплатформенностью. Таким образом, было решено использовать PostgreSQL для реализации части базы данных [14].

Всё необходимое для работы описание параметров, без которых взаимодействие приложения с системой управления базами данных PostgreSQL было бы невозможным, находится в файле конфигураций «application.properties».

В самом начале вышеописанного файла задано имя приложения через параметр spring.application.name. Ему было присвоено значение «demo1».

Далее, с помощью параметра «spring.datasource.url» задаётся подключение к PostgreSQL.

И наконец, с помощью параметров «spring.datasource.name» и «spring.datasource.password» задаются данные для доступа к базе данных.

### **1.1.5 Зависимости программы**

Управление зависимостями – непростая задача, особенно если речь идёт о больших проектах. Однако Spring Boot решает эту проблему путём предоставления набора зависимостей, облегчая жизнь разработчикам. [13]

В текущем проекте зависимости управляются с использованием файла build.gradle. Зависимости, использованные в программе, можно представить в виде списка:

* spring-boot-starter-thymeleaf (для поддержки шаблонизаторов),
* spring-boot-starter-security (для обеспечения безопасности),
* spring-boot-starter-web-services (для работы с веб-службами),
* spring-boot-starter-data-jpa (для работы с даннымми),
* драйвер postgresql (для работы с даннымми),
* lombok (для упрощения кода).

## **Описание интерфейса программы**

Текущий раздел предоставляет полное описание интерфейса информационно-справочной системы.

### **1.2.1 Авторизация (Регистрация)**

При запуске программы пользователя встречает страница авторизации. На рисунке 1 представлена страница авторизации.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 1 – Страница авторизации

Стандартная авторизация производится за счет ввода логина и пароля пользователя. Здесь стоит уделить отдельной внимание реализованной функции уведомления пользователя о неправильном вводе данных авторизации. Если пользователь ввел неверные данные, он увидит сообщение, представленное на рисунке 2.

Изображение выглядит как текст, Шрифт, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 2 – Сообщение об ошибке

У каждого пользователя своя роль, «ROLE\_ADNIN» или «ROLE\_USER» и в зависимости от роли, пользователь будет перенаправлен на соответствующую страницу.

Если у пользователя нет аккаунта, то он может зарегистрироваться в системе, используя ссылку под кнопкой войти. После попадания на страницу регистрации пользователь увидит форму для заполнения. Форма для регистрации представлена на рисунке 3.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 3 – Страница регистрации

После успешной регистрации пользователь может авторизоваться в системе и начать работу. Каждому новому пользователю при регистрации выдаётся роль «ROLE\_USER», которую в дальнейшем при необходимости администратор может изменить.

Дополнением на страницы авторизации является раздел «об авторе», куда пользователь может перейти, нажав соответствующую ссылку.

Оказавшись на странице, представленной на рисунке 4, пользовать может ознакомится с актуальной информацией об авторе проекта.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4 – Об авторе

### **1.2.2 Личный кабинет**

Личный кабинет – это первая страница, которую видит пользователь, авторизовавшись на информационно-справочной системе обращения пользователей в тех. поддержку.

В своём личном кабинете пользователь имеет возможность управлять своими заявками, просматривать их статус, создавать новые и редактировать существующие.

Конечно же функционал, а соответственно и интерфейс пользователей отличается в зависимости от роли. В верхнем правом угру находится информация о текущем пользователе, и кнопка «Выйти» которая предоставляет возможность покинуть текущий раздел информационной системы.

Верхняя часть страницы, представляет название раздела, оповещая пользователя о том, где он в данный момент находится.

Далее мы имеем несколько функциональных блоков.

Кнопка «Добавить заявку» перенаправляет пользователя на страницу создания заявки, представленную на рисунке 7.

Далее, как можно видеть, идёт поисковый блок. Он включает в себя поисковую строку, кнопку «Найти» и кнопку «Очистить».

При введение поискового запроса в поисковую строку и нажатии на кнопку «Найти» пользователь получает результаты из базы данных. Поддерживается полное и частичное совпадение.

На рисунке 5 представлена часть интерфейса личного кабинета пользователя.

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 5 – Личный кабинет, верхняя часть страницы

В центральной/средней части, представленной на рисунке 6 (страница от лица пользователя с ролью «ROLE\_USER»), располагается таблица, в которую происходит вывод всех заявок из базы данных.

Каждая заявка состоит из следующего набора информации, позволяющего быстро взаимодействовать с таблицей:

* номер заявки (уникальный),
* тема заявки,
* дата создания,
* дата изменения,
* статус (стадия работы с заявкой),
* описание заявки (сообщение прикрепленное при создании).

В отдельную колонку вынесена кнопка «Редактировать», существующая для каждой заявки. При нажатии на эту кнопку пользователь попадает на страницу редактирования, представленную на рисунке 8.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 6 – Личный кабинет, полный скриншот

Упомянутая выше страница создания заявки, представленная на рисунке 7, позволяет пользователю создать новую заявку с обращением. Здесь можно указать тему обращения и подробно описать в чем именно заключается вопрос. Дата создания выставляется пользователем самостоятельно. Статус при создании выставляется как «Новая», что логично.

Кнопка «На главную» даёт возможность выйти из текущего раздела.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 7 – Создание заявки

Страница «Редактирование заявки», как видно на рисунке 8, даёт возможно изменить некоторые параметры заявки включая даты изменения статус, описание, что логично. Текущая страница создана для внесения изменений в существующую в базе данных заявку.

Кнопка «На главную» даёт возможность выйти из текущего раздела.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 8 – Редактирование заявки

### **1.2.3 Панель администратора**

Зайдя в систему от лица пользователя «ROLE\_ADMIN», появляется дополнительная функция удаления существующих заявок, реализованная с помощью кнопки «Удалить» в колонке «Действия» основной таблицы в личном кабинете (рисунок 9).

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9 – Личный кабинет пользователя «ADMIN»

Также у администратора есть отдельный раздел, где храниться информация о всех зарегистрированных пользователях системы.

Страница представляет собой таблицу, которая состоит из следующего набора информации:

* email (уникальный),
* номер телефона,
* активность,
* роли,
* заблокировать пользователя.

В колонке «Email», находится электронные почты всех зарегистрированных пользователей. В колонке «Номер телефона», номера пользователей, в колонке «Активность», имеет ли пользователь доступ к системе, в колонке «Роли», права доступа, ROLE\_USER – для обычных пользователей и ROLE\_ADMIN – для администраторов системы. Последняя колонке «Заблокировать пользователя», позволяет администратору заблокировать пользователя. Панель администратора представлена на рисунке 10.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 10 – Панель администратора

# **ЗАКЛЮЧЕНИЕ**

В заключение стоит сказать, что в ходе выполнения текущей курсовой работы поставленная цель была достигнута.

Была разработана информационная система учета обращений пользователей в техническую поддержку типа ServiceDesk для организации коммуникации клиентов организации с технической поддержкой.

Были представлены подробные описания хода выполнения каждого этапа, а именно:

* описания этапа выбора инструментов для создания серверной части и создания серверной части;
* описание этапа выбора инструментов для создания клиентской части создания клиентской части;
* описания этапа выбора и создания базы данных.

Помимо этого, были закреплены практические навыки проектирования и разработки веб-приложения с нуля, используя SpringBoot, Java и HTML,CSS, JS и PostgreSQL.

В дальнейшем планируется продолжение работы над системой в рамках дипломного проекта.

# **СПИСОК ИСТОЧНИКОВ**

1. Введение в Spring Boot: создание простого REST API на Java – [Электронный ресурс]. – URL: https://habr.com/ru/articles/435144/ (дата обращения:20.02.2025). – Текст: электронный.
2. Что такое JavaScript – [Электронный ресурс]. – URL: https://digitalocean.ru/n/chto-takoe-javascript (дата обращения: 20.02.2025). – Текст: электронный.
3. Thymeleaf – [Электронный ресурс]. – URL: https://www.thymeleaf.org/ (дата обращения: 20.02.2025). – Текст: электронный.
4. Spring Boot – [Электронный ресурс]. – URL: https://spring.io/projects/spring-boot (дата обращения: 20.02.2025). – Текст: электронный.
5. Защита приложения Spring MVC с помощью Spring Security – [Электронный ресурс]. – URL: https/www.geeksforgeeks.org/securing-a-spring-mvc-application-with-spring-security/ (дата обращения: 20.02.2025). – Текст: электронный.
6. Регистрация и авторизация с помощью Spring Security на примере простого приложения – [Электронный ресурс]. – URL: https://habr.com/ru/articles/482552/ (дата обращения: 20.02.2025). – Текст: электронный.
7. Почему HTML называется Языком разметки? – [Электронный ресурс]. – URL: https/www.geeksforgeeks.org/why-html-is-called-markup-language/ (дата обращения: 20.02.2025). – Текст: электронный.
8. Типы CSS (каскадная таблица стилей) – [Электронный ресурс]. – URL: https/www.geeksforgeeks.org/types-of-css-cascading-style-sheet/ (дата обращения: 20.02.2025). – Текст: электронный.
9. Bootstrap – [Электронный ресурс]. – URL: https://blog.skillfactory.ru/glossary/bootstrap/?ysclid=m7lq7h6wgy14424758 (дата обращения: 20.02.2025). – Текст: электронный.
10. Запускаем первое веб-приложение на Spring Boot – [Электронный ресурс]. – URL: https://skillbox.ru/media/code/zapuskaem-pervoe-vebprilozhenie-na-spring-boot/ (дата обращения: 20.02.2025). – Текст: электронный.
11. ServiceDesk - cистема учета обращений – [Электронный ресурс]. – URL: https://yandex.ru/q/article/servicedesk (дата обращения: 20.02.2025). – Текст: электронный.
12. Драйвер PostgreSQL JDBC – [Электронный ресурс]. – URL: https://www.geeksforgeeks.org/postgresql-jdbc-driver/ (дата обращения: 03.03.2025). – Текст: электронный.
13. Зависимости в Spring Boot – [Электронный ресурс]. – URL: https://otus.ru/nest/post/1372/ (дата обращения: 03.03.2025). – Текст: электронный.
14. PostgreSQL (база данных) – [Электронный ресурс] – URL: https://skillbox.ru/media/code/postgresql-vsye-chto-nuzhno-znat-dlya-bystrogo-starta/ (дата обращения 20.02.2025) – Текст: электронный

# **ПРИЛОЖЕНИЯ**

Приложение А – логика контроллера заявок.

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Приложение Б – логика контроллера администратора.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

Приложение В – логика контроллера пользователя.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Приложение Г – логика html страницы пользователей.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Приложение Д – логика html страницы администратора.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Приложение Е – логика html страницы редактирования заявок.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Приложение Ж – логика html страницы создания заявок.

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Приложение З – логика репозитория заявок.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Приложение И – логика сущности заявок.

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Приложение К – логика сервиса заявок.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Приложение Л – логика сущности пользователей.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Контент, сгенерированный ИИ, может содержать ошибки.

Приложение М – логика сервиса пользователей.

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Приложение Н – логика безопасности приложения.

Изображение выглядит как текст, снимок экрана, программное обеспечение, компьютер

Контент, сгенерированный ИИ, может содержать ошибки.

Приложение О – логика html страницы входа в приложение.

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.