

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук
Кафедра технологий обработки и защиты информации

Курсовой проект

Система для учета и управления складскими запасами малых и средних
предприятий с аналитикой продаж

Направление 09.03.02 «Информационные системы и технологии»
Обработка информации и машинное обучение

Зав. Кафедрой _____ проф., д.т.н. Сирота А.А.
подпись

Руководитель _____ ст. преп. В.С. Тарасов
подпись

Обучающийся _____ ст. 3 курса оч. отд. К.М. Зинченко
подпись

Обучающийся _____ ст. 3 курса оч. отд. Е.А. Дмитрива
подпись

Обучающийся _____ ст. 3 курса оч. отд. Н.С. Ульянов
подпись

Воронеж 2024

Содержание

Введение	5
1 Постановка задачи.....	6
1.1 Цели создания системы.....	6
1.2 Задачи проекта.....	6
1.3 Функциональные и нефункциональные требования	6
1.3.1 Функциональные требования	6
1.3.2 Нефункциональные требования	7
2 Анализ предметной области	8
2.1 Терминология	8
2.2 Обзор аналогов	9
2.2.1 1С:ERP	10
2.2.2 ЕКАМ.....	10
2.2.3 МойСклад	11
2.3 Моделирование системы.....	12
2.3.1 Диаграмма прецедентов	12
2.3.2 Диаграммы последовательности	13
2.3.3 Диаграмма развертывания	18
2.3.4 Диаграммы состояния	18
2.3.5 Диаграмма объектов.....	20
2.3.6 Диаграмма активности	21
3 Реализация	23
3.1.1 Средства реализации	23
3.1.2 Реализация базы данных	23
3.1.3 Реализация клиентской части	24

3.1.4 Форма для экрана «Приветствие»	24
3.1.5 Форма для экрана «Авторизации».....	25
3.1.6 Форма для экрана «Главная страница»	26
3.1.7 Форма для экрана «Список заказа»	26
3.1.8 Форма для экрана «Профиль»	27
3.1.9 Форма для экрана «Регистрация».....	27
3.1.10 Форма для экрана «Установка пароля».....	28
3.1.11 Форма для экрана «Работники».....	29
3.1.12 Форма для экрана «Редактировать»	29
3.1.13 Форма для экрана «Товар».....	30
3.1.14 Форма для экрана «Список продуктов».....	31
3.1.15 Форма для экрана «Принять товар»	32
3.1.16 Форма для «Составить отчет»	32
3.1.17 Форма для экрана «Остатки товара»	33
3.2 Реализация серверной части	33
3.2.1 Архитектура серверной части приложения	33
3.2.2 Диаграмма классов контролеров	33
3.2.3 Диаграмма классов сервисов	34
3.2.4 Диаграмма классов репозиториев	35
3.2.5 Диаграмма классов моделей	35
3.2.6 Диаграмма классов DTO	35
3.2.7 Диаграмма классов исключений	35
3.2.8 Диаграмма классов конфигураций	36
3.2.9 Диаграмма классов сериализаторов	36

4 Аналитика	37
Заключение	39
Список литературы	40

Введение

Систематизация учета складских запасов предприятия – необходимая часть работы менеджеров малых и средних предприятий, которые хотят развивать свой бизнес и вести учет своих доходов и расходов. Эта необходимость вызвана потребностью избегания переизбытка или дефицита товаров и продукции, которые приводят к потерям прибыли и потере клиентов. Более того, анализ продаж является неотъемлемой частью любого бизнеса, так как он позволяет оценить эффективность продаж, выявить тенденции и спрогнозировать будущие результаты.

В настоящее время появляется множество разнообразных малых и средних предприятий, за 15 лет их выручка выросла более чем в три раза, занятость в этом секторе увеличилась в 1,6 раз и составила 31 млн человек. при этом количество предприятий, имеющих потребность в управлении складскими запасами так же, растет. Разработчики программного обеспечения реагируют на потребности рынка и создают приложения для их удовлетворения. Этим предприятиям необходимо простое, понятное, не требующее долгой установки и внедрения приложение, оптимизирующее их деятельность.

В данной курсовой работе будет рассмотрен процесс разработки веб-приложения для учета складских запасов с аналитикой продаж. Для реализации приложения будут использованы следующие технологии: Vue.js, Java, PostgreSQL.

Целью текущей курсовой работы является практическое применение знаний и умений по разработке web-приложения для создания полноценного коммерческой системы для учета и управления складскими запасами малых и средних предприятий с аналитикой продаж. Работа будет содержать детальное описание всех этапов разработки продукта, включая анализ требований, проектирование, реализацию, тестирование.

1 Постановка задачи

1.1 Цели создания системы

Задачей курсовой работы является разработка веб-приложения для учета и управления складскими запасами малых и средних предприятий с аналитикой продаж.

1.2 Задачи проекта

- авторизация;
- редактирование пользователей;
- редактирование товаров;
- анализ движения товаров внутри предприятия;
- оформление заказа со склада.

1.3 Функциональные и нефункциональные требования

1.3.1 Функциональные требования

Функциональные требования:

1. В системе должна быть возможность авторизации.
2. Система должна предоставлять возможность просмотра остатков товара на складе.
3. Система должна предоставлять возможность создать заказ.
4. Система должна предоставлять возможность просмотр предыдущих заказов.
5. Система должна предоставлять возможность работнику принять задания для их выполнения.
6. Система должна предоставлять возможность сообщения о проблеме.
7. Система должна предоставлять возможность принять поступления на склад.
8. Система должна предоставлять возможность составить отчет по выбранным категориям.

9. Система должна предоставлять возможность редактировать товары.

10. Система должна предоставлять возможность редактировать учетные записи.

1.3.2 Нефункциональные требования

Данное приложение должно удовлетворять следующим основным требованиям:

- приложение должно корректно работать в браузерах Google (версия 90.0.4430.93 или новее), Firefox (версия 90.0.2 или новее), Edge (90.0.818.39 или новее);
- приложение должно работать при одновременном использовании 1000-ю пользователями;
- время ожидания ответа после действия пользователя не должно превышать 8 секунд;
- реализовывать все поставленные задачи.

Требования по безопасности:

- только начальник склада и администратор должны иметь доступ на просмотр и изменение информации о товарах на складе;
- использование механизмов защиты от SQL-инъекций: Применение ORM (Object-Relational Mapping) фреймворков, которые автоматически обрабатывают ввод пользователя и предотвращают SQL-инъекции;
- использование протокола передачи данных HTTPS: Переход на использование HTTPS вместо HTTP для обеспечения шифрования данных в процессе передачи между клиентом и сервером;
- хранение паролей при помощи Bcrypt: Использование алгоритма Bcrypt для хеширования паролей перед их сохранением в базе данных.

2 Анализ предметной области

2.1 Терминология

Таблица 1 - Глоссарий

Веб-приложение	Клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера.
Клиент	Это аппаратный или программный компонент вычислительной системы, посылающий запросы серверу.
Сервер	Выделенный или специализированный компьютер для выполнения сервисного программного обеспечения.
База данных	Это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных.
HTTP	Протокол, позволяющий получать различные ресурсы, например, HTML-документы. Протокол HTTP лежит в основе обмена данными в сети Интернет.
SQL-запросы	Наборы команд для работы с реляционными базами данных.
SQL-инъекция	Внедрении в запрос произвольного SQL-кода, который может повредить или уничтожить данные, хранящиеся в БД или предоставить несанкционированный доступ к ним.
Дизайн-макет	Схематичное изображение финальной идеи с указанием всех деталей. В нем указываются концепция, шрифты, тексты, изображения, расположение всех элементов и общая картина продукта.

Аутентификация	Процедура проверки подлинности, например, проверка подлинности пользователя путем сравнения введенного им пароля с паролем, сохраненным в базе данных.
Авторизация	Предоставление определенному лицу или группе лиц прав на выполнение определенных действий.
Учетная запись	Запись, которая содержит сведения, необходимые для идентификации пользователя при подключении к системе, а также информацию для авторизации и учёта.
Фреймворк	Программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.
Аккаунт	Персональная страница пользователя или личный кабинет, который создается после регистрации на сайте или в приложении.
REST	Архитектурный стиль взаимодействия компонентов распределённого приложения в сети.
API	Описание взаимодействия одной компьютерной программы с другой
Пользователь	Человек, использующий приложение

2.2 Обзор аналогов

Разрабатывая приложение, основной задачей которого является учет и управления складскими запасами малых и средних предприятий с аналитикой продаж, необходимо рассматривать разработку с точки зрения актуальности и уникальности проекта. Для оценки этих качеств необходимо прибегнуть к рассмотрению аналогов разрабатываемого приложения, адекватно оценивая все положительные и негативные черты того или иного продукта.

2.2.1 1C:ERP

Детализированная база: Магазины "Бытовая техника" / Борисов Федор Михайлович / (НС) Приемка

Поступления то... По: ТДРТ-000002 Заказы постав... За: ТДРТ-000118 Приемка товаров Пр: РТ-00000006 Определения... Оп: ЧРТ-000002 Сотрудники

Приемка товаров

Магазин: Склад:

Принадлежность ордера: Распоряжения на приемку

Дата	№	Магазин	Склад	Описание	Сумма докум...	Ответственный	Комментарий
29.01.2015	РТ-00000079	Магазин "Детские ...	Отдел магазина "	Перемещение товаров ТД...		Борисов Федор М...	
29.01.2015	РТ-00000080	Магазин "Детские ...	Отдел магазина "	Перемещение товаров ТД...		Борисов Федор М...	
29.01.2015	РТ-00000081	Магазин "Детские ...	Отдел магазина "	Перемещение товаров ТД...		Борисов Федор М...	
29.01.2015	РТ-00000082	Магазин "Детские ...	Отдел магазина "	Перемещение товаров ТД...		Борисов Федор М...	
29.01.2015	РТ-00000083	Магазин "Детские ...	Отдел магазина "	Перемещение товаров ТД...		Борисов Федор М...	
29.01.2015	РТ-00000084	Магазин "Детские ...	Отдел магазина "	Перемещение товаров ТД...		Борисов Федор М...	
30.01.2015	РТ-00000085	Магазин "Детские ...	Отдел магазина "	Перемещение товаров ТД...		Борисов Федор М...	
30.01.2015	РТ-00000086	Магазин "Детские ...	Отдел магазина "	Перемещение товаров ТД...		Борисов Федор М...	
31.01.2015	РТ-00000087	Магазин "Детские ...	Отдел магазина "	Перемещение товаров ТД...		Борисов Федор М...	
31.01.2015	РТ-00000088	Магазин "Детские ...	Отдел магазина "	Перемещение товаров ТД...		Борисов Федор М...	
02.02.2015	РТ-00000089	Магазин "Бытовая...	Общий склад	Поступление товаров ТДР...	2 745 030,00	Борисов Федор М...	
02.02.2015	РТ-00000090	Магазин "Бытовая...	Отдел "Техника д...	Поступление товаров ТДР...	316 809,00	Борисов Федор М...	
19.02.2015	РТ-00000091	Магазин "Бытовая...	Отдел "Холодильн...	Заказ поставщику ТДРТ-0...	329 656,00	Борисов Федор М...	
19.02.2015	РТ-00000092	Магазин "Бытовая...	Отдел "Бытовая т...	Заказ поставщику ТДРТ-0...	87 320,00	Борисов Федор М...	
19.02.2015	РТ-00000093	Магазин "Бытовая...	Общий склад	Заказ поставщику ТДРТ-0...	42 516,00	Борисов Федор М...	
19.02.2015	РТ-00000094	Магазин "Бытовая...	Отдел "Техника д...	Заказ поставщику ТДРТ-0...	310 457,00	Борисов Федор М...	
19.02.2015	РТ-00000095	Магазин "Бытовая...	Отдел "Телевизио...	Заказ поставщику ТДРТ-0...	56 170,00	Борисов Федор М...	

Рисунок 1 – Интерфейс программы 1C:ERP

Недостатки:

- сложный интерфейс, обучение займет время;
- настройка и обслуживание требует специалиста с высоким уровнем квалификации;
- просмотр аналитических учетов.

Преимущества:

- создание документации;
- настройка рабочего места кассира;
- высокая цена за решение «из коробки».

2.2.2 ЕКАМ

Артикул	Наименование	Остаток	Резерв	Ожидание	Доступно	Ед. изм.	Секст.	Сум. секст.	Цена продажи
6	Баклажан 1кг	-11661.648	0.0	0.0	-11661.648	кг	0,00 руб	0,00 руб	299,00 руб
	Бонаква	-2	0	0	-2	шт	0,00 руб	0,00 руб	55,00 руб
5	Виноград белый 1кг	-148.07	0.0	0.0	-148.07	кг	0,00 руб	0,00 руб	229,00 руб
23	Вода Жемчужина Байкала природная артезианская, негазированная 1,25л	79	0	1	80	шт	40,00 руб	3 160,00 руб	89,90 руб
28.1	Гранулы Расти Большой Чудесника (Вкус: Клубника / Длина: 100)	-55	0	0	-55	шт	0,00 руб	0,00 руб	59,90 руб
28.3	Гранулы Расти Большой Чудесника (Длина: 120 / Вкус: Банан)	2	0	0	2	шт	20,00 руб	40,00 руб	59,90 руб
28.2	Гранулы Расти Большой Чудесника (Длина: 150 / Вкус: Шоколад)	-1	0	0	-1	шт	0,00 руб	0,00 руб	59,90 руб
14.1	Кондиционер Lenor 1л (Вид: Альпийские Луга)	4	0	0	4	шт	60,00 руб	240,00 руб	175,00 руб
14.2	Кондиционер Lenor 1л (Вид: Детский)	-49	0	0	-49	шт	0,00 руб	0,00	
	Кондиционер Lenor 1л (Вид: ...)								

Рисунок 2– Интерфейс программы ЕКАМ

Недостатки:

— необходима долгая настройка под конкретное предприятие.

Преимущества:

- создание документации;
- настройка онлайн-касс;
- кассовые смены.

2.2.3 МойСклад

Наименование	Остаток	Неснижаемый остат...	Резерв	Ожидание	Доступно	Ед. изм.	Оптовая
Диск	150		110	0	40		100,00
Колесо	149		3	0	146 шт		3 000,00
Шина	-1		0	0	-1		200,00
Ботинки/Кожаные							
Ботинки зимние	-1	0	0	0	-1		5 000,00
Ботинки зимние (40)	13	0	6	0	7		5 000,00
Ботинки зимние (41)	9	0	2	5	12		5 000,00
Ботинки зимние (42)	9	0	2	5	12		5 000,00
Сапоги женские	-1		0	0	-1		6 000,00

Рисунок 3– Интерфейс программы МойСклад

Недостатки:

- необходима интеграция с кассой для корректной работы.

Преимущества:

- генерация штрихкодов;
- поддерживаются дисконтные карты, создание клиентской базы;
- отсутствие синхронизации с внешними устройствами;
- удобный интерфейс, позволяющий легко вводить сотрудников в должность.

2.3 Моделирование системы

2.3.1 Диаграмма прецедентов

Рассмотрим полную диаграмму для использования приложения разными типами пользователей. В данном случае необходимость составления диаграммы прецедентов продиктована прежде всего тем, что use-case диаграмма — это инструмент для моделирования системы и понимания ее функциональности и потребностей пользователей. Они помогают в определении основных действий, которые пользователь должен совершить в системе, чтобы достичь определенных целей. Они также позволяют определить возможные риски и проблемы, которые могут возникнуть в ходе использования системы. Данная диаграмма представлена на рисунке 4.

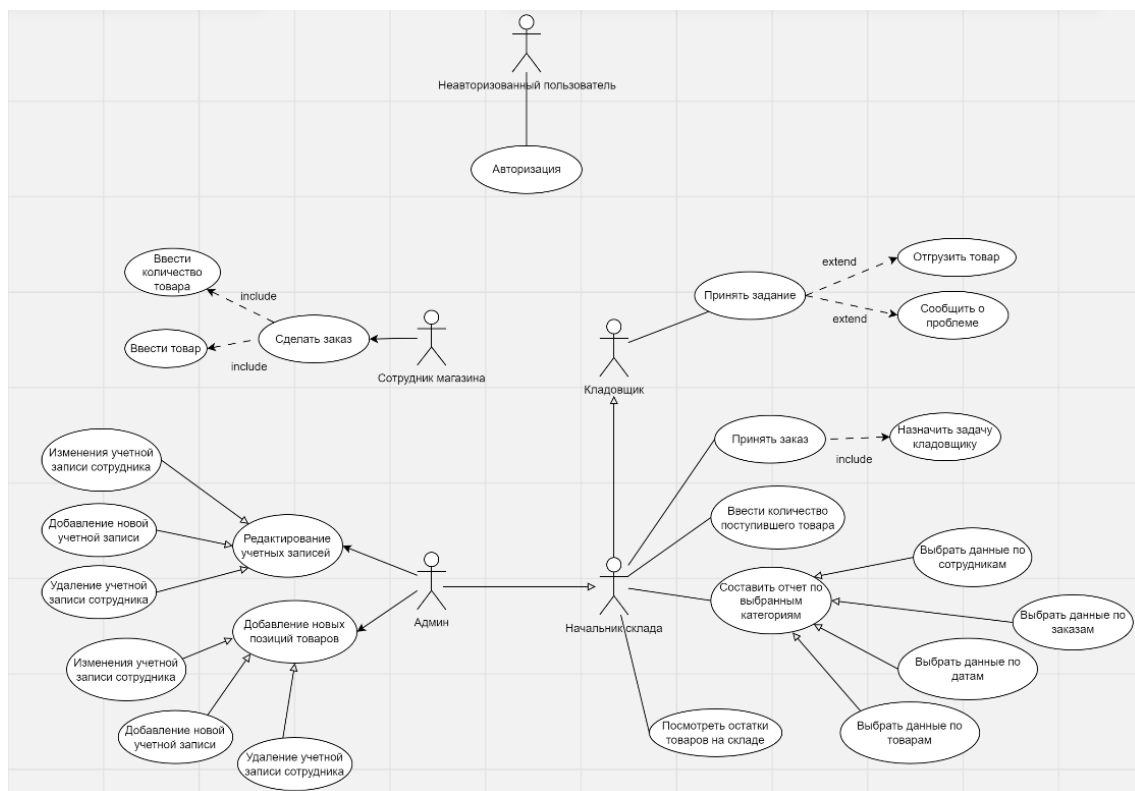


Рисунок 4— Use-Case диаграмма пользования приложением

2.3.2 Диаграммы последовательности

Диаграмма последовательности является важным инструментом для проекта, который помогает более глубоко понимать процесс, улучшать его эффективность и упрощать взаимодействие.

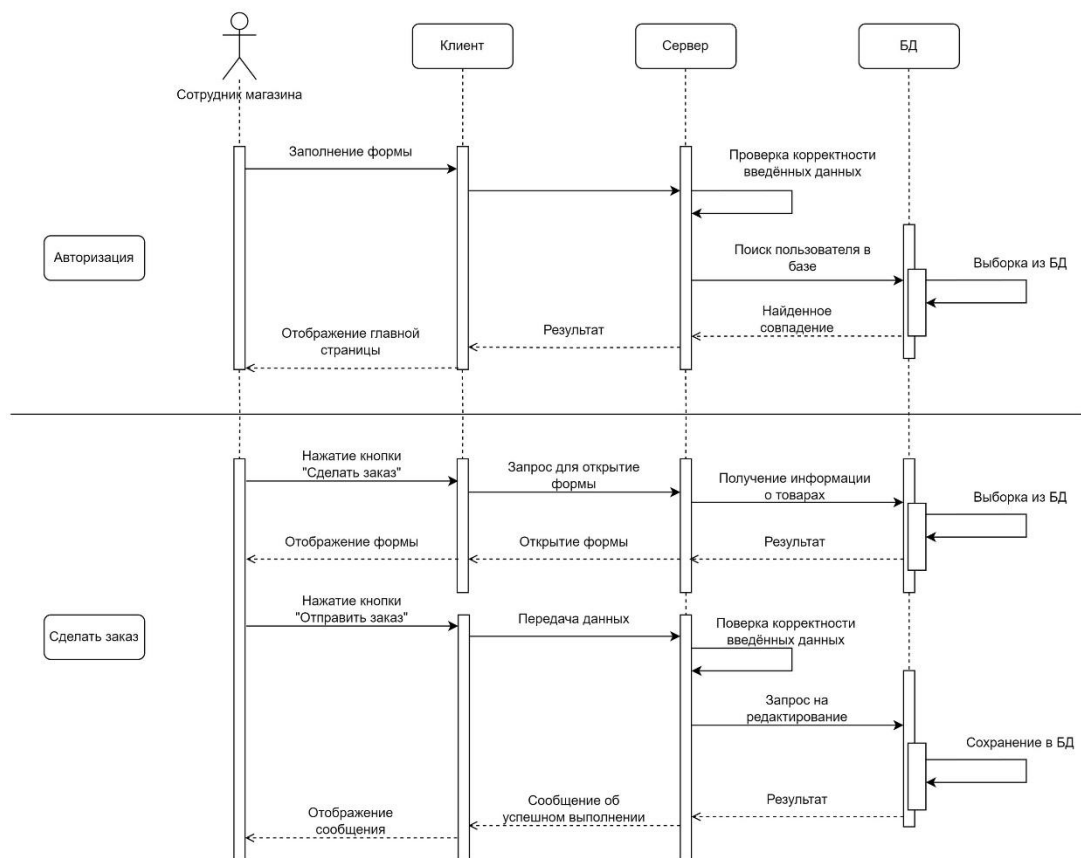


Рисунок 5— Диаграмма последовательности сотрудника магазина

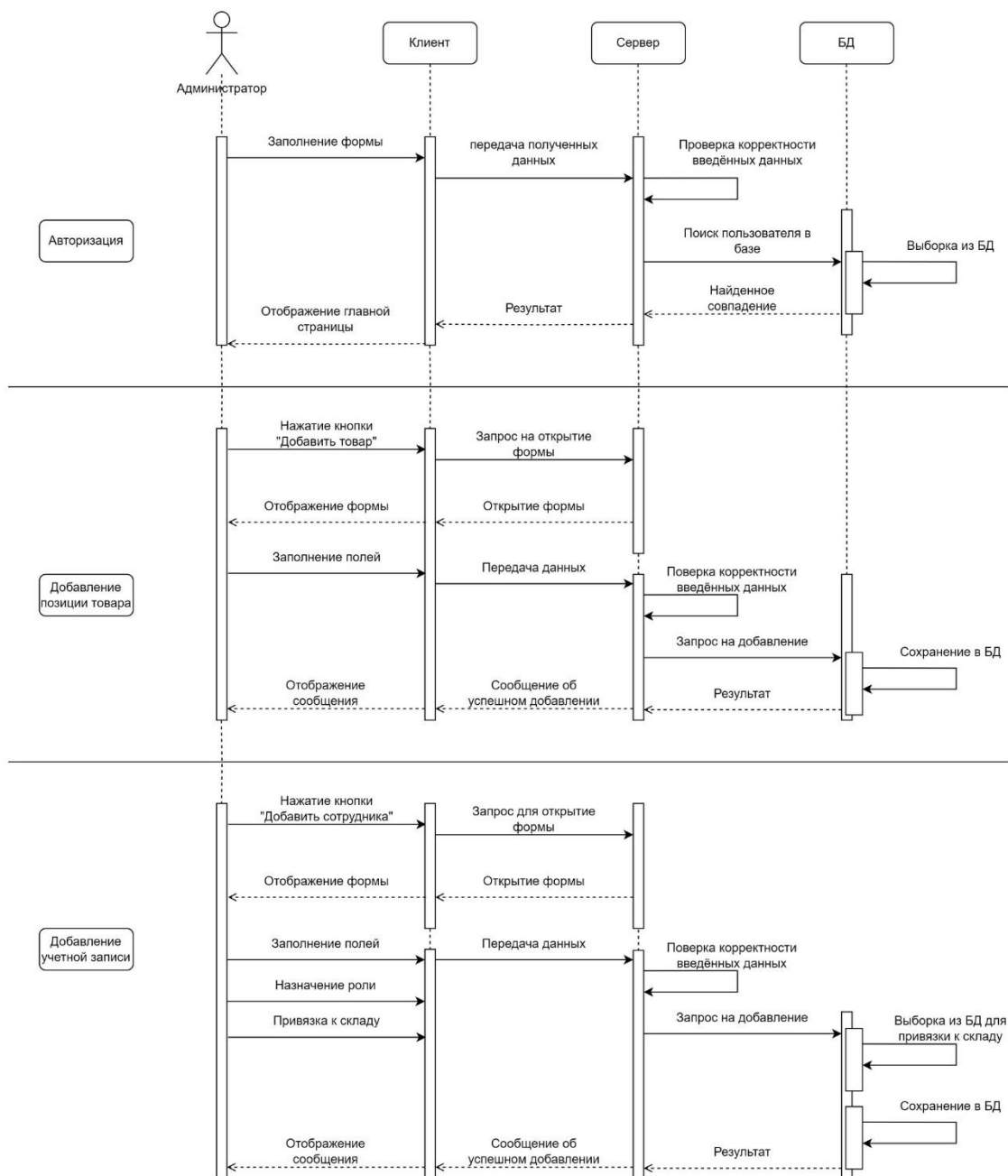


Рисунок 6 – Диаграмма последовательности администратора

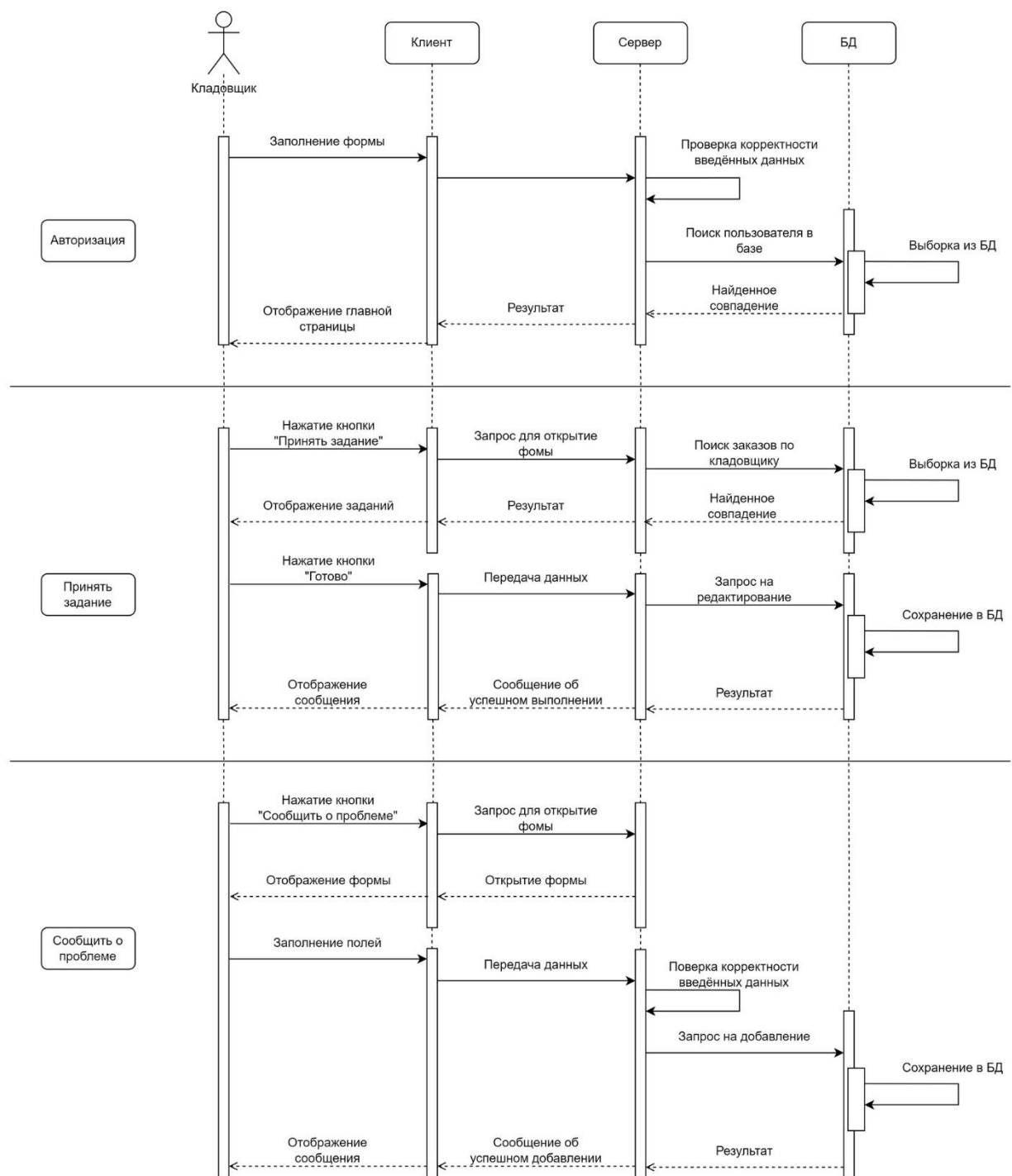


Рисунок 7 – Диаграмма последовательности кладовщика

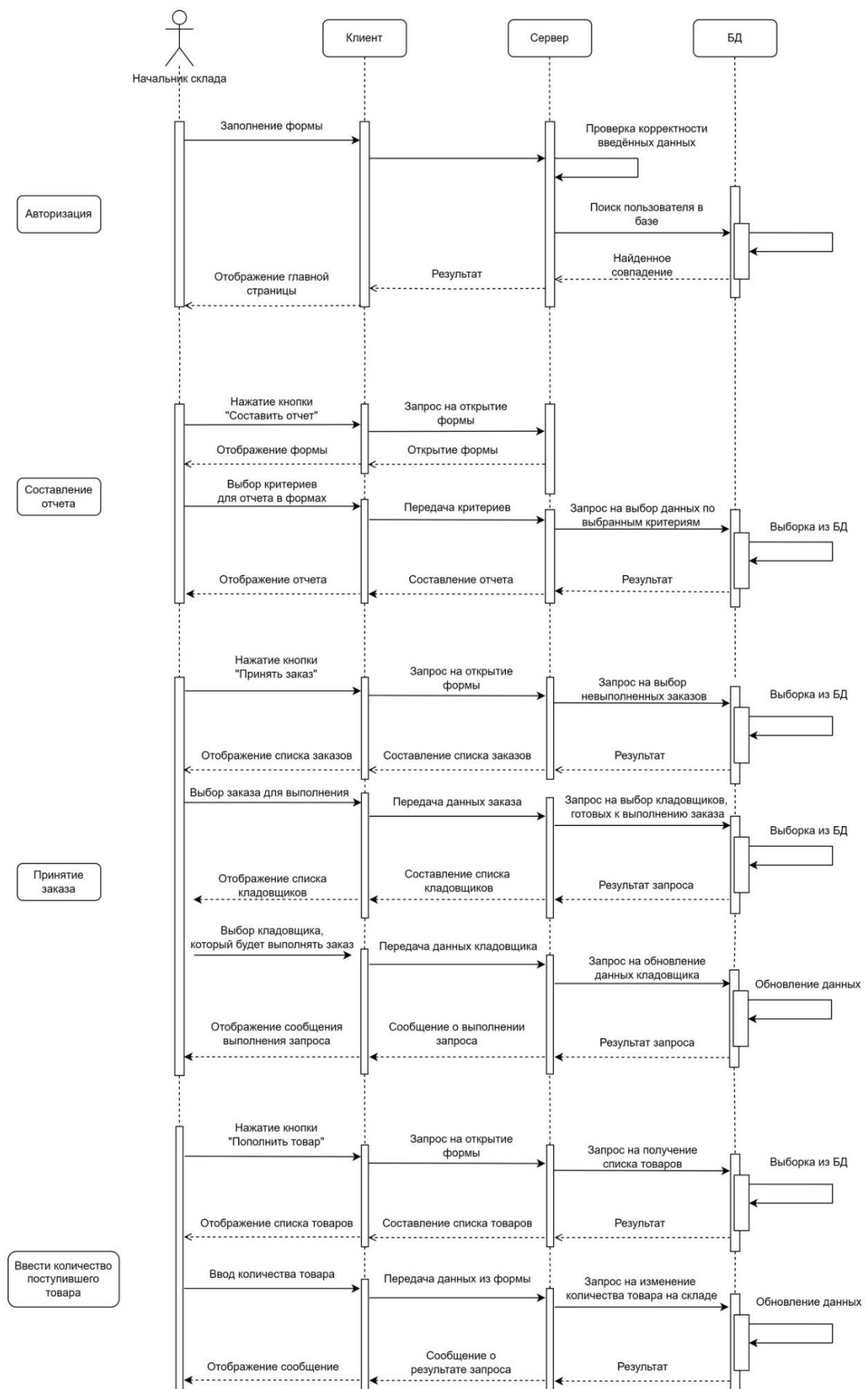


Рисунок 8 – Диаграмма последовательности начальника склада

2.3.3 Диаграмма развертывания

Диаграмма развертывания позволяет определить требования к аппаратному обеспечению, планировать установку и настройку компонентов системы, а также оценивать ее производительность и масштабируемость. Данная диаграмма представлена на рисунке 9.

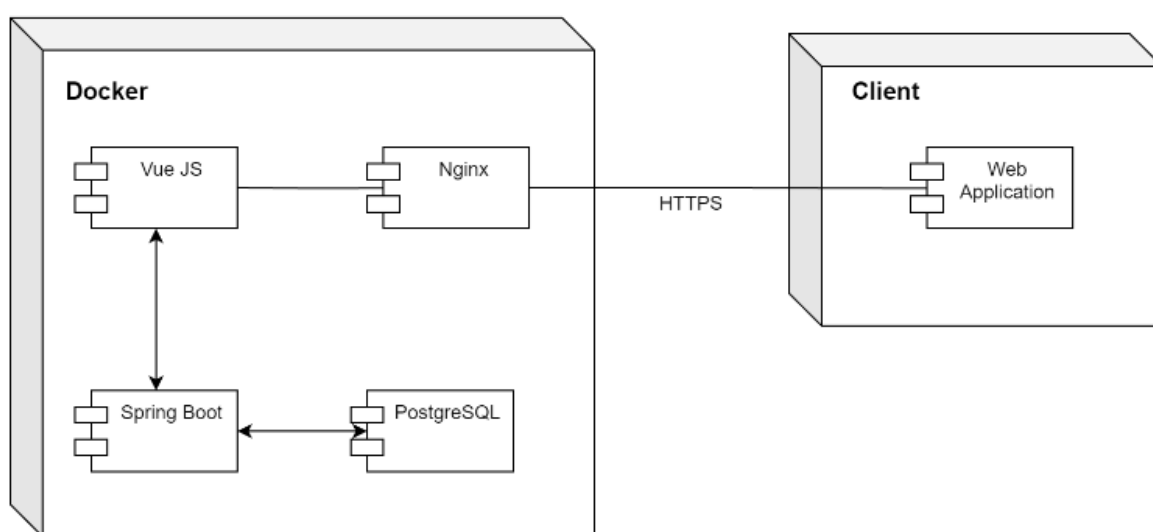


Рисунок 9– Диаграмма развертывания приложения

2.3.4 Диаграммы состояния

Диаграмма состояния позволяет определить возможные сценарии поведения системы, выделить ключевые состояния и переходы между ними, а также оценить ее надежность и устойчивость к ошибкам. Для нашего проекта были спроектированы 3 диаграммы для состояний премиум пользователь, авторизованного пользователя и неавторизованного пользователя. Данные диаграммы представлены на рисунках 10-12.

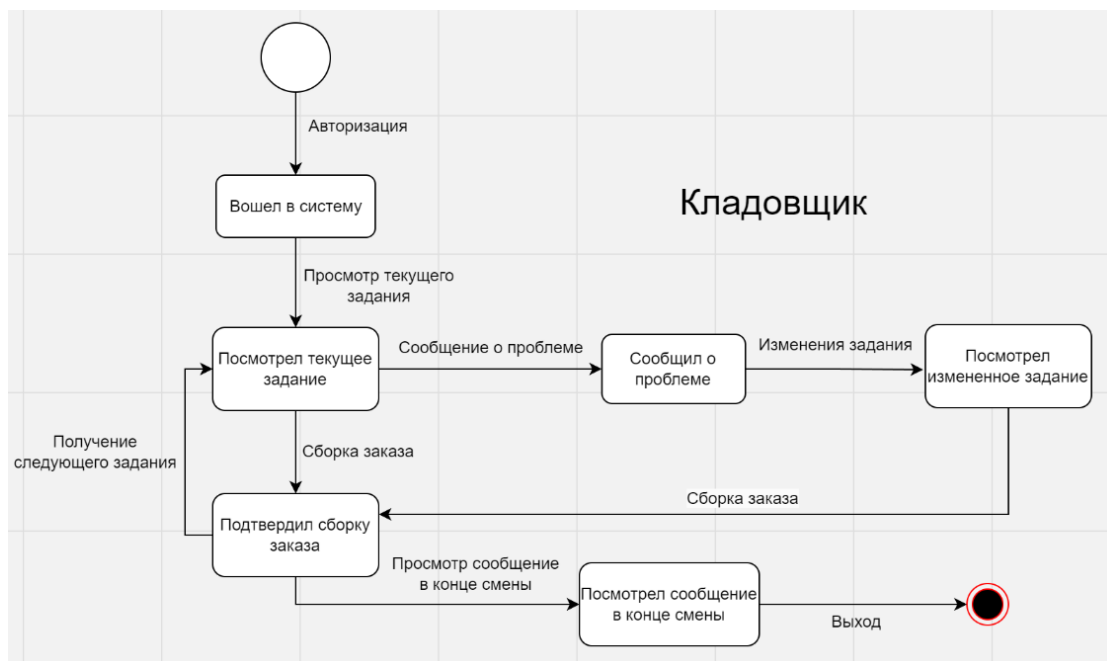


Рисунок 10– Диаграмма состояния кладовщика

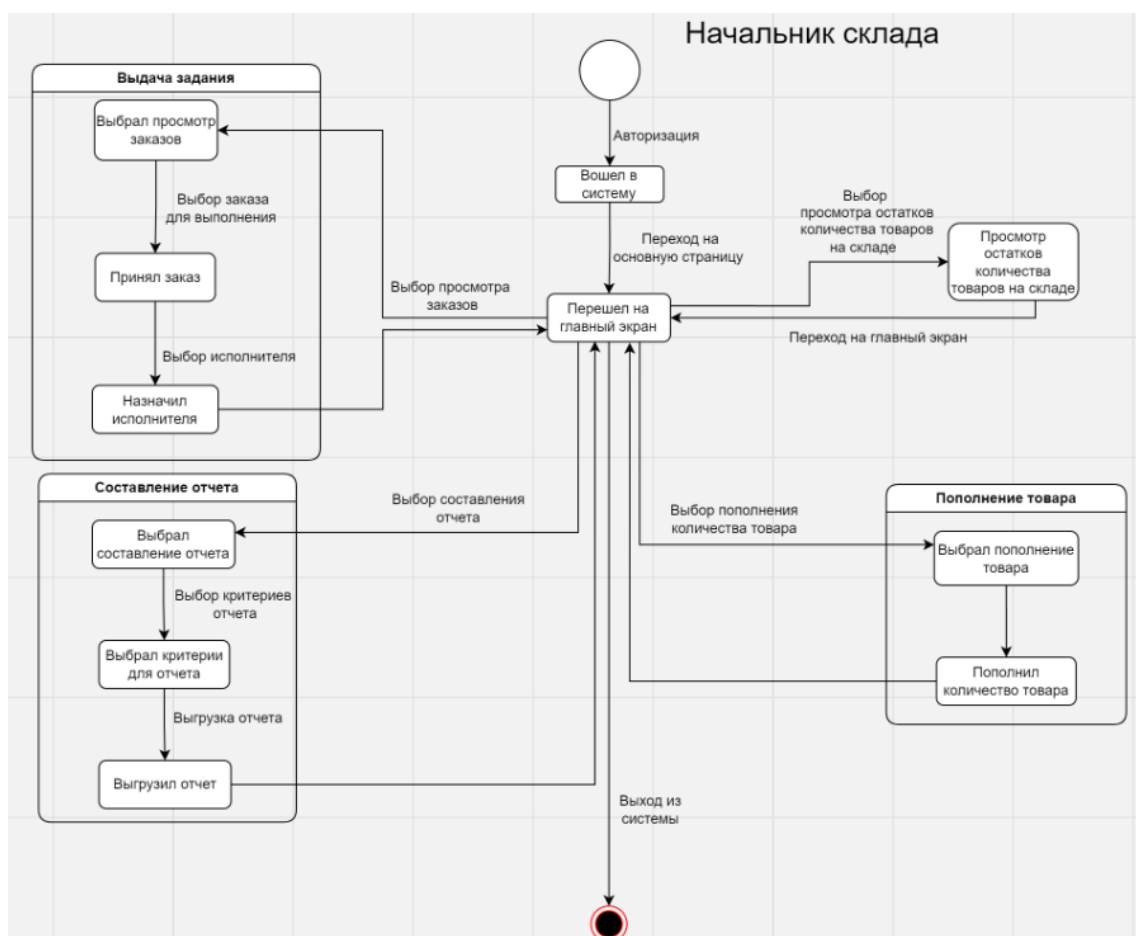


Рисунок 11– Диаграмма состояния начальника склада

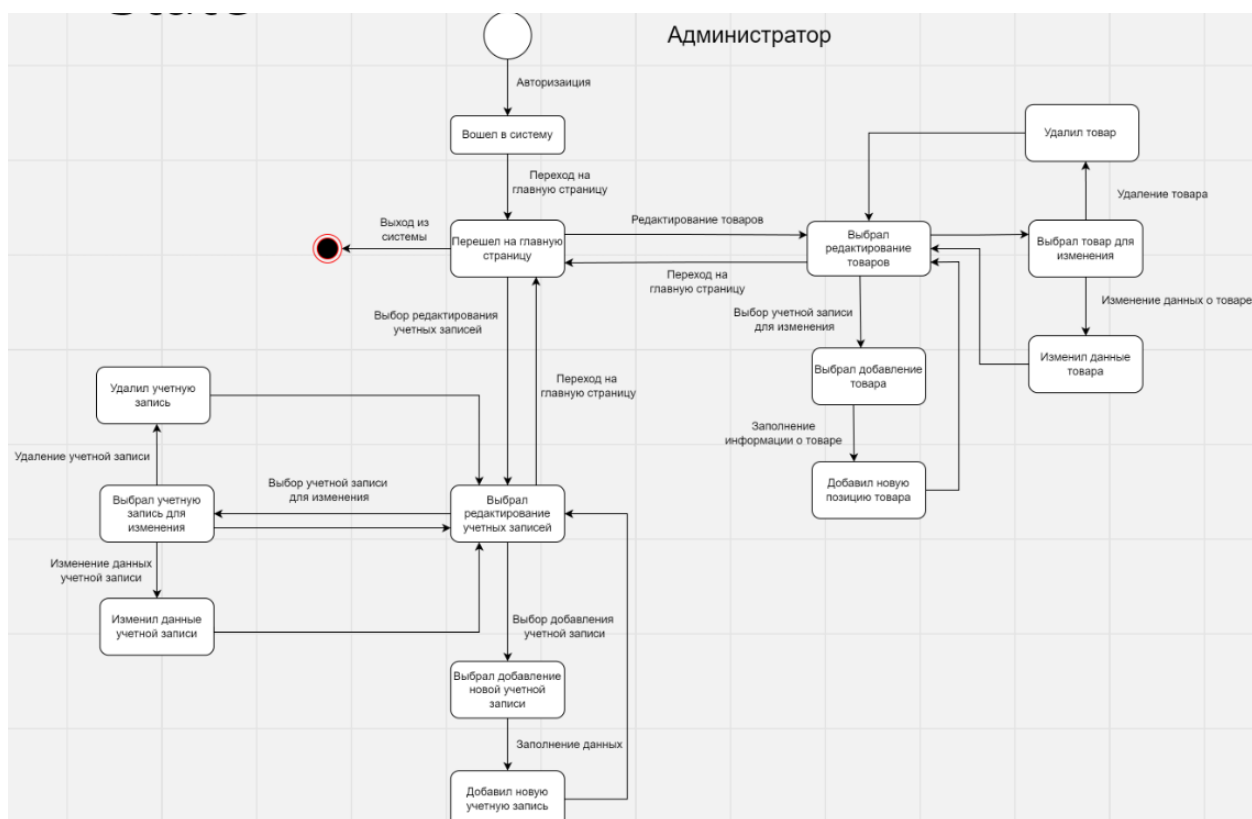


Рисунок 12– Диаграмма состояния администратора

2.3.5 Диаграмма объектов

Диаграмма объектов позволяет определить классы объектов, их атрибуты и методы, а также взаимодействие между ними. Она помогает разработчикам лучше понимать структуру системы и проектировать ее более эффективно. Данная диаграмма представлена на рисунке 13.

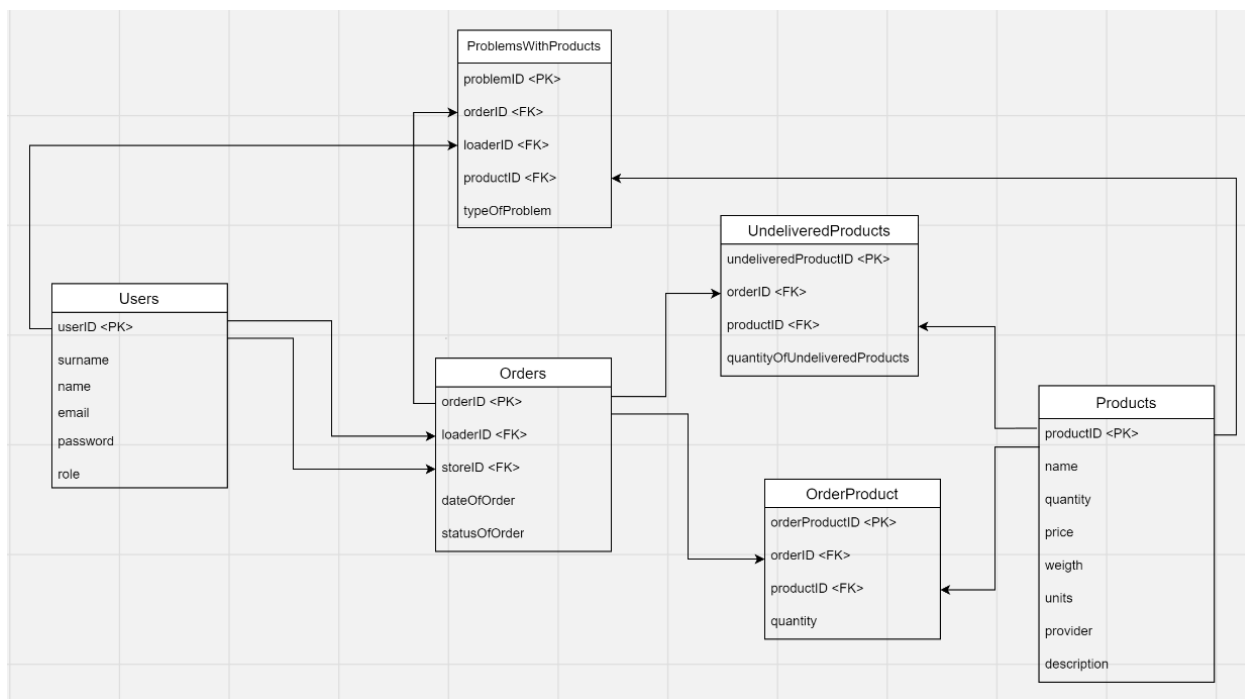


Рисунок 13– Диаграмма объектов

2.3.6 Диаграмма активности

Диаграмма активности помогает разработчикам лучше понимать процессы в системе, выявлять узкие места и оптимизировать их. Она также может использоваться для описания бизнес-процессов и управления проектами. Для данного проекта были спроектированы 3 диаграммы активности для премиум пользователя, авторизованного пользователя и неавторизованного пользователя. Данные диаграммы представлены на рисунке 14.

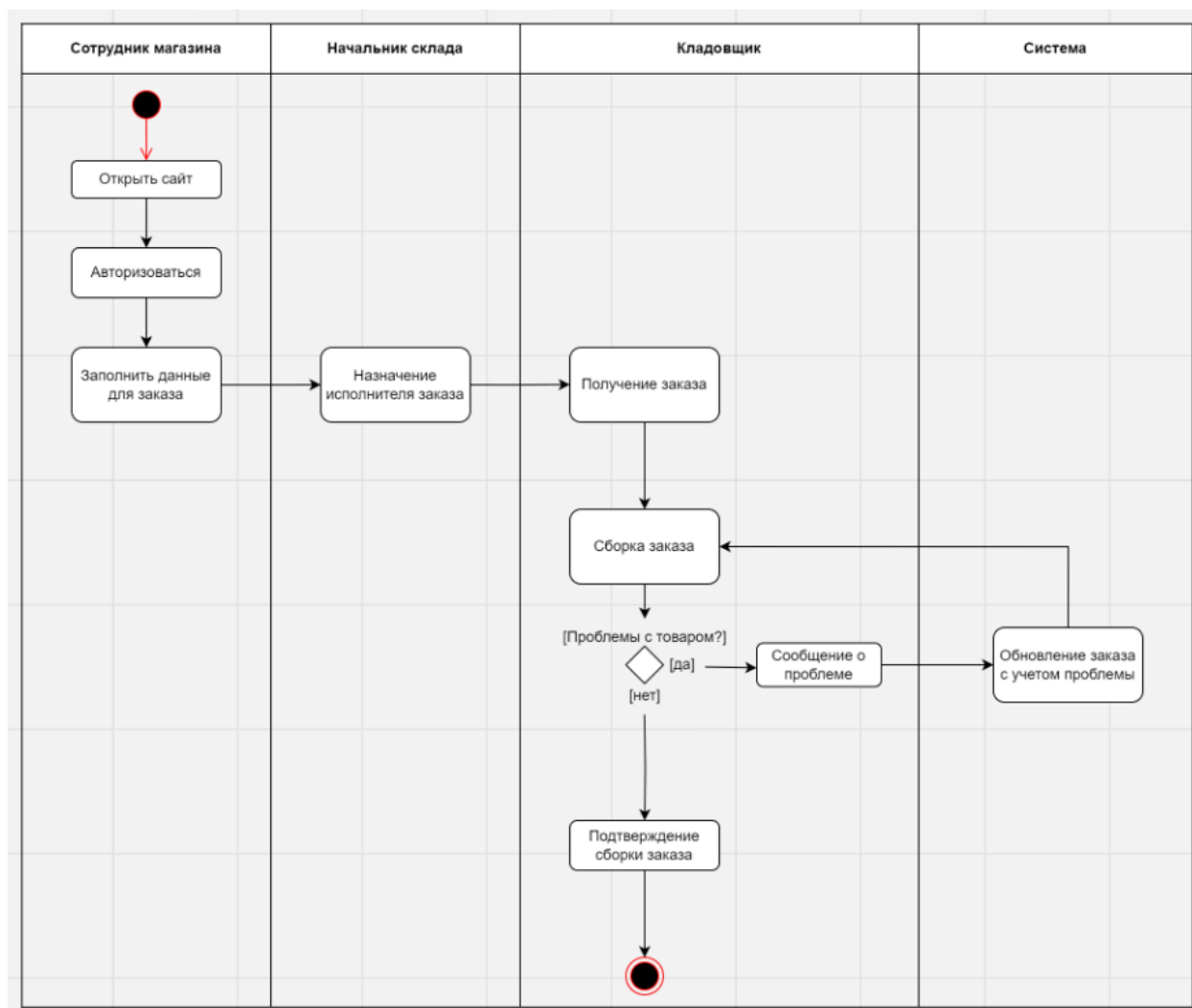


Рисунок 14 – Диаграмма активности

3 Реализация

3.1.1 Средства реализации

Ниже приведен перечень используемых технологий.

Backend

- Программная платформа Spring Boot;
- фреймворк Spring;
- СУБД PostgreSQL;
- API Swagger;
- язык программирования Java.

Frontend:

- язык гипертекстовой разметки HTML;
- формальный язык описания внешнего вида документа CSS;
- язык программирования JavaScript;
- клиент Certbot для создания и получения SSL сертификата;
- фреймворк Vue.js.

Инструменты для ведения документации:

- Miro;
- YouTrack;
- Figma.

Дополнительный инструментарий:

- GitHub.

3.1.2 Реализация базы данных

Для хранения данных была выбрана база данных PostgreSQL. Она является продуктом с открытым исходным кодом, который поддерживается многими серверами. PostgreSQL поддерживает множественные типы данных, такие как числа разной точности, тексты с различными кодировками, изображения, звуки, видео, XML-документы, JSON-объекты и многие другие.

3.1.3 Реализация клиентской части

Клиентская часть нашего приложения была реализована с помощью языка гипертекстовой разметки HTML для структурирования контента, формальный язык описания внешнего вида документа CSS для создания визуально привлекательного интерфейса, и язык программирования JavaScript для обеспечения динамичности и интерактивности.

Для обеспечения безопасности использовался клиент Certbot для создания и получения SSL-сертификатов, гарантируя безопасность данных, передаваемых между клиентом и сервером. Также был использован фреймворк Vue.js для создания масштабируемого и эффективного приложения.

3.1.4 Форма для экрана «Приветствие»

Данный экран предлагает ввести почту и пароль пользователя и нажать кнопку “Вход”. При успешной авторизации осуществляется переход на экран “Профиль” для авторизованных пользователей. В профиле пользователя есть возможность выйти из аккаунта, посмотреть политику конфиденциальности.



Войти

Рисунок 15 – Форма экрана «Приветствие»

3.1.5 Форма для экрана «Авторизации»

На данном экране предоставляется возможность ввести почту и пароль.

Авторизоваться
Добро Пожаловать

Электронная почта

example@example.com

Пароль

Авторизоваться

Рисунок 16 – Форма экрана «Авторизации»

3.1.6 Форма для экрана «Главная страница»

Данный экран появляется при входе в приложение авторизованного пользователя, в нем содержится информация о сменах сотрудника и задачах на день. Вверху расположены кнопки «Главная» и «Выход».



Рисунок 17 – Форма экрана «Главная страница»

3.1.7 Форма для экрана «Список заказа»

На этом экране кладовщик имеет возможность просмотреть состав заказа, а также есть возможность нажать кнопку «завершить».



Рисунок 18 – Форма экрана «Список заказа»

3.1.8 Форма для экрана «Профиль»

На этом экране есть возможность выйти из аккаунта. Посмотреть политику конфиденциальности приложения. Попросить помощи касательно работы приложения.

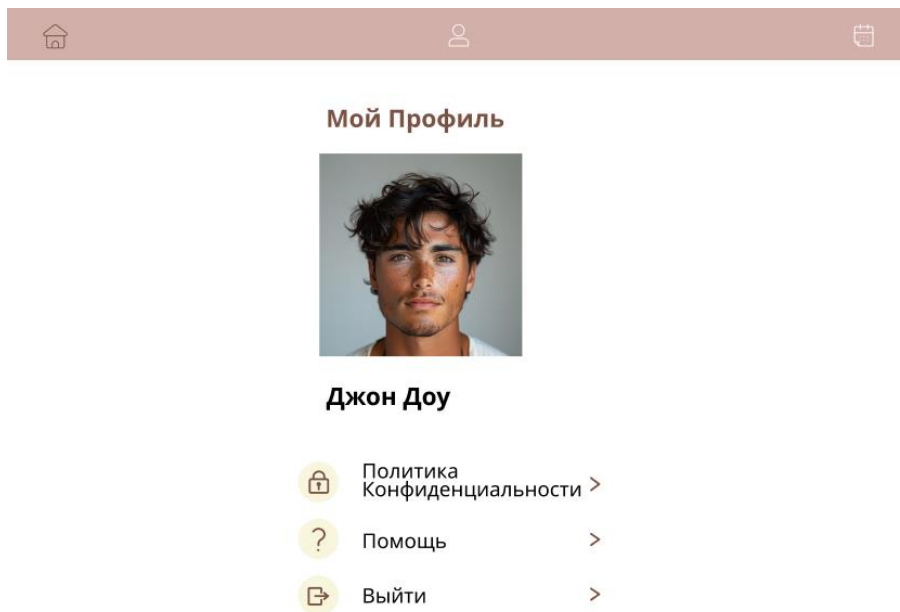


Рисунок 19 – Форма экрана «Профиль»

3.1.9 Форма для экрана «Регистрация»

Данный экран дает возможность администратору добавить нового пользователя. Нужно будет поочерёдно указать следующие данные:

- имя;
- фамилия;
- почта;
- номер телефона;
- дату рождения;
- должность.

Новый Аккаунт

Имя
John

Фамилия
Doe

Почта
example@example.com

Номер телефона
890000000000

Дата Рождения
DD / MM / YYY

должность
storekeeper

Зарегистрироваться

Рисунок 20 – Форма экрана «Регистрация»

3.1.10 Форма для экрана «Установка пароля»

На этом экране есть возможность установить пароль.

Установка Пароля

Пароль

Подтвердите Пароль

Создать Новый Пароль

Рисунок 21 – Форма экрана «Установка пароля»

3.1.11 Форма для экрана «Работники»

На данном экране можно просмотреть всех работников, а также отредактировать их профили, удалить их или добавить нового сотрудника.

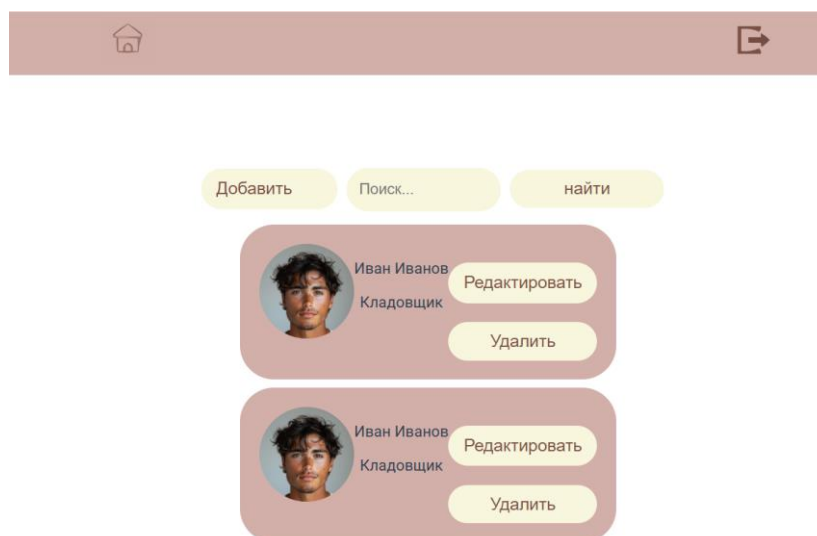


Рисунок 22 – Форма экрана «Работники»

3.1.12 Форма для экрана «Редактировать»

На данном экране можно отредактировать личные данные сотрудника и сохранить измененные данные.

Редактирование

Имя
John

Фамилия
Doe

Электронная почта
example@example.com

Номер телефона
89000000000

Дата Рождения
DD / MM / YYYY

Должность
storekeeper ▼

Редактировать

Рисунок 23 – Форма экрана «Редактировать»

3.1.13 Форма для экрана «Товар»

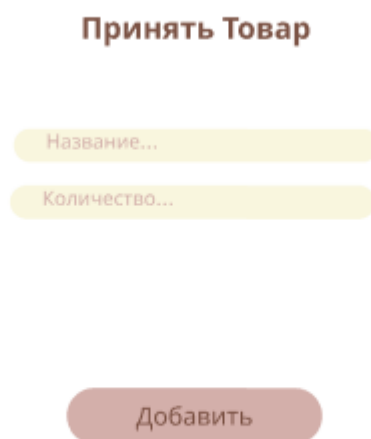
Данный экран предоставляет возможность добавить новый товар. Нужно будет поочерёдно указать следующие данные:

- название;
- описание;
- артикул;
- вес;
- единица измерения;
- поставщик.

После чего необходимо нажать кнопку «Добавить»

3.1.15 Форма для экрана «Принять товар»

На данном экране предоставляется возможность ввести название товара и указать количество, которое поступило на склад. А также есть возможность указать артикул, по которому можно будет найти товар на складе.



Принять Товар

Название...


Количество...

Добавить

Рисунок 26 – Форма экрана «Принять товар»

3.1.16 Форма для «Составить отчет»

На данном экране есть возможность составить отчет.



Отчет

Тип

Тип

Период

Период

Создать

Рисунок 27 – Форма экрана «Составить отчет»

3.1.17 Форма для экрана «Остатки товара»

На данном экране есть возможность посмотреть количество товара на складе. Есть возможность отсортировать по количеству.

Артикул	Название продукта	Цена	Вес	Мера измерения	Поставщик	Описание
1	Вареники	22	100	шт	five	Хорошие вареники
2	Вареники	22	100	шт	five	Хорошие вареники
3	Вареники	22	100	шт	five	Хорошие вареники
4	Вареники	22	100	шт	five	Хорошие вареники

Рисунок 28 – Форма экрана «Остатки товара»

3.2 Реализация серверной части

В качестве языка был выбран строго типизированный объектно-ориентированный язык программирования Java. Он остается очень популярным языком программирования в разработке благодаря своим мощным возможностям и широкому спектру инструментов для разработки. К тому же существует огромное количество фреймворков и библиотек, написанных на Java, которые в перспективе можно легко интегрировать в проект.

Также применяется программная платформа Docker для развертывания приложений.

3.2.1 Архитектура серверной части приложения

Серверная часть приложения реализована соответственно трехслойной архитектуре веб-приложения с API Rest с использованием фреймворка Spring boot. Данный фреймворк предоставляет возможности для работы с базами данных, а также механизм внедрения зависимостей, который позволяет сделать компоненты программы слабосвязанными, а всю программу в целом более гибкой, адаптируемой и расширяемой.

3.2.2 Диаграмма классов контролеров

Контроллеры в Spring Boot отвечают за обработку входящих HTTP-запросов и управление потоком выполнения приложения. Данные классы образуют так называемый первый микросервисный слой. Контроллеры определяют различные методы для обработки запросов. В нашем случае они представлены классами на рисунке

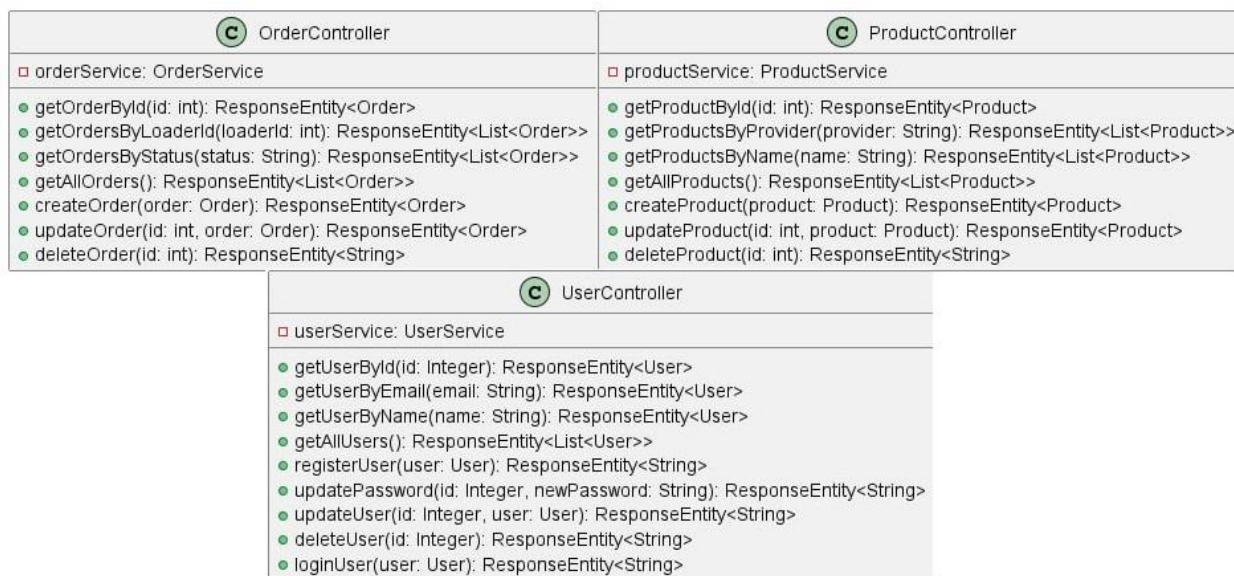


Рисунок 29 – Контроллеры

3.2.3 Диаграмма классов сервисов

Сервисы в Spring Boot представляют второй слой бизнес-логики приложения. Они содержат бизнес-логику и выполняют операции, которые требуются для обработки запросов контроллеров. Сервисы могут использовать другие компоненты, такие как репозитории, для доступа к данным. Классы сервисы представлены на рисунке

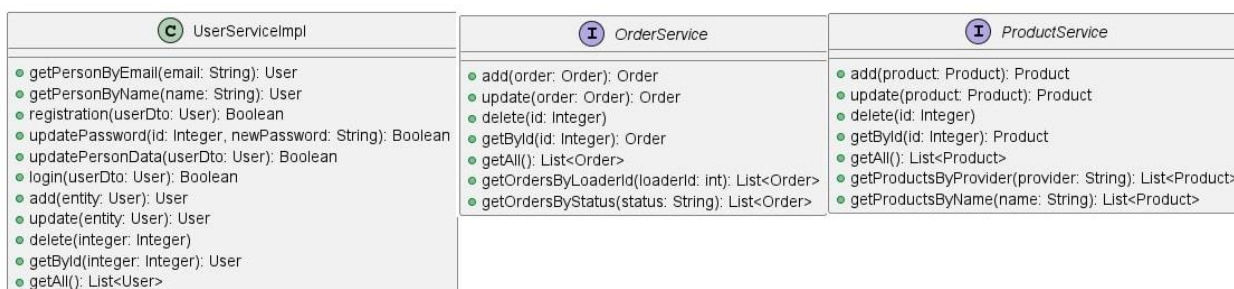


Рисунок 30 – Сервисы

3.2.4 Диаграмма классов репозиториев

Репозитории в Spring Boot обеспечивают доступ к данным, таким как базы данных или внешние сервисы. Данные классы образуют третий микросервисный слой. Они используются для выполнения операций чтения и записи данных. Репозитории используют специфические аннотации, такие как `query`, для определения пользовательских запросов к базе данных. Классы репозиториев представлены на рисунке


 <i>UserRepository</i>	 <i>OrderRepository</i>	 <i>ProductRepository</i>
<ul style="list-style-type: none">• <code>findByEmail(email: String): User</code>• <code>findByName(name: String): User</code>	<ul style="list-style-type: none">• <code>findByLoaderId(loaderId: int): List<Order></code>• <code>findByStatus(status: String): List<Order></code>	<ul style="list-style-type: none">• <code>findByProvider(provider: String): List<Product></code>• <code>findByName(name: String): List<Product></code>

Рисунок 31 – Диаграмма классов репозиториев

3.2.5 Диаграмма классов моделей

Модели представляют данные, с которыми работает приложение. Они являются простыми Java-классами, содержащими поля, геттеры и сеттеры для доступа к данным. Эти классы образуют последний микросервисный слой. Модели используются для передачи данных между контроллерами, сервисами и репозиториями. Классы моделей представлены на рисунке

3.2.6 Диаграмма классов DTO

DTO представляют объекты, используемые для передачи данных между слоями приложения или между приложением и клиентом. Они содержат только необходимые данные и не содержат бизнес-логики. В DTO используются в RESTful API для передачи данных между клиентом и сервером. Диаграмма классов DTO представлена на рисунке

3.2.7 Диаграмма классов исключений

Исключения в Java используются для обработки ошибок и необычных ситуаций, которые могут возникнуть во время выполнения программы. Они позволяют предусмотреть возможные проблемы и определить, как программа должна реагировать на них. Например, если программа пытается найти

пользователя из базы данных которого не существует, может быть сгенерировано исключение, и для обработки этой ситуации вынесется сообщения об ошибке. Классы исключений представлены на рисунке

3.2.8 Диаграмма классов конфигураций

Конфигурации в Spring Boot позволяют настраивать ваше приложение. Данные классы конфигурации, определяют бины и другие настройки приложения. Конфигурации используют аннотации `bean`, `value`, `autowired`, для настройки зависимостей и значений свойств. Классы конфигураций представлены на рисунке

3.2.9 Диаграмма классов сериализаторов

Классы сериализаторы, или в нашем случае мапперы, используются для преобразования данных между различными типами или моделями. Сериализация применяется для представления модели основного объекта в DTO. В Spring Boot для этого используется библиотека `MapStruct`, которая предоставляет функционал для генерации кода сериализации и десериализации. Данные классы представлены на рисунке

4 Аналитика

Для сбора метрик для приложения было выбрано AppMetrica от Яндекс. Это система для сбора данных об использовании приложения пользователями. Она отличается высокой скоростью и удобством настройки метрик для мобильных приложений. Кроме того, она предлагает интуитивный интерфейс и понятное руководство по использованию.



















Название цели	Описание	Номер цели
 1   Составление заказа	 составная цель клик по кнопке "Сделать заказ" (ID: 330907195): <ul style="list-style-type: none">• url содержит: /shop клик по кнопке "Добавить продукт" (ID: 330907196): <ul style="list-style-type: none">• url содержит: /accept	330907055  
 2   Выполнение задания	 составная цель клик по кнопке "Работа" (ID: 330907221): <ul style="list-style-type: none">• url содержит: /storekeeper клик по кнопке с номером заказа (ID: 330907222): <ul style="list-style-type: none">• url содержит: /work	330907220  
 3   Добавление сотрудника	 составная цель клик по кнопке "Сотрудники" (ID: 330907325): <ul style="list-style-type: none">• url содержит: /employees клик по кнопке "Добавить" (ID: 330907326): <ul style="list-style-type: none">• url содержит: /registration	330907324  

Рисунок 32 – Целевые события

<div> <div></div> <div>4</div> <div></div> <div></div> </div> <div>Составление отчета</div>	<div> <div></div> <div>составная цель</div> <div>330907383</div> <div></div> <div></div> </div> <div> <div>клик по кнопке "Составить отчет"</div> <div>(ID: 330907384):</div> <ul style="list-style-type: none"> • url содержит: /home </div> <div> <div>клик по кнопке "Составить отчет"</div> <div>(ID: 330907385):</div> <ul style="list-style-type: none"> • url содержит: /report </div>
<div> <div></div> <div>5</div> <div></div> <div></div> </div> <div>Добавление товара</div>	<div> <div></div> <div>составная цель</div> <div>330907930</div> <div></div> <div></div> </div> <div> <div>клик по кнопке "Товары"</div> <div>(ID: 330907931):</div> <ul style="list-style-type: none"> • url содержит: /product </div> <div> <div>заполнение полей и добавление товара</div> <div>(ID: 330907932):</div> <ul style="list-style-type: none"> • url содержит: /submit_product </div>

Рисунок 33 – Целевые события

Заключение

В ходе выполнения данного курсового проекта был выполнен анализ предметной области и аналогов разрабатываемого приложения.

Для разработки приложения были разработаны макеты интерфейса, выбрана платформа приложения, построены UML диаграммы.

Для контроля версий был создан репозиторий GitHub.

Список литературы

1. Крейга Уоллса Spring в действии. - Москва: Цветное, 2022.
2. Хансом Бергстеном Java Server Pages. - Москва: Эксмо, 2019.
3. Хассаном Джирдехом, Нейтом Мюрреем, Ари Лернером Fullstack Vue 3: The Complete Guide to Vue.js. - Санк-Петербург: Эксмо, 2021.
4. Карл Вигерс, Джой Битти Разработка требований к программному обеспечению. - Санк-Петербург: Русская редакция, 2014.