Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве**

**Российской Федерации»**

Департамент анализа данных и машинного обучения

Курсовая работа по дисциплине

«Современные технологии программирования»

на тему:

**«Разработка информационно-справочного веб-сервиса с использованием библиотеки Spring Boot»**

Выполнила:

студент ПИ21-6

Епифанова Кристина

**Москва**

**2023**

ОГЛАВЛЕНИЕ

[I. ВВЕДЕНИЕ 3](#_Toc133877043)

[II. ПОСТАНОВКА ЗАДАЧИ 4](#_Toc133877044)

[III. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc133877045)

[IV. АКТУАЛЬНОСТЬ АВТОМАТИЗАЦИИ 6](#_Toc133877046)

[V. АЛГОРИТМИЧЕСКИЕ РЕШЕНИЯ 7](#_Toc133877047)

[Клиент 7](#_Toc133877048)

[Сервер 7](#_Toc133877049)

[VI. ОПИСАНИЕ ИНТЕРФЕЙСА ПРОГРАММЫ 9](#_Toc133877050)

[Окно авторизации клиента 9](#_Toc133877051)

[Окно регистрации нового пользователя 10](#_Toc133877052)

[Главное окно программы 10](#_Toc133877053)

[Окно добавления нового заказа 11](#_Toc133877054)

[Окно изменения товара 12](#_Toc133877055)

[VII. СОСТАВ ПРИЛОЖЕНИЯ 15](#_Toc133877056)

[Сервер 15](#_Toc133877057)

[База данных 15](#_Toc133877058)

[Клиент 17](#_Toc133877059)

[VIII. НАЗНАЧЕНИЕ И СОСТАВ КЛАССОВ ПРОГРАММЫ 18](#_Toc133877060)

[IX. ЗАКЛЮЧЕНИЕ 21](#_Toc133877061)

[X. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 23](#_Toc133877062)

[XI. ПРИЛОЖЕНИЯ 24](#_Toc133877063)

# I. ВВЕДЕНИЕ

В условиях увеличивающейся конкуренции на рынке, компаниям необходимо управлять цепь поставок эффективно и надежно. Цепь поставок включает в себя поставщиков, производство, складирование, транспортировку и доставку продукции клиентам. Для успешного управления цепями поставок необходима информационно-справочная система (ИСС), которая позволяет собирать, хранить и обрабатывать данные обо всех этапах цепи поставок. ИСС предоставляет компаниям инструменты управления процессом поставок и обеспечивает оперативное реагирование на изменения внешних факторов.

Иной вариант проекта курсовой работы связан с созданием программного продукта - ИСС управления цепями поставок, основанного на современных информационных технологиях. Работа включает изучение требований к системе, методов и технологий ее проектирования, также рассмотрение способов ее реализации и интеграции в существующую инфраструктуру компании. Информационно-справочная система управления цепями поставок облегчит и оптимизирует процессы и основана на использовании информационных технологий. Компании смогут быстро реагировать на изменения рынка и оперативно принимать решения, которые помогут повысить эффективность работы и сократить затраты. Расмотрен пример определенной компании, где рассмотрены основные функции системы, а также проведен анализ и оценка эффективности системы.

# II. ПОСТАНОВКА ЗАДАЧИ

Для выполнения задания необходимо разработать клиент-серверное решение на основе библиотеки Spring Boot, используя пользовательские классы и таблицы в СУБД. В приложении на стороне клиента необходимо создать несколько окон с интерфейсом для взаимодействия с пользователем и логикой переходов между окнами. Для связи Spring с СУБД следует использовать ORM, а для размещения контроллеров, сервисов и репозиториев с логикой таблиц СУБД на стороне сервера - модель MVC.

Разработанное приложение должно выполнять следующие функции: отображать данные предметной области в таблице, добавлять, удалять и редактировать объекты в БД, фильтровать и сортировать записи, обновлять изменения в БД, загружать данные из БД и отображать статистические данные. При этом приложение должно корректно обрабатывать ошибки: отображать сообщения об ошибках на странице, а не приводить к аварийному завершению приложения в случае некорректного ввода данных, противоречивых или недопустимых значений данных, отсутствии данных по запросу пользователя и других нештатных ситуациях.

# 

# III. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Предметная область автоматизируется через веб-сервис с простой логикой: пользователь проходит авторизацию в информационно-справочной системе, чтобы получить доступ к таблице в базе данных. Для администраторов доступны функции добавления, удаления и редактирования записей. Всего используется 2 сущности для представления данных со стороны клиента: Operation - описывающая товары, и Client – описывающая поставки.

С серверной стороны, для корректной регистрации и авторизации в системе, объявляется сущность UserInfo, содержащая информацию о пользователях, администраторах и дилерах веб-сервиса. Подробные характеристики сущностей приводятся в разделе "База данных" главы "Состав приложения"

# IV. АКТУАЛЬНОСТЬ АВТОМАТИЗАЦИИ

Автоматизация управления цепями поставок является ключевой задачей многих компаний в разных секторах экономики, особенно в условиях быстро меняющейся экономической и технологической ситуации в мире. Модернизированные аспекты в обеспечении услуг и товаров требуют более быстрой и точной обработки заказов, управления запасами установки мониторинга, анализа и улучшения эффективности работы цепей поставок.

В этом контексте курсовая работа на тему разработки информационно-справочной системы управления цепями поставок становится важной задачей. Такая система помогает автоматизировать процессы управления цепями поставок, упрощает доступ к информации, снижает число ошибок и повышает эффективность использования ресурсов компании. Система может обладать модулями управления заказами, контроля запасов, мониторинга и анализа производительности поставщиков и управления производственными мощностями.

Кроме того, следует отметить, что разработка информационно-справочной системы управления цепями поставок привлекательна для курсовой работы, так как требует знаний и опыта в области программирования и баз данных, а также позволяет изучить методы проектирования и разработки сложных систем. В рамках курсовой работы возможно рассмотреть теоретические и практические аспекты разработки информационных систем, а также осуществить реализацию клиент-серверного решения на основе Spring Boot и использования ORM для связи с БД.

# V. АЛГОРИТМИЧЕСКИЕ РЕШЕНИЯ

## Клиент

Этот код описывает функционирование сервера, который использует Spring и сокеты. При поступлении запроса сервер анализирует его URL и его соответствие с функциями контроллеров, чтобы определить подходящую функцию для обработки. Если сервер не может обработать запрос, то он возвращает ошибку 404 (NOT FOUND). Если функция контроллера существует, но запрос не соответствует ожидаемым параметрам, то сервер возвращает ошибку 400 (BAD REQUEST). В случае соответствия запроса параметрам контроллера, сервер вызывает соответствующий метод репозитория JPA, чтобы извлечь данные из базы данных. Эти данные обрабатываются и отправляются клиенту.

Этот механизм реализует функции регистрации и аутентификации пользователей. При регистрации пользователь предоставляет информацию о себе в запросе, которая затем проверяется и сохраняется в базе данных. Зашифрованный ключ также генерируется и сохраняется в БД, который в дальнейшем используется при авторизации пользователя в системе. При получении запроса сервер проверяет правильность введенного пользователем пароля и, если он неправильный, отправляет сообщение об ошибке.

## Сервер

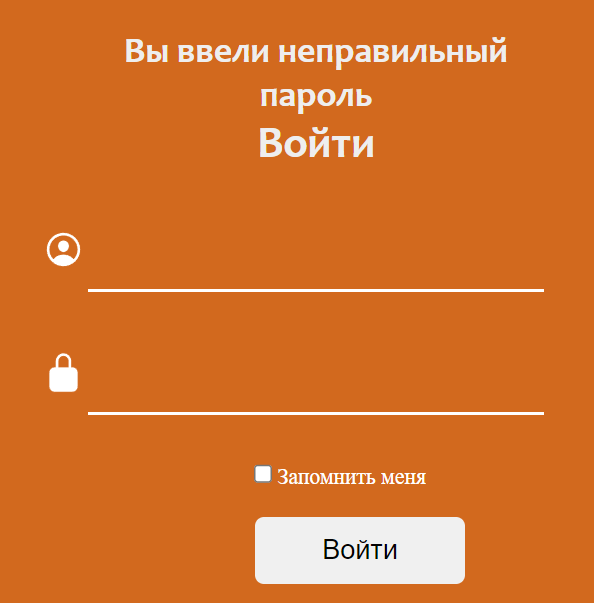
При получении запроса от клиента, сервер сначала проверяет наличие контроллера, который соответствует переданному URL. Если контроллер отсутствует, сервер возвращает ошибку 404 "not found". В противном случае, Spring анализирует переданные данные и их соответствие требованиям аргументов контроллера. Если данные не корректны, сервер отправляет ошибку 400 "bad request". Если же данные верны, контроллер вызывает сервис, который содержит логику обработки данных, а затем сервис вызывает методы репозитория с полученной информацией. После получения данных, сервис обрабатывает их и возвращает обратно к контроллеру. Контроллер проводит дополнительную обработку и отправляет ответ клиенту, который отправил запрос.

# VI. ОПИСАНИЕ ИНТЕРФЕЙСА ПРОГРАММЫ

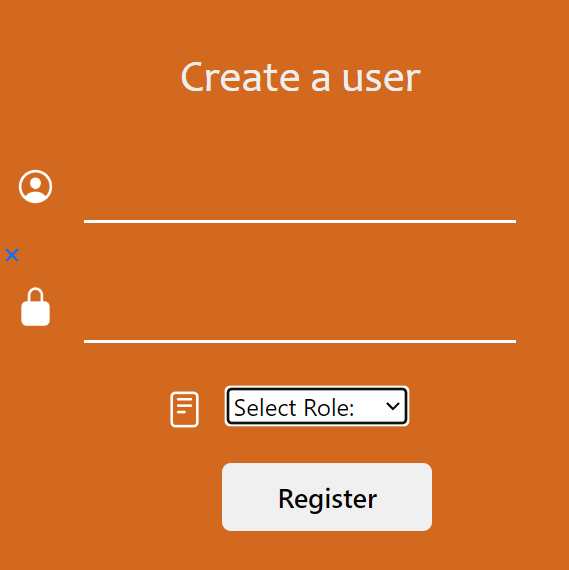
## Окно авторизации клиента

Чтобы использовать программу, необходимо пройти авторизацию путем ввода логина и пароля. Возможность сохранения этих данных для автоматической авторизации при будущих запусках программы также предусмотрена. Если пользователь еще не зарегистрирован в программе, ему доступна форма регистрации. В случае успешной авторизации пользователь перенаправляется на главную форму программы. Визуальный интерфейс окна авторизации и регистрации можно увидеть на представленном ниже рисунке. 

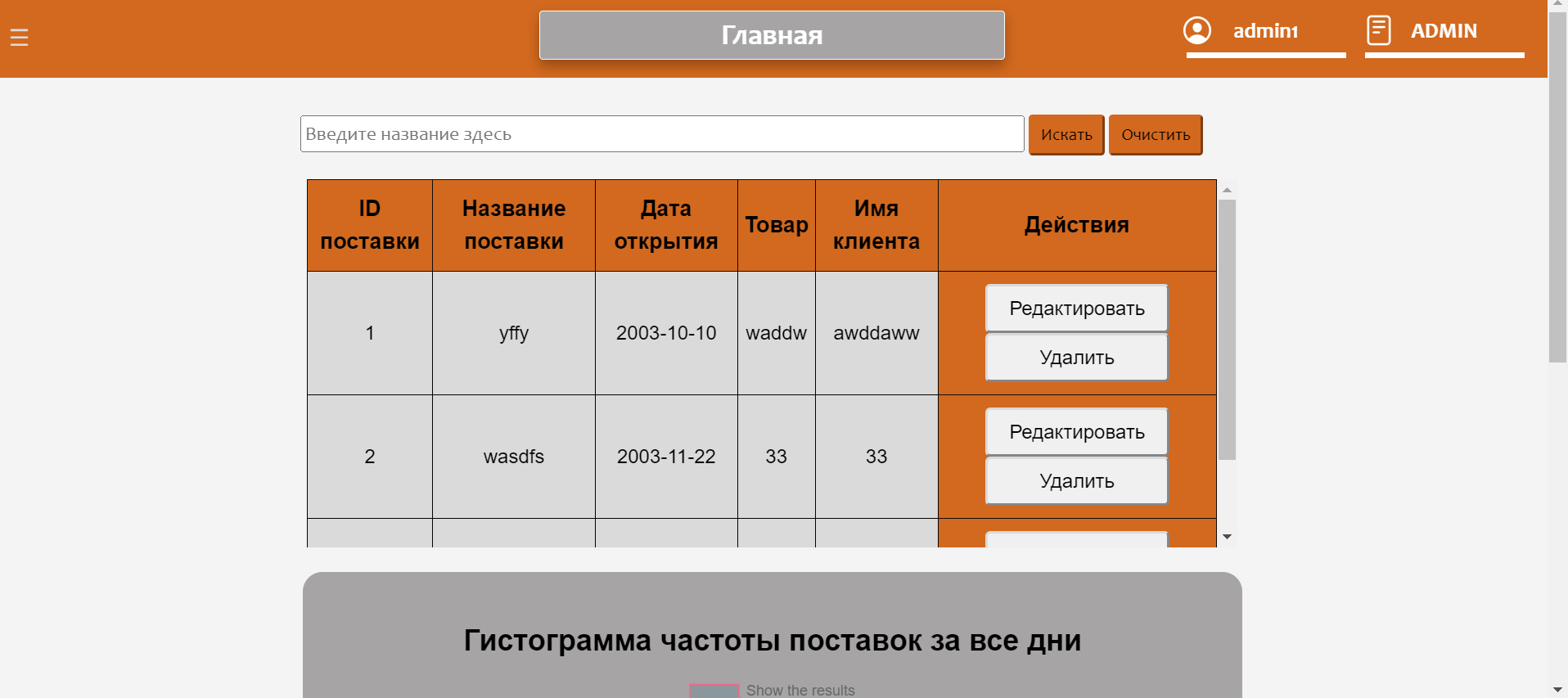
Форма авторизации также включает проверку введенного пароля на корректность. В случае неверного ввода пароля, на экране появляется сообщение об ошибке.

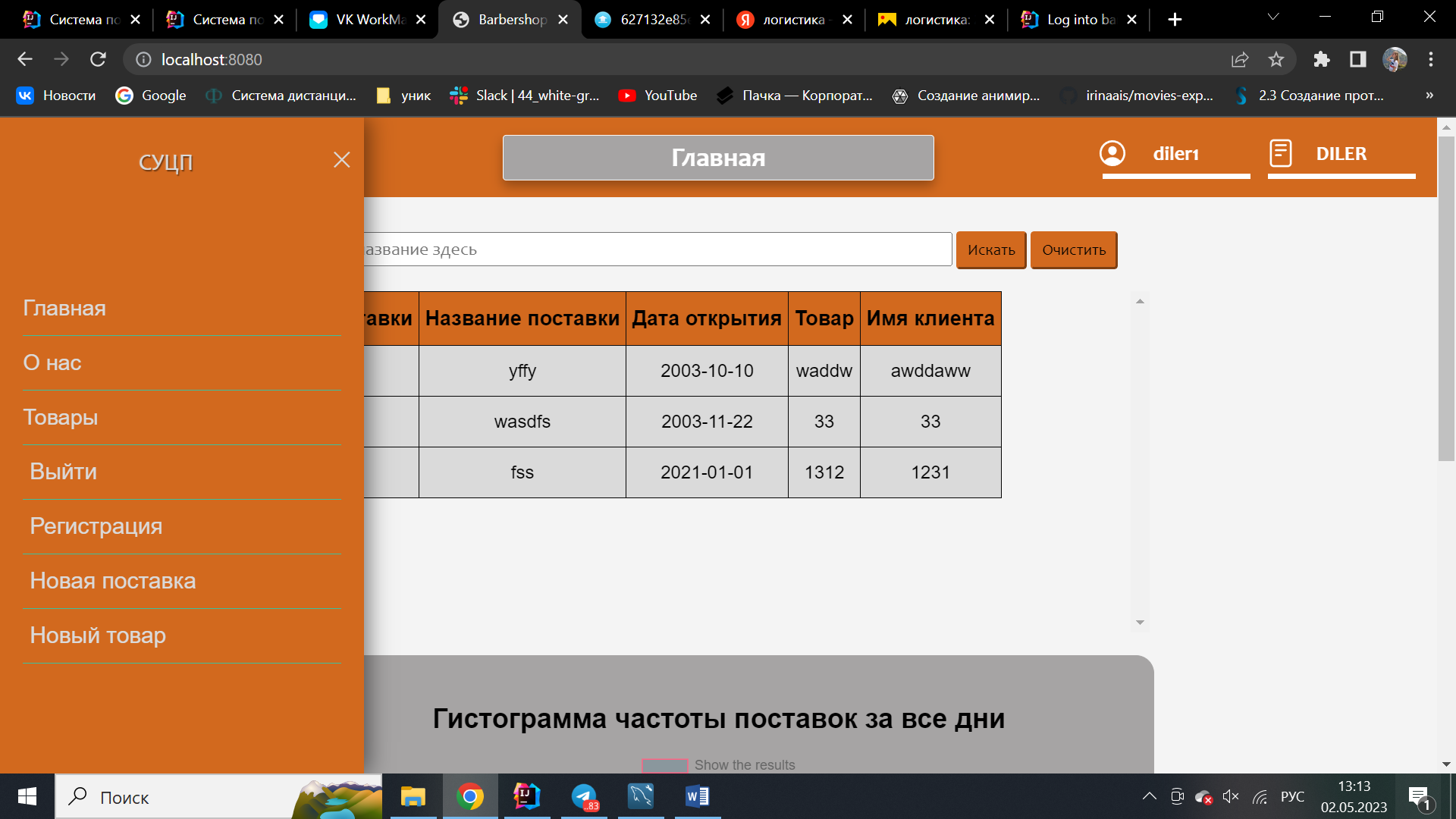


## Окно регистрации нового пользователя

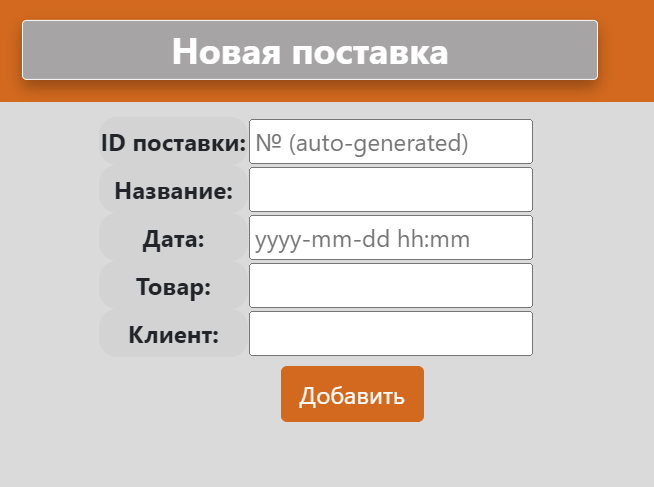
Для создания нового пользователя программа запросит логин и пароль, а также позволит выбрать уровень доступа при создании нового профиля. Уровень доступа может быть дилерским (который имеет право на добавление записей), администраторским (который имеет право на редактирование и удаление записей) или пользовательским (который имеет право только на просмотр записей). Интерфейс окна регистрации может быть найден на представленном ниже рисунке.

## Главное окно программы

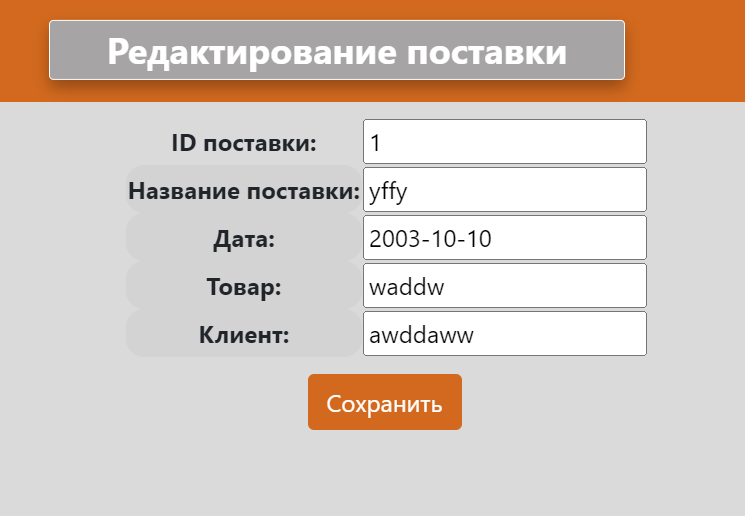
На главной странице сервиса размещена таблица с поставками, где пользователь может выбрать критерий сортировки и использовать форму поиска по ключевым словам, чтобы упростить поиск нужной поставки. Кроме того, на странице есть кнопки для добавления новых товаров и просмотра графиков. 



## Окно добавления новой поставки

После успешной авторизации и перенаправления на главную страницу, пользователь с дилерскими правами может добавлять данные в базе поставок. Для добавления новой поставки пользователю доступно соответствующее окно, которое показано на рисунке. 

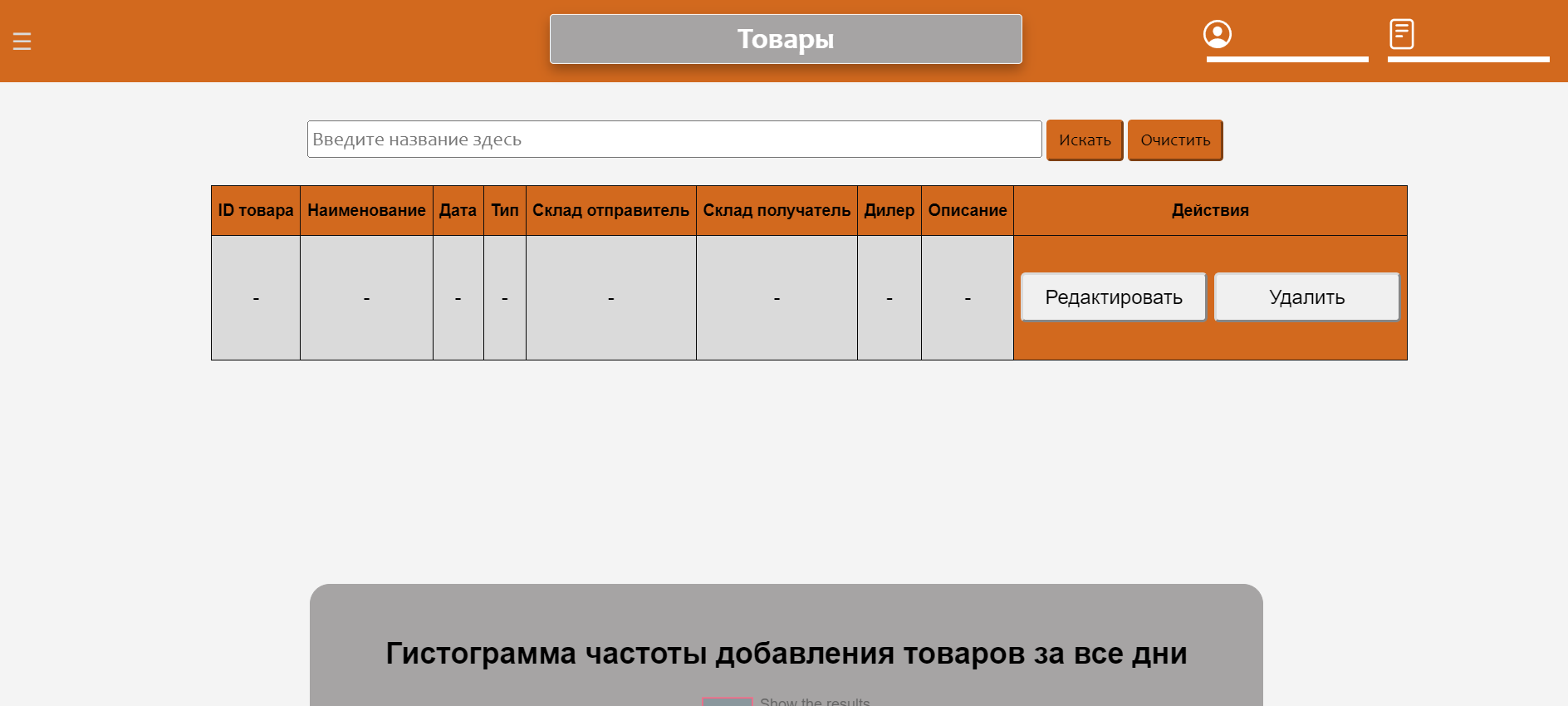
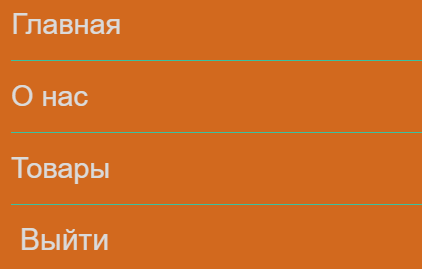
## Окно изменения поставки

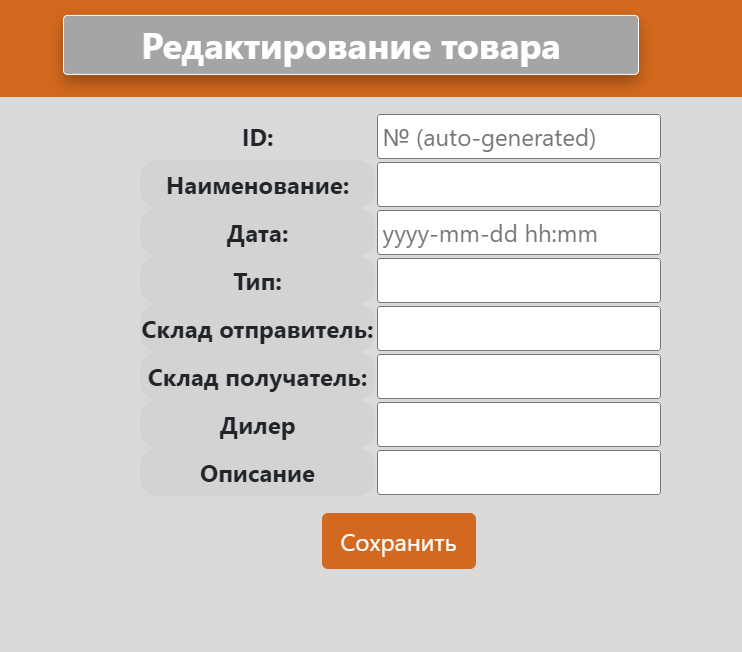


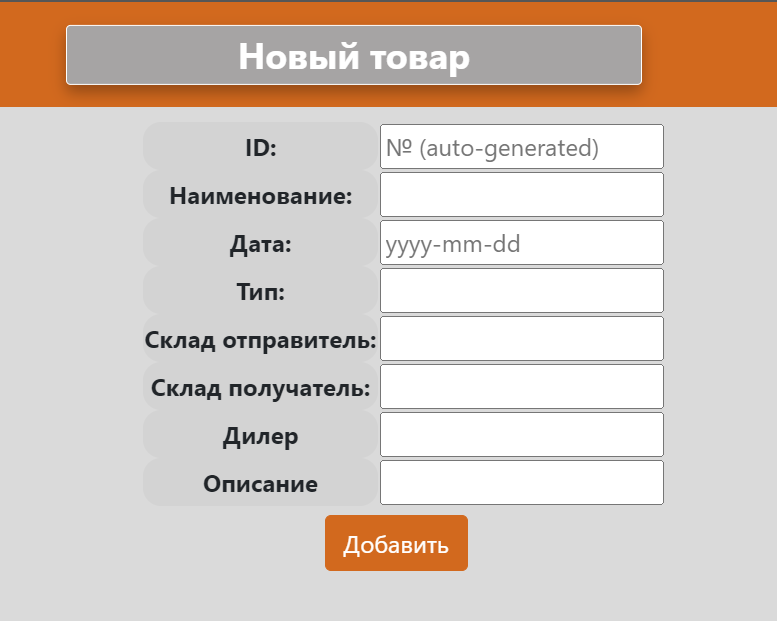
Также есть возможность изменить данные уже имеющейся поставки, для этого предусмотрено отдельное окно, пример которого можно увидеть на другом рисунке.

При нажатии на кнопку «Сохранить» измененные данные сохраняются, окно изменения закрывается и производится возврат на основную страницу сервиса.

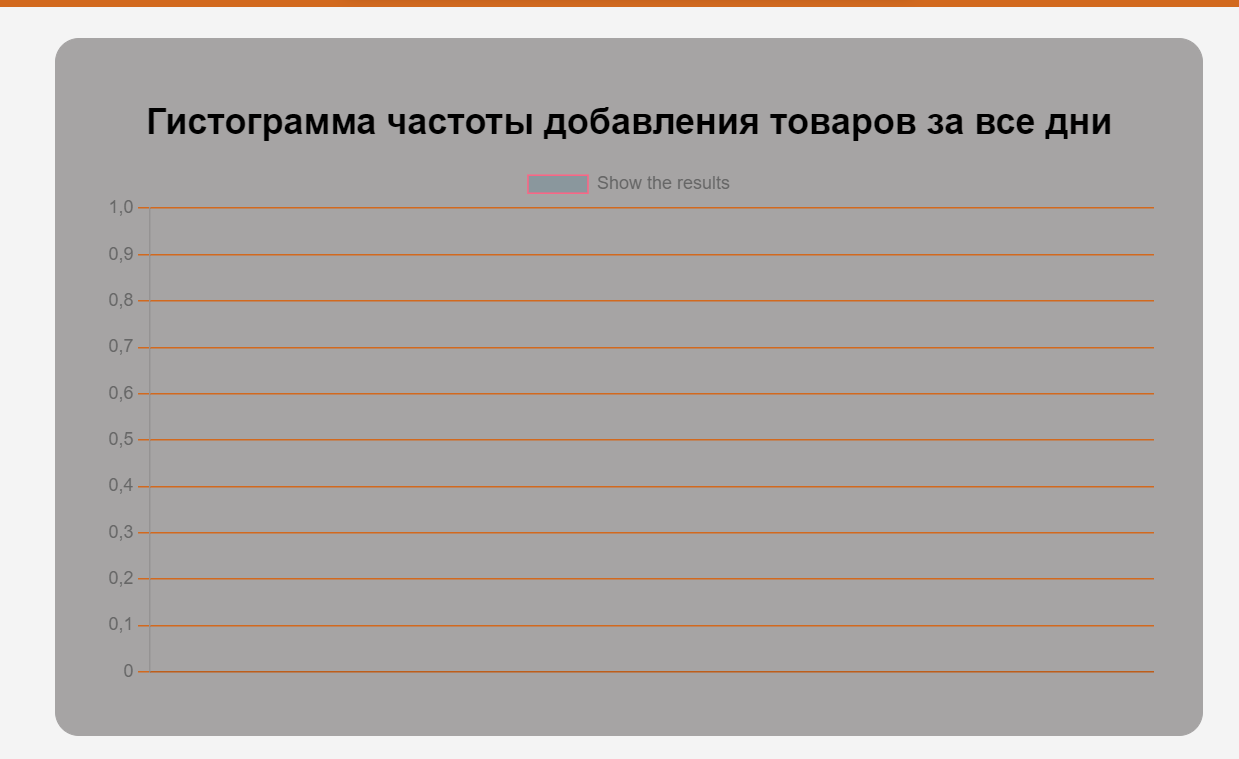
Аналогичные страницы существуют при переходе по кнопке «Текущие операции» в заголовке главной страницы:







Ниже на странице также присутсвует статистический элемент – гистограмма количества заказов по дням, представленная на фото ниже:



# 

# VII. СОСТАВ ПРИЛОЖЕНИЯ

## Сервер

## Сервер включает в себя несколько компонентов:

## Spring Web, предназначенный для создания веб-приложений, в том числе RESTful, с помощью Spring Model View Controller;

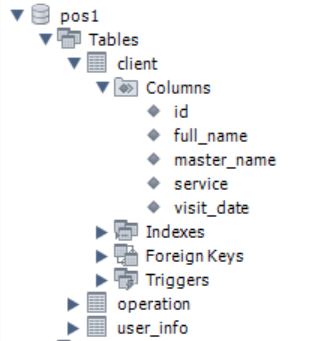
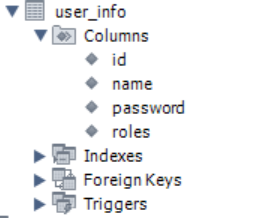
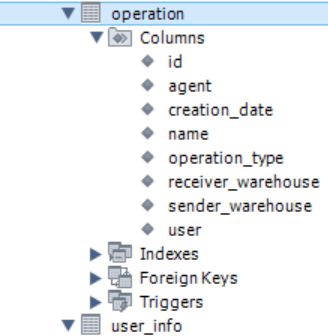
## Spring Boot Gradle plugin, плагин для создания решения.

## MySQL, система управления реляционными базами данных, которая позволяет хранить, организовывать и извлекать данные из базы данных. В данном проекте она используется совместно с JDBC (Java Database Connectivity);

## Gradle, основной инструмент сборки проекта;

## База данных

Для управления базой данных в данном проекте используется реляционная система управления базами данных MySQL, которая содержит несколько таблиц с соответствующими полями.

1. Operation – таблица товаров. Содержит следующие атрибуты:
   * id – уникальный идентификатор товара;
   * agent – дилер, предоставляющий товар;
   * creation\_date – дата регистрирования товара;
   * name – наименование товара;
   * operation\_type – тип товара;
   * sender\_warehouse – склад отправитель;
   * receiver\_warehouse – склад получатель;
   * user – описание товара.
2. UserInfo – таблица данных авторизированных пользователей. Содержит следующие атрибуты:
   * id – уникальный идентификатор поставки;
   * full\_name –наименование поставки;
   * master\_name – клиент, заказавший поставку;
   * service – товар, который должны доставить;
   * visit\_date – дата регистрации поставки.

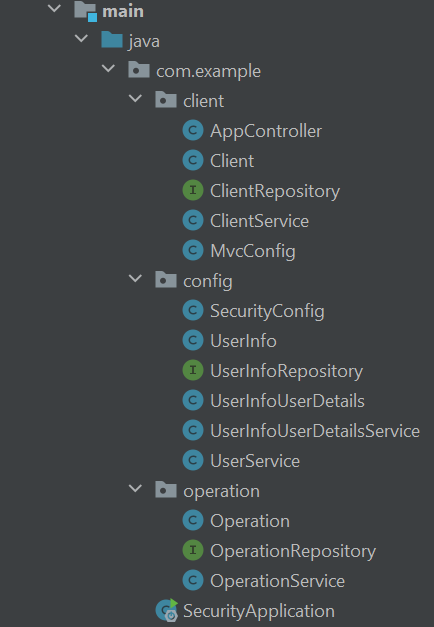
## Клиент

В состав клиентской части сервиса входят следующие внешние библиотеки:

* Jakarta Persistence – отвечает за работу с объектно-реляционным отображением (ORM) в Java;
* Spring Security – предназначена для обеспечения безопасности веб-приложения;
* Gradle – основной сборщик решения (автозагрузка и менеджмент необходимых библиотек, работа с плагинами);

# VIII. НАЗНАЧЕНИЕ И СОСТАВ КЛАССОВ ПРОГРАММЫ

Всего задействовано 12 классов и 9 веб-страниц для корректной работы решения. Среди них можно выделить следующие группы:



* Контроллер (1 шт.) - класс для обработки запросов и связи с классами реализации и веб-страницами;
* Классы конфигураций (2 шт.) - необходимы для переопределения стандартной конфигурации Spring Boot с целью расширения функционала;
* Сущности запросов (4 шт.) - входные данные для дальнейшей обработки на сервере;
* Сервисы (4 шт.) - используются для обработки данных, полученных от контроллеров. Они служат посредниками между контроллерами и репозиториями;
* Репозитории (3 шт.) - представляют коллекции (таблицы) в СУБД.
* Программа запуска (1 шт.) - класс, который используется для запуска сервера.

Рассмотрим все классы подробнее:

* SecurityConfig - класс, отвечающий за конфигурацию Spring Security, настройки доступа к URL, авторизации и аутентификации пользователей, а также за использование BCryptPasswordEncoder для шифрования паролей;
* UserInfo - модель информации о пользователе, которая хранится в базе данных и использует аннотации из библиотеки Jakarta Persistence для определения, что это сущность базы данных;
* UserInfoRepository - интерфейс расширяющий интерфейс JpaRepository из Spring Data JPA, предоставляющий методы для работы с сущностью UserInfo в базе данных;
* UserInfoUserDetails - класс, который создает объект UserDetails для пользователя UserInfo и используется для аутентификации пользователя в Spring Security;
* UserInfoUserDetailsService - сервис, реализующий интерфейс UserDetailsService, необходимый для аутентификации пользователей в системе;
* UserService - сервис для добавления новых пользователей в репозиторий UserInfoRepository с зашифрованным паролем при помощи объекта PasswordEncoder.
* AppController – контроллер для обработки http запросов
* UserController – контроллер для обработки http запросов;
* MvcConfig – конфигурацию Spring MVC (Model-View-Controller), который позволяет настраивать отображения для URL-адресов без необходимости создания контроллеров. В данном случае, метод addViewControllers() добавляет представления контроллеров;
* Operation – представляет сущность Operation, описывает поля таблицы базы данных и предоставляет методы для доступа к этим полям;
* SecurityApplication – точка входа в приложение, используется для запуска Spring Boot приложения и создания контекста приложения.;
* OperationRepository – отвечает за методы для доступа к хранилищу данных и операции чтения и записи сущности Operation;
* OperationService – отвечает за обработку запросов, связанных с сущностью Operation, используя репозиторий OperationRepository.
* index.html
* about\_us.html
* blog.html
* edit\_client.html
* new\_client.html
* login\_page.html
* register.html
* new\_post.html
* edit\_post.html

# IX. ЗАКЛЮЧЕНИЕ

Данная курсовая работа была посвящена разработке веб-сервиса, который был успешно реализован с использованием HTML, реляционной базы данных MySQL и библиотеки Spring Boot. В ходе работы были выполнены все поставленные задачи, и созданный сервис имеет интуитивно понятный пользовательский интерфейс, что делает его удобным для использования для любых пользователей.

Более того, разработанный сервис имеет большой потенциал для дальнейшей модернизации и расширения. Возможна доработка пользовательского интерфейса, расширение базы данных с добавлением новых таблиц и усложнение функционала. Также можно добавить новые роли для иерархии прав доступа к базе данных и ее изменениям.

В процессе выполнения курсовой работы был получен значительный опыт работы с базами данных, пользовательскими интерфейсами и разметкой веб-страниц. Этот опыт может быть полезен для дальнейшей разработки веб-сервисов и приложений. Кроме того, при написании данной работы были изучены принципы и практики разработки веб-сервисов с использованием библиотеки Spring Boot и реляционных баз данных, а также применены знания в области HTML и CSS для создания пользовательского интерфейса.

В результате созданного веб-сервиса была достигнута цель проекта - создание функционального приложения, позволяющего пользователю хранить, добавлять и изменять данные в базе данных.

Полученный опыт и знания будут полезны для дальнейшей работы с веб-технологиями, а также в различных проектах, связанных с базами данных и разработкой веб-сервисов.

# X. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Учебная и научная литература

1. Володченкова Л.А., Козырев Д.В. Разработка серверной части программного приложения для удаленного хранения данных // МСиМ. 2020. №1 (53).
2. Байдыбеков А.А., Гильванов Р.Г., Молодкин И.А. СОВРЕМЕННЫЕ ФРЕЙМВОРКИ ДЛЯ РАЗРАБОТКИ WEB-ПРИЛОЖЕНИЙ // Интеллектуальные технологии на транспорте. 2020. №4 (24).
3. Гасанов Заурбек Зубаирович Анализ производительности многопоточных программ, написанных на языках Java и Go // Наука и образование сегодня. 2018. №6 (29).
4. Барабанов В.Ф., Донских А.К., Гребенникова Н.И., Кенин С.Л. ПОЛУЧЕНИЕ МЕТРИК JAVA-ПРИЛОЖЕНИЯ В КОНТЕЙНЕРАХ DOCKER // Вестник ВГТУ. 2020. №2.
5. О.А. Ляшенко, О.О. Конашков, Н.А. Солодкая СРАВНИТЕЛЬНЫЙ АНАЛИЗ ВЫПОЛНЕНИЯ ЗАПРОСОВ К СЕРВЕРАМ БАЗ ДАННЫХ MYSQL И MONGODB // Вестник Херсонского национального технического университета. 2019. №4 (71).

# XI. ПРИЛОЖЕНИЯ

Приложение A. Исходный код программы