

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ

Кафедра информационных систем  
Направление подготовки  
09.03.04 – Программная инженерия

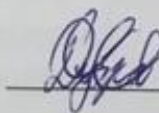
**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

(бакалаврская работа)

на тему:

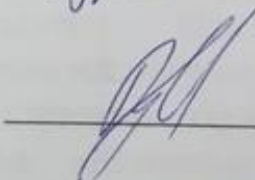
«Разработка автоматизированной информационной системы для учёта  
заказов спецтехники»

Студент группы 2181121



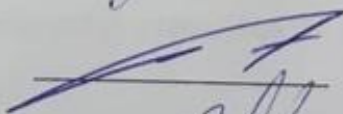
Е.И. Дубровский

Руководитель ВКР



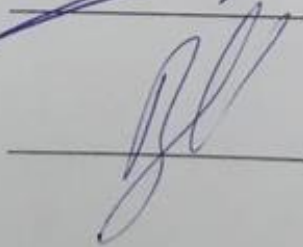
Р.А. Валиев

Нормоконтролер



Э.Л. Хазиев

Заведующий кафедрой  
информационных систем



Р.А. Валиев

Набережные Челны – 2022

## АННОТАЦИЯ

Выпускная квалификационная работа (бакалаврская работа) на тему «Разработка автоматизированной информационной системы для учёта заказов спецтехники» содержит 92 страницы пояснительной записки и 8 листов графического материала.

ИНФОРМАЦИОННАЯ СИСТЕМА, БАЗА ДАННЫХ, ПРОЕКТИРОВАНИЕ, УЧЁТ ЗАКАЗОВ, СПЕЦТЕХНИКА, ОБСЛУЖИВАНИЕ КЛИЕНТОВ.

Объект работы – ООО «ГрандТех», дилерский центр, специализирующийся на продаже спецтехники различных габаритов и производителей.

Целью данной работы является снижение среднего времени обслуживания клиентов за счет создания информационной системы для учёта заказов спецтехники, что позволит упростить взаимодействие между отделами.

Теоретическое исследование проводилось методом анализа литературы и нормативных справочников, положений предприятия, функциональных обязанностей. Практическое обследование объекта проводилось на основе системного анализа.

Разработана информационная система для учёта заказов спецтехники.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	5
1 Анализ требований к системе .....	7
1.1 Описание бизнес-требований .....	7
1.2 Разработка модели данных .....	10
1.2.1 Выделение существительных и глаголов .....	10
1.2.2 Разработка концептуальной карты .....	10
1.2.3 Разработка ER-модели.....	12
1.2.4 Разработка ER-модели с атрибутами сущностей.....	12
1.3 Моделирование бизнес-процессов .....	13
1.3.1 Описание предметной области. ....	13
1.3.2 Табличное представление результатов анализа процесса.....	13
1.3.3 BPMN2 модель процессов.....	15
1.4 Разработка пользовательских требований .....	16
1.4.1 Диаграмма вариантов использования.....	16
1.4.2 Спецификации для каждого варианта использования.....	19
1.5 Разработка функциональных требований .....	33
1.6 Разработка нефункциональных требований .....	42
1.7 Вывод по главе .....	43
2. Проектирование информационной системы.....	44
2.1 Функциональная модель системы .....	44
2.2 Разработка алгоритмов системы.....	46
2.2.1 Алгоритм авторизации .....	46
2.2.2 Добавление данных .....	47
2.2.3 Редактирование данных .....	48
2.2.4 Удаление данных.....	49
2.2.5 Восстановление данных .....	51
2.3 Проектирование базы данных.....	53
2.4 Вывод по главе .....	56

3. Реализация информационной системы .....	57
3.1 Реализация архитектуры информационной системы .....	57
3.2 Интерфейс информационной системы .....	57
3.2.1 Реализация страницы авторизации .....	57
3.2.2 Реализация страницы «Оформление/выдача заказов» .....	58
3.2.3 Реализация страницы «Закупка спецтехники» .....	61
3.2.4 Реализация страницы «Клиенты» .....	62
3.2.5 Реализация страницы «Ассортимент спецтехники» .....	64
3.2.6 Реализация страницы «Поставщики» .....	67
3.2.7 Реализация страницы «Сотрудники» .....	69
3.3 Вывод по главе .....	71
ЗАКЛЮЧЕНИЕ .....	72
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	73
ПРИЛОЖЕНИЕ А .....	75

## **ВВЕДЕНИЕ**

Дилерский центр «ГрандТех» специализируется на продаже спецтехники разных габаритов и от нескольких производителей. В текущей реализации процесса есть потребность в эффективном обслуживании клиентов. Проблема состоит в том, что доступ к актуальной информации о конкретном заказе есть только у отдела, сопровождающего заказ в этот момент, что не позволяет давать клиентам быструю обратную связь и вынуждает работников одного отдела отвлекать работников другого с целью получения актуальной информации.

Актуальность работы обусловлена необходимостью упростить взаимодействие между отделами и снизить среднее время обслуживания клиентов.

Объектом исследования в данной выпускной квалификационной работе является ООО «ГрандТех».

Предметом исследования будет являться система для учёта заказов спецтехники.

Целью данной работы является снижение среднего времени обслуживания клиентов за счет создания информационной системы для учёта заказов спецтехники, что позволит упростить взаимодействие между отделами.

Для достижения поставленной цели требуется решить следующие задачи:

- описать бизнес-требования к системе;
- разработать модель данных;
- смоделировать бизнес-процессы;
- разработать пользовательские требования;
- разработать функциональные требования;
- разработать нефункциональные требования;
- разработать функциональную модель;

- разработать алгоритмы системы;
- проектировать базу данных;
- реализовать информационную систему.

## **1 Анализ требований к системе**

### **1.1 Описание бизнес-требований**

Дилерский центр спецтехники «ГрандТех» испытывает трудности с обслуживанием клиентов из-за плохо согласованной работы отдела по продажам и отдела по управлению закупками. Продавцам приходится звонить в отдел по управлению закупками, чтобы узнать о статусе заказа. В свою очередь ответственному за закупки приходится лично отвечать на каждую заявку. Было принято решение разработать информационную систему для упрощения взаимодействия между отделами и для снижения среднего времени обслуживания клиентов.

Реализация системы предоставит бизнесу следующие возможности:

- возможность каждому продавцу самостоятельно проверить нужную ему информацию без звонков в отдел по управлению закупками;
- увеличение продуктивности работы отдела по продажам и отдела по управлению закупками за счёт продуманной удобной системы;
- минимизация ошибок при оформлении и сопровождении заказа, устранение организационного беспорядка.

Решение преследует следующие цели:

- внедрение системы учёта заказов в III квартале 2022 года;
- сократить среднее время на обслуживание клиентов на 10%.

Критерием успеха проекта является реализация базовой функциональности.

Решение представляет собой веб-приложение на платформе ASP.NET Core. Приложение позволит дилерскому центру сократить среднее время обслуживания клиентов за счёт облегчения взаимодействия между отделом по продажам и отделом по управлению закупками. Приложение позволит продавцам оформлять заказы клиентов, просматривать актуальную информацию о заказах и вносить изменения в заказ в связи со статусом оплаты

заказа и передачи доставленной техники клиенту. Ответственный за закупки сможет вносить в заказы изменения в связи с их статусом и обновлять предлагаемый дилерским центром ассортимент спецтехники.

К бизнес-рискам можно отнести потребность в более продвинутой информационной системе в случае увеличения компании или расширения спектра услуг, что может привести к большим затратам.

Неизменным фактором в процессе создания решения является то, что на это время разработчики не будут привлекаться для работы на других проектах.

Ниже приведены высокоуровневые функции будущего решения, среди которых:

- авторизация;
- управление учётными записями;
- оформление заказа;
- просмотр информации о заказах;
- просмотр информации о моделях спецтехники;
- управление заказами;
- обновление данных о клиентах;
- обновление ассортимента спецтехники;
- обновление данных о поставщиках;
- поиск, сортировка, фильтрация по ассортименту;
- поиск, сортировка, фильтрация по заказам;

Ниже приведён минимальный набор высокоуровневых функций, без которых решение не будет жизнеспособным:

- оформление заказа;
- просмотр информации о заказах;
- просмотр информации о моделях спецтехники;
- управление заказами;
- обновление данных о клиентах;
- обновление ассортимента спецтехники;
- обновление данных о поставщиках.



Ниже приведены ключевые участники проекта в форме роли, ценности от решения, отношения к проекту, интересов и ограничений:

- руководитель:
  - 1) Ценность: ускорение работы предприятия.
  - 2) Отношение: заинтересованность.
  - 3) Интересы: минимальные затраты, увеличение прибыли.
  - 4) Ограничения: стоимость обслуживания.
- продавец:
  - 1) Ценность: облегчение работы, полная информированность.
  - 2) Отношение: высокая заинтересованность.
  - 3) Интересы: уменьшение бумажной работы, свобода.
  - 4) Ограничения: глубокое ознакомление с программным интерфейсом.
- ответственный за закупки:
  - 1) Ценность: облегчение работы.
  - 2) Отношение: высокая заинтересованность.
  - 3) Интересы: уменьшение бумажной работы, свобода.
  - 4) Ограничения: глубокое ознакомление с программным интерфейсом.

Приоритетами проекта с позиции функций, качества, сроков, расходов и персонала являются:

- функции: все функции MVP должны быть выполнены.
- качество: основная функциональность должна работать без ошибок.
- срок: версия MVP должна быть запущена в тестовую эксплуатацию до конца II квартала 2022 года.
- расходы: приблизительно 200000 рублей на разработку, экономический расчёт не произведён.
- персонал: руководитель проекта, бэк-энд разработчик, фронт-энд разработчик, тестировщик.

Для роли «Ответственный за закупки» определены следующие варианты использования:

- просмотр информации о заказах;
- просмотр информации о моделях спецтехники;
- управление заказами;
- обновление данных о поставщиках;
- обновление ассортимента спецтехники.

Для роли «Продавец» определены следующие варианты использования:

- оформление заказа;
- просмотр информации о моделях спецтехники;
- просмотр ассортимента спецтехники;
- управление заказами;
- обновление данных о клиентах.

## 1.2 Разработка модели данных

### 1.2.1 Выделение существительных и глаголов

В ходе анализа бизнес-требований был сформирован список из существительных-объектов, глаголов и существительных-ролей. Данный список приведён в рисунке 1.1.

Существительные, глаголы и роли			
№	Существительные	Глаголы	Роли
1	Заказ	Содержит	Продавец
2	Спецтехника	Оформляет	Ответственный за закупки
3	Клиент	Обновляет	
4	Поставщик		

Рисунок 1.1 - Существительные, глаголы и роли

### 1.2.2 Разработка концептуальной карты

В ходе анализа была построена концептуальная карта предметной области [2], приведённая на рисунке 1.2.

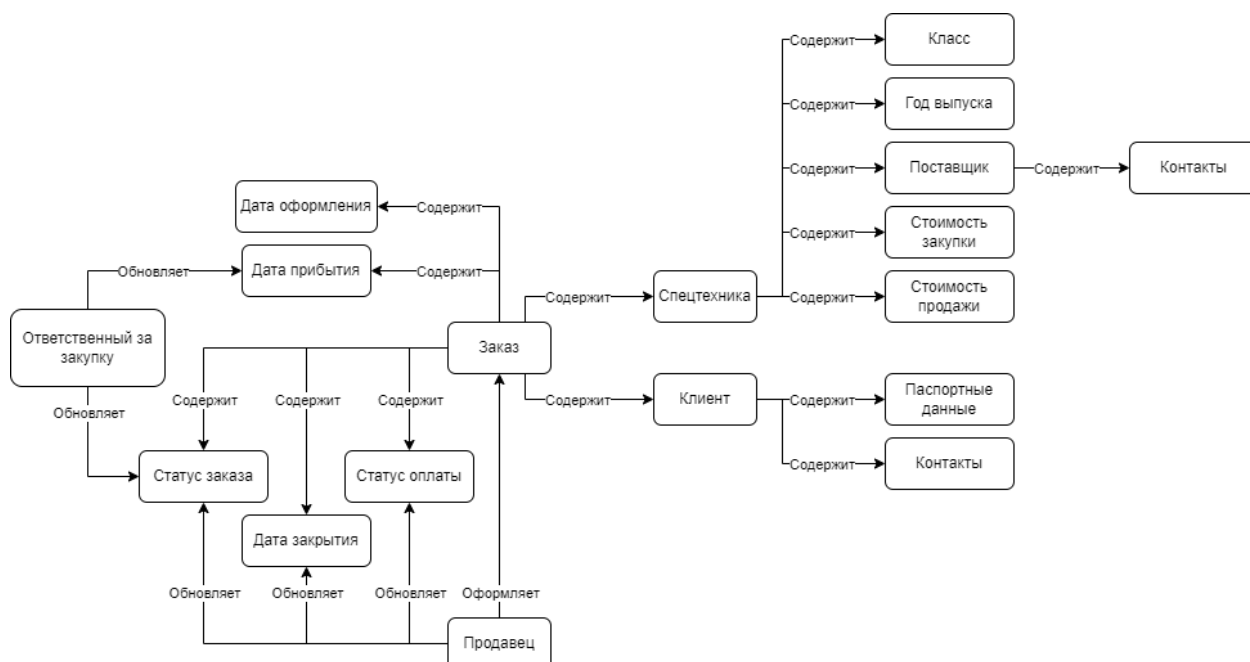


Рисунок 1.2 - Концептуальная карта предметной области

Были выявлены основные сущности: заказ, спецтехника, клиент и поставщик.

Спецтехника содержит класс, поставщика, год выпуска стоимость закупки и стоимость продажи.

Клиент содержит паспортные данные и контакты.

Поставщик содержит контакты.

Продавец оформляет заказ, содержащий клиента, спецтехнику, дату оформления, статус оплаты и статус заказа.

После получения предоплаты или полной оплаты продавец обновляет статус оплаты.

Совершая закупку заказа, ответственный за закупку обновляет статус заказа и дату прибытия.

После передачи заказа клиенту продавец закрывает заказ, обновляя статус заказа и дату закрытия.

### 1.2.3 Разработка ER-модели

В ходе анализа было выявлено, что одной из основных сущностей является «Заказ». Заказ обязательно содержит клиента и спецтехнику. Спецтехника обязательно содержит поставщика. На рисунке 1.3 показана ER-модель проекта без атрибутов [3].



Рисунок 1.3 – ER-модель проекта без атрибутов

Заказ всегда должен иметь одну спецтехнику и одного клиента. Клиент может иметь много заказов.

Спецтехника всегда должна иметь одного поставщика. Поставщик может иметь много спецтехники.

### 1.2.4 Разработка ER-модели с атрибутами сущностей

На рисунке 1.4 показана ER-модель проекта с атрибутами.



Рисунок 1.4 – ER-модель проекта с атрибутами

Атрибутами сущности «Заказ» являются id заказа, id спецтехники, id клиента, дата оформления, дата прибытия, дата закрытия, статус заказа и статус оплаты.

Атрибутами сущности «Спецтехника» являются id спецтехники, наименование, класс, год выпуска, id поставщика, стоимость закупки и стоимость продажи.

Атрибутами сущности «Поставщик» являются id поставщика, наименование и контакты.

Атрибутами сущности «Клиент» являются id клиента, имя, паспортные данные и контакты.

### **1.3 Моделирование бизнес-процессов**

#### **1.3.1 Описание предметной области.**

В текущей реализации процесса есть потребность в эффективном обслуживании клиентов. Проблема состоит в том, что доступ к актуальной информации о конкретном заказе есть только у отдела, сопровождающего заказ в этот момент, что не позволяет давать клиентам быструю обратную связь и вынуждает одних работников одного отдела отвлекать работников другого с целью получения актуальной информации.

#### **1.3.2 Табличное представление результатов анализа процесса**

На рисунках 1.5-1.8 приведены результаты анализа процессов ТО-ВЕ в табличном виде.

Анализ процесса оформления заказа			
№	Название	Тип	Роль
1	Заказ оформлен	Результат	Продавец
2	Обновить статус оплаты	Действие	Продавец
3	Аванс получен	Событие	Продавец
4	Запросить оплату аванса	Действие	Продавец
5	Оформить заказ	Действие	Продавец
6	Начало оформления заказа	Событие	Продавец

Рисунок 1.5 – Анализ процесса оформления заказа (скриншот из Excel)

Анализ процесса закупки заказа у поставщика			
№	Название	Тип	Роль
1	Ожидание доставки	Результат	Ответственный за закупки
2	Уведомление клиента о дате поставки	Событие	Ответственный за закупки
3	Обновить статус заказа	Действие	Ответственный за закупки
4	Заказать поставку	Действие	Ответственный за закупки
5	Заказ отменён	Результат	Ответственный за закупки
6	Уведомление клиента об отмене	Событие	Ответственный за закупки
7	Отменить заказ	Действие	Ответственный за закупки
8	Нет возможности закупки	Событие	Ответственный за закупки
9	Есть возможность закупки	Событие	Ответственный за закупки
10	Узнать о возможности закупки	Действие	Ответственный за закупки
11	Начало закупки	Событие	Ответственный за закупки

Рисунок 1.6 – Анализ процесса закупки заказа у поставщика (скриншот из Excel)

Анализ процесса уведомления клиента			
№	Название	Тип	Роль
1	Уведомление получено	Результат	Продавец
2	Уведомить клиента о статусе заказа	Действие	Продавец
3	Нужно уведомить клиента	Событие	Продавец

Рисунок 1.7 – Анализ процесса уведомления клиента (скриншот из Excel)

Анализ процесса передачи заказа клиенту			
№	Название	Тип	Роль
1	Заказ закрыт	Результат	Продавец
2	Закрыть заказ	Действие	Продавец
3	Передать заказ клиенту	Событие	Продавец
4	Обновить статус оплаты	Действие	Продавец
5	Оплата получена	Событие	Продавец
6	Запросить оплату остальной суммы	Действие	Продавец
7	Клиент прибыл за заказом	Событие	Продавец
8	Уведомление клиента о прибытии заказа	Событие	Продавец
9	Заказ прибыл	Событие	Продавец

Рисунок 1.8 – Анализ процесса передачи заказа клиенту (скриншот из Excel)

### 1.3.3 BPMN2 модель процессов

На рисунке 1.9 приведена BPMN2 модель процессов TO-BE [4].

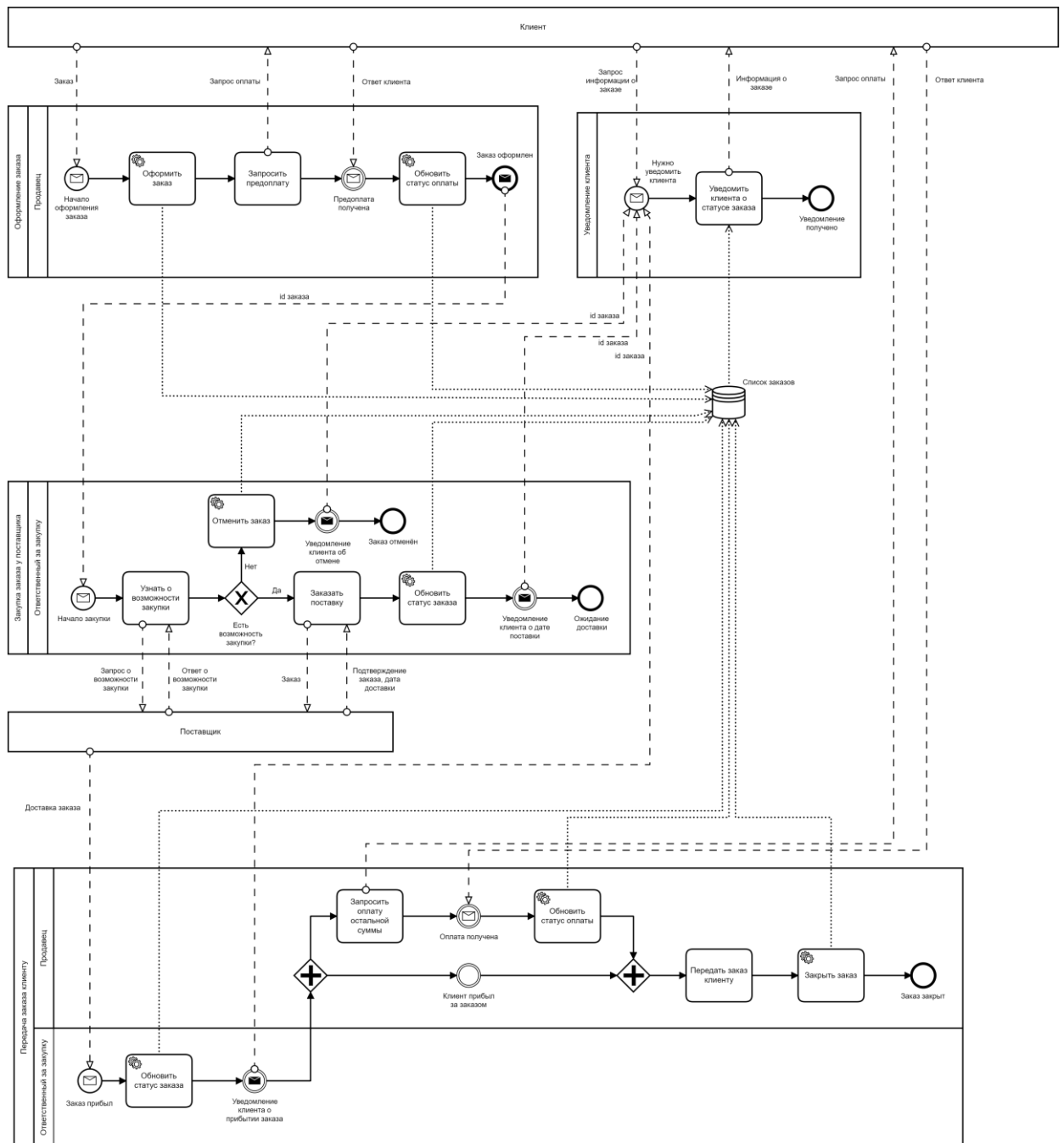


Рисунок 1.9 – BPMN2 модель процессов TO-BE

Процесс оформления заказа начинается с потребности клиента в модели спецтехники. Продавец оформляет заказ в соответствии с требованиями клиента и запрашивает у клиента предоплату заказа. После получения

предоплаты продавец обновляет статус оплаты заказа в системе, тем самым передавая заказ на закупку.

Ответственный за закупку проверяет возможность закупки требуемой спецтехники у поставщика. При отсутствии возможности закупки ответственный за закупку отменяет заказ и уведомляет клиента об отмене. При наличии возможности закупки ответственный за закупку осуществляет закупку требуемой спецтехники у поставщика и обновляет статус заказа в системе. Затем он сообщает клиенту дату поставки, предоставленную поставщиком.

После получения заказа от поставщика ответственный за закупку обновляет статус заказа и уведомляет клиента о прибытии заказа. Затем продавец запрашивает у клиента оплату оставшейся суммы заказа. После получения оплаты продавец обновляет статус оплаты заказа. Продавец передаёт заказ прибывшему клиенту и закрывает заказ.

## **1.4 Разработка пользовательских требований**

### **1.4.1 Диаграмма вариантов использования**

В ходе анализа бизнес-требований, бизнес-процессов и модели данных были выявлены варианты использования и потенциальные акторы: «Продавец», «Ответственный за закупку» и «Администратор».

Для актора «Продавец» были выявлены следующие варианты использования:

- авторизоваться;
- оформить заказ;
- обновить статус оплаты заказа;
- закрыть заказ;
- удалить заказ;
- посмотреть заказы конкретного клиента;



- редактировать данные клиента;
- посмотреть заказы на конкретную модель спецтехники;
- посмотреть список моделей спецтехники от конкретного поставщика.

Для актора «Ответственный за закупку» были выявлены следующие варианты использования:

- авторизоваться;
- обновить статус заказа;
- отменить заказ;
- посмотреть заказы конкретного клиента;
- посмотреть заказы на конкретную модель спецтехники;
- добавить модель спецтехники;
- редактировать данные о модели спецтехники;
- удалить модель спецтехники из ассортимента;
- восстановить модель спецтехники в ассортименте;
- посмотреть список моделей спецтехники от конкретного поставщика;
- добавить поставщика;
- редактировать данные о поставщике;
- удалить поставщика;
- восстановить поставщика.

Для актора «Администратор» были определены все варианты использования акторов «Продавец» и «Ответственный за закупку», а также выявлены следующие уникальные варианты использования:

- посмотреть заказы конкретного сотрудника;
- добавить пользователя;
- редактировать учётную запись пользователя;
- удалить учётную запись пользователя;
- восстановить учётную запись пользователя.

На рисунке 1.10 представлена диаграмма вариантов использования для акторов «Продавец» и «Ответственный за закупку».

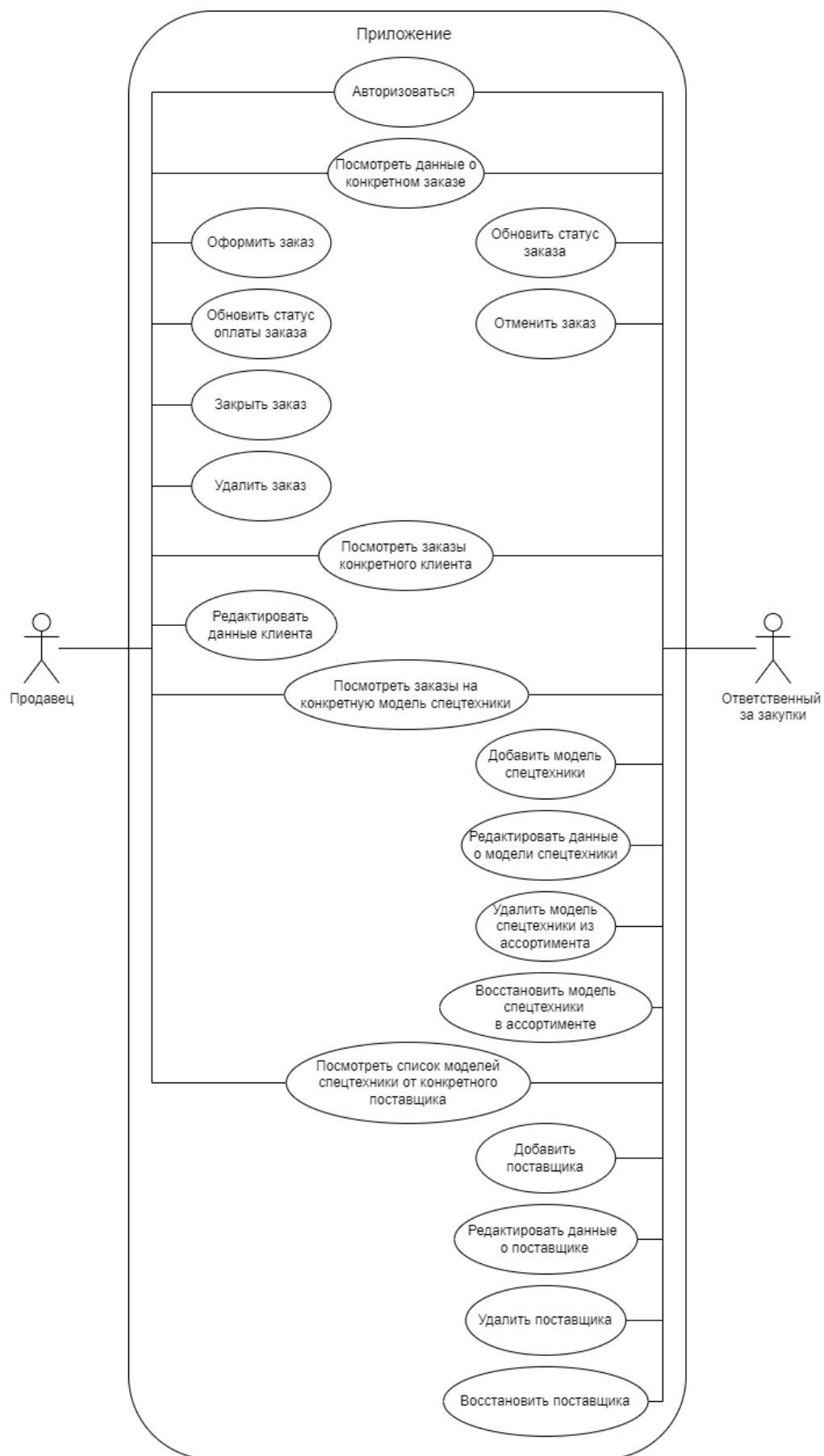


Рисунок 1.10 – Диаграмма вариантов использования для акторов «Продавец» и «Ответственный за закупку»

На рисунке 1.11 представлена диаграмма уникальных вариантов использования для актора «Администратор».

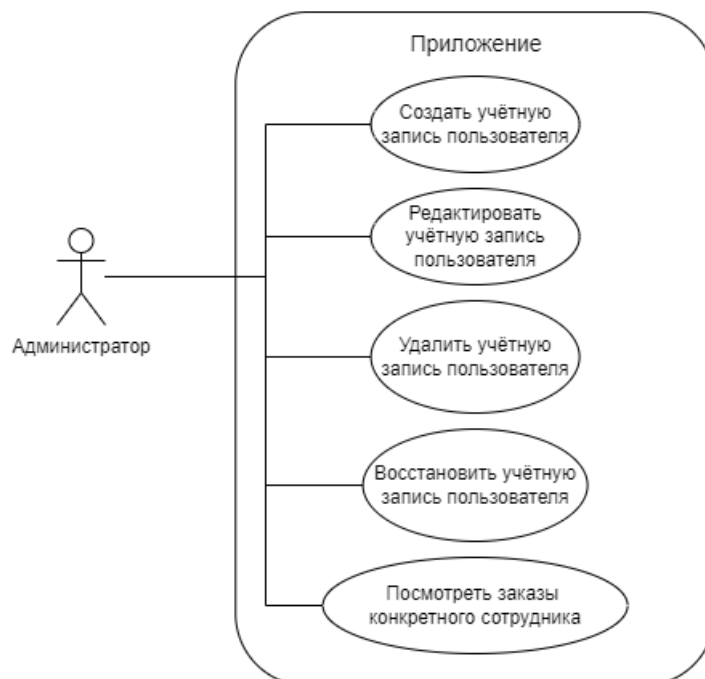


Рисунок 1.11 – Диаграмма уникальных вариантов использования для актора «Администратор»

#### 1.4.2 Спецификации для каждого варианта использования.

В таблицах 1.1-1.24 приведены спецификации для вариантов использования [6].

Таблица 1.1 - Вариант использования «Авторизоваться»

УС-1. Авторизоваться	
<b>Определение / Ценность</b>	Операция предназначена для получения пользователем возможности взаимодействовать с системой в соответствии с правами его учётной записи.
<b>Пользователь</b>	Администратор, продавец, ответственный за закупки
<b>Предусловие</b>	Пользователь не авторизирован и инициировал выполнение операции
<b>Основной сценарий</b>	1. Система предоставляет форму для заполнения данных для входа учётной записи.

Продолжение таблицы 1.1

	2. Пользователь заполняет форму и подтверждает авторизацию. 3. Система отображает уведомление об успешной авторизации.
<b>Альтернативные пути</b>	2а. Пользователь отказывается от выполнения операции. 2а.1. Система завершает выполнение операции. 2б. Пользователь неверно заполняет форму. 2б.1. Система отображает уведомление об ошибке. 2б.2. Работа возобновляется с шага 1.
<b>Результат</b>	Пользователь авторизован в системе.

Таблица 1.2 - Вариант использования «Создать учётную запись пользователя»

<b>УС-2. Создать учётную запись пользователя</b>	
<b>Определение / Ценность</b>	Операция предназначена для создания учётной записи пользователя с целью последующего взаимодействия с системой в соответствии с правами данной учётной записи.
<b>Пользователь</b>	Администратор
<b>Предусловие</b>	Пользователь авторизован и инициировал выполнение операции
<b>Основной сценарий</b>	1. Система предоставляет форму для заполнения данных будущей учётной записи. 2. Пользователь заполняет форму и подтверждает создание учётной записи. 3. Система отображает уведомление об успешном создании учётной записи.
<b>Альтернативные пути</b>	2а. Пользователь отказывается от выполнения операции. 2а.1. Система завершает выполнение операции. 2б. Пользователь неверно заполняет форму. 2б.1. Система отображает уведомление об ошибке. 2б.2. Работа возобновляется с шага 1.
<b>Результат</b>	Создана учётная запись пользователя

Таблица 1.3 - Вариант использования «Посмотреть заказы конкретного сотрудника»

<b>УС-3. Посмотреть заказы конкретного сотрудника</b>	
<b>Определение / Ценность</b>	Операция предназначена для получения списка заказов выбранного сотрудника.
<b>Пользователь</b>	Администратор
<b>Предусловие</b>	Пользователь авторизован и инициировал выполнение операции
<b>Основной сценарий</b>	1. Система отображает список сотрудников. 2. Пользователь выбирает нужного сотрудника. 3. Система отображает список заказов выбранного сотрудника.
<b>Альтернативные пути</b>	2а. Пользователь отказывается от выполнения операции. 2а.1. Система завершает выполнение операции.
<b>Результат</b>	Список заказов выбранного сотрудника доступен.

Таблица 1.4 - Вариант использования «Редактировать учётную запись пользователя»

<b>УС-4. Редактировать учётную запись пользователя</b>	
<b>Определение / Ценность</b>	Операция предназначена для изменения данных об учётной записи пользователя в связи с ошибкой при её создании или по запросу пользователя.
<b>Пользователь</b>	Администратор
<b>Предусловие</b>	Пользователь авторизован и инициировал выполнение операции
<b>Основной сценарий</b>	1. Система отображает список созданных учётных записей. 2. Пользователь выбирает учётную запись для редактирования. 3. Система предоставляет форму для заполнения новых данных. 4. Пользователь заполняет форму и подтверждает редактирование учётной записи. 5. Система отображает уведомление об успешном редактировании учётной записи.
<b>Альтернативные пути</b>	2а. Пользователь отказывается от выполнения операции. 2а.1. Система завершает выполнение операции. 4а. Пользователь отказывается от выполнения операции. 4а.1. Система завершает выполнение операции.

Продолжение таблицы 1.4

	<p>46. Пользователь неверно заполняет форму.</p> <p>46.1. Система отображает уведомление об ошибке.</p> <p>46.2. Работа возобновляется с шага 3.</p>
<b>Результат</b>	Данные учётной записи отредактированы.

Таблица 1.5 - Вариант использования «Удалить учётную запись пользователя»

<b>УС-5. Удалить учётную запись пользователя</b>	
<b>Определение / Ценность</b>	Операция предназначена для удаления учётной записи пользователя для предотвращения авторизации с использованием данных этой записи.
<b>Пользователь</b>	Администратор
<b>Предусловие</b>	Пользователь авторизован и инициировал выполнение операции
<b>Основной сценарий</b>	<p>1. Система отображает список созданных учётных записей.</p> <p>2. Пользователь выбирает учётную запись для удаления.</p> <p>3. Система запрашивает подтверждение удаления.</p> <p>4. Пользователь подтверждает удаление.</p> <p>5. Система отображает уведомление об успешном удалении учётной записи.</p>
<b>Альтернативные пути</b>	<p>2а. Пользователь отказывается от выполнения операции.</p> <p>2а.1. Система завершает выполнение операции.</p> <p>4а. Пользователь не даёт подтверждение.</p> <p>4а.1. Работа возобновляется с шага 1.</p>
<b>Результат</b>	Учётная запись отмечена как удалённая.

Таблица 1.6 - Вариант использования «Восстановить учётную запись пользователя»

<b>УС-6. Восстановить учётную запись пользователя</b>	
<b>Определение / Ценность</b>	Операция предназначена для восстановления учётной записи пользователя для возобновления возможности авторизации с использованием данных этой записи.
<b>Пользователь</b>	Администратор

Продолжение таблицы 1.6

<b>Предусловие</b>	Пользователь авторизован и инициировал выполнение операции
<b>Основной сценарий</b>	1. Система отображает список удалённых учётных записей. 2. Пользователь выбирает учётную запись для восстановления. 3. Система запрашивает подтверждение восстановления. 4. Пользователь подтверждает восстановление. 5. Система отображает уведомление об успешном восстановлении учётной записи.
<b>Альтернативные пути</b>	2а. Пользователь отказывается от выполнения операции. 2а.1. Система завершает выполнение операции. 4а. Пользователь не даёт подтверждение. 4а.1. Работа возобновляется с шага 1.
<b>Результат</b>	Учётная запись восстановлена.

Таблица 1.7 - Вариант использования «Посмотреть информацию о конкретном заказе»

<b>УС-7. Посмотреть данные о конкретном заказе</b>	
<b>Определение / Ценность</b>	Операция предназначена для получения данных о заказе.
<b>Пользователь</b>	Администратор, продавец, ответственный за закупки
<b>Предусловие</b>	Пользователь авторизован и инициировал выполнение операции
<b>Основной сценарий</b>	1. Система отображает список заказов. 2. Пользователь выбирает нужный заказ. 3. Система отображает данные о заказе.
<b>Альтернативные пути</b>	2а. Пользователь отказывается от выполнения операции. 2а.1. Система завершает выполнение операции.
<b>Результат</b>	Данные о заказе получены.

Таблица 1.8 - Вариант использования «Посмотреть заказы конкретного клиента»

<b>УС-8. Посмотреть заказы конкретного клиента</b>	
<b>Определение / Ценность</b>	Операция предназначена для получения списка заказов выбранного клиента.

Продолжение таблицы 1.8

<b>Пользователь</b>	Администратор, продавец, ответственный за закупки
<b>Предусловие</b>	Пользователь авторизирован и инициировал выполнение операции
<b>Основной сценарий</b>	1. Система отображает список клиентов. 2. Пользователь выбирает нужного клиента. 3. Система отображает список заказов выбранного клиента.
<b>Альтернативные пути</b>	2а. Пользователь отказывается от выполнения операции. 2а.1. Система завершает выполнение операции.
<b>Результат</b>	Список заказов выбранного клиента доступен.

Таблица 1.9 - Вариант использования «Посмотреть заказы на конкретную модель спецтехники»

<b>УС-9. Посмотреть заказы на конкретную модель спецтехники</b>	
<b>Определение / Ценность</b>	Операция предназначена для получения списка заказов на конкретную модель спецтехники.
<b>Пользователь</b>	Администратор, продавец, ответственный за закупки
<b>Предусловие</b>	Пользователь авторизирован и инициировал выполнение операции
<b>Основной сценарий</b>	1. Система отображает список доступной спецтехники. 2. Пользователь выбирает нужную модель спецтехники. 3. Система отображает список заказов на выбранную модель спецтехники.
<b>Альтернативные пути</b>	2а. Пользователь отказывается от выполнения операции. 2а.1. Система завершает выполнение операции.
<b>Результат</b>	Список заказов на выбранную модель спецтехники получен.

Таблица 1.10 - Вариант использования «Оформить заказ»

<b>УС-10. Оформить заказ</b>	
<b>Определение / Ценность</b>	Операция предназначена для добавления нового заказа в список заказов, а также нового клиента при условии, что данные клиента не были использованы при оформлении ранее.
<b>Пользователь</b>	Продавец
<b>Предусловие</b>	Пользователь авторизирован и инициировал выполнение операции



Продолжение таблицы 1.10

<b>Основной сценарий</b>	<p>1. Система предоставляет форму для оформления заказа.</p> <p>2. Пользователь заполняет форму и подтверждает оформление заказа.</p> <p>3. Система отображает уведомление об успешном оформлении заказа.</p>
<b>Альтернативные пути</b>	<p>2а. Пользователь неверно заполняет форму.</p> <p>2а.1. Система отображает уведомление об ошибке.</p> <p>2а.2. Работа возобновляется с шага 1.</p> <p>2б. Пользователь отказывается от выполнения операции.</p> <p>4б.1. Система завершает выполнение операции.</p>
<b>Результат</b>	Заказ оформлен.

Таблица 1.11 - Вариант использования «Закрыть заказ»

<b>УС-11. Закрыть заказ</b>	
<b>Определение / Ценность</b>	Операция предназначена для изменения информации о статусе заказа в связи с его закрытием.
<b>Пользователь</b>	Продавец
<b>Предусловие</b>	Пользователь авторизирован и инициировал выполнение операции
<b>Основной сценарий</b>	<p>1. Система отображает список заказов.</p> <p>2. Пользователь выбирает заказ для закрытия.</p> <p>3. Система запрашивает подтверждение закрытия.</p> <p>4. Пользователь подтверждает закрытие.</p> <p>5. Система отображает уведомление об успешном закрытии заказа.</p>
<b>Альтернативные пути</b>	<p>2а. Пользователь отказывается от выполнения операции.</p> <p>2а.1. Система завершает выполнение операции.</p> <p>4а. Пользователь не даёт подтверждение.</p> <p>4а.1. Работа возобновляется с шага 1.</p>
<b>Результат</b>	Заказ закрыт.

Таблица 1.12 - Вариант использования «Отменить заказ»

<b>УС-12. Отменить заказ</b>	
<b>Определение / Ценность</b>	Операция предназначена для отмены заказа в связи с невозможностью его исполнения.
<b>Пользователь</b>	Ответственный за закупки
<b>Предусловие</b>	Пользователь авторизирован и инициировал выполнение операции
<b>Основной сценарий</b>	<ol style="list-style-type: none"> <li>1. Система отображает список заказов.</li> <li>2. Пользователь выбирает заказ для отмены.</li> <li>3. Система запрашивает подтверждение отмены.</li> <li>4. Пользователь подтверждает отмену.</li> <li>5. Система отображает уведомление об успешной отмене заказа.</li> </ol>
<b>Альтернативные пути</b>	<ol style="list-style-type: none"> <li>2а. Пользователь отказывается от выполнения операции. <ol style="list-style-type: none"> <li>2а.1. Система завершает выполнение операции.</li> </ol> </li> <li>4а. Пользователь не даёт подтверждение. <ol style="list-style-type: none"> <li>4а.1. Работа возобновляется с шага 1.</li> </ol> </li> </ol>
<b>Результат</b>	Заказ помечен как отменённый.

Таблица 1.13 - Вариант использования «Удалить заказ»

<b>УС-13. Удалить заказ</b>	
<b>Определение / Ценность</b>	Операция предназначена для удаления заказа из-за ошибки при его оформлении.
<b>Пользователь</b>	Продавец,
<b>Предусловие</b>	Пользователь авторизирован и инициировал выполнение операции
<b>Основной сценарий</b>	<ol style="list-style-type: none"> <li>1. Система отображает список заказов.</li> <li>2. Пользователь выбирает заказ для удаления.</li> <li>3. Система запрашивает подтверждение удаления.</li> <li>4. Пользователь подтверждает удаление.</li> <li>5. Система отображает уведомление об успешном удалении заказа.</li> </ol>
<b>Альтернативные пути</b>	<ol style="list-style-type: none"> <li>2а. Пользователь отказывается от выполнения операции. <ol style="list-style-type: none"> <li>2а.1. Система завершает выполнение операции.</li> </ol> </li> <li>4а. Пользователь не даёт подтверждение. <ol style="list-style-type: none"> <li>4а.1. Работа возобновляется с шага 1.</li> </ol> </li> </ol>
<b>Результат</b>	Заказ удалён из списка заказов.

Таблица 1.14 - Вариант использования «Обновить статус заказа»

<b>UC-14. Обновить статус заказа</b>	
<b>Определение / Ценность</b>	Операция предназначена для изменения информации о статусе заказа в связи с его закупкой у поставщика или прибытием.
<b>Пользователь</b>	Администратор, продавец, ответственный за закупки
<b>Предусловие</b>	Пользователь авторизирован и инициировал выполнение операции
<b>Основной сценарий</b>	<ol style="list-style-type: none"> <li>1. Система отображает список заказов.</li> <li>2. Пользователь выбирает заказ для обновления статуса.</li> <li>3. Система предоставляет форму для заполнения новых данных о статусе заказа.</li> <li>4. Пользователь заполняет форму и подтверждает обновление статуса заказа.</li> <li>5. Система отображает уведомление об успешном обновлении статуса заказа.</li> </ol>
<b>Альтернативные пути</b>	<ol style="list-style-type: none"> <li>2а. Пользователь отказывается от выполнения операции. <ol style="list-style-type: none"> <li>2а.1. Система завершает выполнение операции.</li> </ol> </li> <li>4а. Пользователь отказывается от выполнения операции. <ol style="list-style-type: none"> <li>4а.1. Система завершает выполнение операции.</li> </ol> </li> <li>4б. Пользователь неверно заполняет форму. <ol style="list-style-type: none"> <li>4б.1. Система отображает уведомление об ошибке.</li> <li>4б.2. Работа возобновляется с шага 3.</li> </ol> </li> </ol>
<b>Результат</b>	Статус заказа обновлён.

Таблица 1.15 - Вариант использования «Обновить статус оплаты заказа»

<b>UC-15. Обновить статус оплаты заказа</b>	
<b>Определение / Ценность</b>	Операция предназначена для изменения информации о статусе оплаты заказа в связи с поступлением оплаты от клиента.
<b>Пользователь</b>	Продавец
<b>Предусловие</b>	Пользователь авторизирован и инициировал выполнение операции
<b>Основной сценарий</b>	<ol style="list-style-type: none"> <li>1. Система отображает список заказов.</li> <li>2. Пользователь выбирает заказ для обновления статуса оплаты.</li> <li>3. Система запрашивает подтверждение обновления статуса оплаты заказа.</li> <li>4. Пользователь подтверждает обновление статуса оплаты заказа.</li> </ol>

Продолжение таблицы 1.15

	5. Система отображает уведомление об успешном обновлении статуса оплаты заказа.
<b>Альтернативные пути</b>	2а. Пользователь отказывается от выполнения операции. 2а.1. Система завершает выполнение операции. 4а. Пользователь отказывается от выполнения операции. 4а.1. Система завершает выполнение операции.
<b>Результат</b>	Статус оплаты заказа обновлён.

Таблица 1.16 - Вариант использования «Добавить модель спецтехники»

<b>UC-16. Добавить модель спецтехники</b>	
<b>Определение / Ценность</b>	Операция предназначена для добавления новой модели спецтехники в список доступной в ассортименте спецтехники.
<b>Пользователь</b>	Ответственный за закупки
<b>Предусловие</b>	Пользователь авторизирован и инициировал выполнение операции
<b>Основной сценарий</b>	1. Система предоставляет форму для заполнения данных о модели спецтехники. 2. Пользователь заполняет форму и подтверждает добавление модели спецтехники. 3. Система отображает уведомление об успешном добавлении модели спецтехники.
<b>Альтернативные пути</b>	2а. Пользователь отказывается от выполнения операции. 2а.1. Система завершает выполнение операции. 2б. Пользователь неверно заполняет форму. 2б.1. Система отображает уведомление об ошибке. 2б.2. Работа возобновляется с шага 1.
<b>Результат</b>	Модель спецтехники добавлена в ассортимент.

Таблица 1.17 - Вариант использования «Редактировать данные о модели спецтехники»

<b>UC-17. Редактировать данные о модели спецтехники</b>	
<b>Определение / Ценность</b>	Операция предназначена для изменения данных о модели спецтехники в связи с ошибкой при её создании.

Продолжение таблицы 1.17

<b>Пользователь</b>	Ответственный за закупки
<b>Предусловие</b>	Пользователь авторизирован и инициировал выполнение операции
<b>Основной сценарий</b>	<p>1. Система отображает список доступной в ассортименте спецтехники.</p> <p>2. Пользователь выбирает модель спецтехники для редактирования.</p> <p>3. Система предоставляет форму для заполнения новых данных.</p> <p>4. Пользователь заполняет форму и подтверждает редактирование.</p> <p>5. Система отображает уведомление об успешном редактировании модели спецтехники.</p>
<b>Альтернативные пути</b>	<p>2а. Пользователь отказывается от выполнения операции.</p> <p>2а.1. Система завершает выполнение операции.</p> <p>4а. Пользователь отказывается от выполнения операции.</p> <p>4а.1. Система завершает выполнение операции.</p> <p>4б. Пользователь неверно заполняет форму.</p> <p>4б.1. Система отображает уведомление об ошибке.</p> <p>4б.2. Работа возобновляется с шага 3.</p>
<b>Результат</b>	Данные о модели спецтехники отредактированы.

Таблица 1.18 - Вариант использования «Удалить модель спецтехники»

<b>UC-18. Удалить модель спецтехники</b>	
<b>Определение / Ценность</b>	Операция предназначена для удаления модели спецтехники из списка доступной в ассортименте спецтехники для предотвращения оформления заказа на эту модель.
<b>Пользователь</b>	Ответственный за закупки
<b>Предусловие</b>	Пользователь авторизирован и инициировал выполнение операции
<b>Основной сценарий</b>	<p>1. Система отображает список доступной в ассортименте спецтехники.</p> <p>2. Пользователь выбирает модель спецтехники для удаления.</p> <p>3. Система запрашивает подтверждение удаления.</p> <p>4. Пользователь подтверждает удаление.</p> <p>5. Система отображает уведомление об успешном удалении модели спецтехники.</p>

Продолжение таблицы 1.18

<b>Альтернативные пути</b>	<p>2а. Пользователь отказывается от выполнения операции.</p> <p>2а.1. Система завершает выполнение операции.</p> <p>4а. Пользователь не даёт подтверждение.</p> <p>4а.1. Работа возобновляется с шага 1.</p>
<b>Результат</b>	Модель спецтехники отмечена как удалённая.

Таблица 1.19 - Вариант использования «Восстановить модель спецтехники»

<b>УС-19. Восстановить модель спецтехники</b>	
<b>Определение / Ценность</b>	Операция предназначена для восстановления модели спецтехники в списке доступной в ассортименте спецтехники для возобновления возможности оформления заказа на эту модель.
<b>Пользователь</b>	Ответственный за закупки
<b>Предусловие</b>	Пользователь авторизирован и инициировал выполнение операции
<b>Основной сценарий</b>	<p>1. Система отображает список удалённой спецтехники.</p> <p>2. Пользователь выбирает модель спецтехники для восстановления.</p> <p>3. Система запрашивает подтверждение восстановления.</p> <p>4. Пользователь подтверждает восстановление.</p> <p>5. Система отображает уведомление об успешном восстановлении модели спецтехники.</p>
<b>Альтернативные пути</b>	<p>2а. Пользователь отказывается от выполнения операции.</p> <p>2а.1. Система завершает выполнение операции.</p> <p>4а. Пользователь не даёт подтверждение.</p> <p>4а.1. Работа возобновляется с шага 1.</p>
<b>Результат</b>	Модель спецтехники восстановлена.

Таблица 1.20 - Вариант использования «Добавить поставщика»

<b>УС-20. Добавить поставщика</b>	
<b>Определение / Ценность</b>	Операция предназначена для добавления нового поставщика в список доступных поставщиков.
<b>Пользователь</b>	Ответственный за закупки
<b>Предусловие</b>	Пользователь авторизирован и инициировал выполнение операции

Продолжение таблицы 1.20

<b>Основной сценарий</b>	<p>1. Система предоставляет форму для заполнения данных о поставщике.</p> <p>2. Пользователь заполняет форму и подтверждает добавление поставщика.</p> <p>3. Система отображает уведомление об успешном добавлении поставщика.</p>
<b>Альтернативные пути</b>	<p>2а. Пользователь отказывается от выполнения операции.</p> <p>2а.1. Система завершает выполнение операции.</p> <p>б. Пользователь неверно заполняет форму.</p> <p>2б.1. Система отображает уведомление об ошибке.</p> <p>2б.2. Работа возобновляется с шага 1.</p>
<b>Результат</b>	Поставщик добавлен.

Таблица 1.21 - Вариант использования «Редактировать данные о поставщике»

<b>UC-21. Редактировать данные о поставщике</b>	
<b>Определение / Ценность</b>	Операция предназначена для изменения данных о поставщике в связи с ошибкой при его создании или обновлении данных.
<b>Пользователь</b>	Ответственный за закупки
<b>Предусловие</b>	Пользователь авторизован и инициировал выполнение операции
<b>Основной сценарий</b>	<p>1. Система отображает список поставщиков.</p> <p>2. Пользователь выбирает поставщика для редактирования.</p> <p>3. Система предоставляет форму для заполнения новых данных.</p> <p>4. Пользователь заполняет форму и подтверждает редактирование.</p> <p>5. Система отображает уведомление об успешном редактировании поставщика.</p>
<b>Альтернативные пути</b>	<p>2а. Пользователь отказывается от выполнения операции.</p> <p>2а.1. Система завершает выполнение операции.</p> <p>4а. Пользователь отказывается от выполнения операции.</p> <p>4а.1. Система завершает выполнение операции.</p> <p>4б. Пользователь неверно заполняет форму.</p> <p>4б.1. Система отображает уведомление об ошибке.</p> <p>4б.2. Работа возобновляется с шага 3.</p>

Продолжение таблицы 1.21

<b>Результат</b>	Данные о поставщике отредактированы.
------------------	--------------------------------------

Таблица 1.22 - Вариант использования «Удалить поставщика»

<b>UC-22. Удалить поставщика</b>	
<b>Определение / Ценность</b>	Операция предназначена для удаления поставщика из списка доступных поставщиков для предотвращения добавления спецтехники от этого поставщика.
<b>Пользователь</b>	Ответственный за закупки
<b>Предусловие</b>	Пользователь авторизирован и инициировал выполнение операции
<b>Основной сценарий</b>	<ol style="list-style-type: none"> <li>1. Система отображает список поставщиков.</li> <li>2. Пользователь выбирает поставщика для удаления.</li> <li>3. Система запрашивает подтверждение удаления.</li> <li>4. Пользователь подтверждает удаление.</li> <li>5. Система отображает уведомление об успешном удалении поставщика.</li> </ol>
<b>Альтернативные пути</b>	<ol style="list-style-type: none"> <li>2а. Пользователь отказывается от выполнения операции. <ol style="list-style-type: none"> <li>2а.1. Система завершает выполнение операции.</li> </ol> </li> <li>4а. Пользователь не даёт подтверждение. <ol style="list-style-type: none"> <li>4а.1. Работа возобновляется с шага 1.</li> </ol> </li> </ol>
<b>Результат</b>	Поставщик отмечен как удалённый.

Таблица 1.23 - Вариант использования «Восстановить поставщика»

<b>UC-23. Восстановить поставщика</b>	
<b>Определение / Ценность</b>	Операция предназначена для восстановления поставщика в списке поставщиков для добавления спецтехники от этого поставщика.
<b>Пользователь</b>	Ответственный за закупки
<b>Предусловие</b>	Пользователь авторизирован и инициировал выполнение операции
<b>Основной сценарий</b>	<ol style="list-style-type: none"> <li>1. Система отображает список удалённых поставщиков.</li> <li>2. Пользователь выбирает поставщика для восстановления.</li> <li>3. Система запрашивает подтверждение восстановления.</li> <li>4. Пользователь подтверждает восстановление.</li> <li>5. Система отображает уведомление об успешном восстановлении поставщика.</li> </ol>



Продолжение таблицы 1.23

<b>Альтернативные пути</b>	2а. Пользователь отказывается от выполнения операции. 2а.1. Система завершает выполнение операции. 4а. Пользователь не даёт подтверждение. 4а.1. Работа возобновляется с шага 1.
<b>Результат</b>	Поставщик восстановлен.

Таблица 1.24 - Вариант использования «Редактировать данные клиента»

<b>UC-24. Редактировать данные клиента</b>	
<b>Определение / Ценность</b>	Операция предназначена для изменения данных о клиенте в связи с ошибкой при его добавлении или их обновлении.
<b>Пользователь</b>	Ответственный за закупки
<b>Предусловие</b>	Пользователь авторизован и инициировал выполнение операции
<b>Основной сценарий</b>	1. Система отображает список клиентов. 2. Пользователь выбирает клиента для редактирования. 3. Система предоставляет форму для заполнения новых данных. 4. Пользователь заполняет форму и подтверждает редактирование. 5. Система отображает уведомление об успешном редактировании клиента.
<b>Альтернативные пути</b>	2а. Пользователь отказывается от выполнения операции. 2а.1. Система завершает выполнение операции. 4а. Пользователь отказывается от выполнения операции. 4а.1. Система завершает выполнение операции. 4б. Пользователь неверно заполняет форму. 4б.1. Система отображает уведомление об ошибке. 4б.2. Работа возобновляется с шага 3.
<b>Результат</b>	Данные о клиенте отредактированы.

## 1.5 Разработка функциональных требований

На основе пользовательских требований был сформирован список функциональных требований [7]. В таблицах 1.25-1.48 приведены функциональные требования к системе.

Таблица 1.25 - Требования к функции «Авторизоваться»

Идентификатор	Функции/требования
<b>F1</b>	Авторизоваться
<b>FR-1.1</b>	Система должна предоставить форму для заполнения данных учётной записи.
<b>FR-1.2</b>	Система должна выдавать уведомление при успешной авторизации.
<b>FR-1.3</b>	Система должна выдавать ошибку в случае, если форма заполнена неверно.

Таблица 1.26 - Требования к функции «Создать учётную запись пользователя»

Идентификатор	Функции/требования
<b>F2</b>	Создать учётную запись пользователя
<b>FR-2.1</b>	Система должна предоставить пользователю форму для заполнения данных будущей учётной записи.
<b>FR-2.2</b>	Система должна выдавать уведомление при успешном создании учётной записи.
<b>FR-2.3</b>	Система должна выдавать ошибку в случае, если форма заполнена неверно.

Таблица 1.27 - Требования к функции «Посмотреть заказы конкретного сотрудника»

Идентификатор	Функции/требования
<b>F3</b>	Посмотреть заказы конкретного сотрудника
<b>FR-3.1</b>	Система должна предоставить пользователю список сотрудников для выбора получения нужных заказов.
<b>FR-3.2</b>	Система должна отобразить список заказов выбранного сотрудника.

Таблица 1.28 - Требования к функции «Редактировать учётную запись пользователя»

Идентификатор	Функции/требования
<b>F4</b>	Редактировать учётную запись пользователя
<b>FR-4.1</b>	Система должна предоставить пользователю список созданных учётных записей для выбора редактируемой записи.
<b>FR-4.2</b>	Система должна предоставить форму для заполнения новых данных учётной записи.
<b>FR-4.3</b>	Система должна выдавать уведомление при успешном редактировании учётной записи.
<b>FR-4.4</b>	Система должна выдавать ошибку в случае, если форма заполнена неверно.

Таблица 1.29 - Требования к функции «Удалить учётную запись пользователя»

Идентификатор	Функции/требования
<b>F5</b>	Удалить учётную запись пользователя
<b>FR-5.1</b>	Система должна предоставить пользователю список созданных учётных записей для выбора удаляемой записи.
<b>FR-5.2</b>	Система должна запрашивать у пользователя подтверждение удаления записи.
<b>FR-5.3</b>	Система должна выдавать уведомление при успешном удалении учётной записи.

Таблица 1.30 - Требования к функции «Восстановление учётной записи пользователя»

Идентификатор	Функции/требования
<b>F6</b>	Восстановление учётной записи пользователя
<b>FR-6.1</b>	Система должна предоставить пользователю список удалённых учётных записей для выбора восстанавливаемой записи.

Продолжение таблицы 1.30

<b>FR-6.2</b>	Система должна запрашивать у пользователя подтверждение восстановления записи.
<b>FR-6.3</b>	Система должна выдавать уведомление при успешном восстановлении учётной записи.

Таблица 1.31 - Требования к функции «Посмотреть информацию о конкретном заказе»

<b>Идентификатор</b>	<b>Функции/требования</b>
<b>F7</b>	Посмотреть информацию о конкретном заказе
<b>FR-7.1</b>	Система должна предоставить пользователю список заказов для выбора просматриваемого заказа.
<b>FR-7.2</b>	Система должна отобразить информацию о выбранном заказе.

Таблица 1.32 - Требования к функции «Посмотреть заказы конкретного клиента»

<b>Идентификатор</b>	<b>Функции/требования</b>
<b>F8</b>	Посмотреть заказы конкретного клиента
<b>FR-8.1</b>	Система должна предоставить пользователю список клиентов.
<b>FR-8.2</b>	Система должна отобразить список заказов выбранного клиента.

Таблица 1.33 - Требования к функции «Посмотреть заказы на конкретную модель спецтехники»

<b>Идентификатор</b>	<b>Функции/требования</b>
<b>F9</b>	Посмотреть заказы на конкретную модель спецтехники
<b>FR-9.1</b>	Система должна предоставить пользователю список спецтехники.
<b>FR-9.2</b>	Система должна отобразить список заказов на выбранную модель спецтехники.

Таблица 1.34 - Требования к функции «Оформить заказ»

Идентификатор	Функции/требования
<b>F10</b>	Оформить заказ
<b>FR-10.1</b>	Система должна предоставить пользователю форму для заполнения данных о заказе.
<b>FR-10.2</b>	Система должна выдавать уведомление при успешном оформлении заказа.
<b>FR-10.3</b>	Система должна выдавать ошибку в случае, если форма заполнена неверно.

Таблица 1.35 - Требования к функции «Закрыть заказ»

Идентификатор	Функции/требования
<b>F11</b>	Закрыть заказ
<b>FR-11.1</b>	Система должна предоставить пользователю список заказов для выбора закрываемого заказа.
<b>FR-11.2</b>	Система должна запрашивать у пользователя подтверждение закрытия заказа.
<b>FR-11.3</b>	Система должна выдавать уведомление при успешном закрытии заказа.

Таблица 1.36 - Требования к функции «Отменить заказ»

Идентификатор	Функции/требования
<b>F12</b>	Отменить заказ
<b>FR-12.1</b>	Система должна предоставить пользователю список заказов для выбора отменяемого заказа.
<b>FR-12.2</b>	Система должна запрашивать у пользователя подтверждение отмены заказа.
<b>FR-12.3</b>	Система должна выдавать уведомление при успешной отмене заказа.

Таблица 1.37 - Требования к функции «Удалить заказ»

Идентификатор	Функции/требования
<b>F13</b>	Удалить заказ

Продолжение таблицы 1.37

<b>FR-13.1</b>	Система должна предоставить пользователю список заказов для выбора удаляемого заказа.
<b>FR-13.2</b>	Система должна запрашивать у пользователя подтверждение удаления заказа.
<b>FR-13.3</b>	Система должна выдавать уведомление при успешном удалении заказа.

Таблица 1.38 - Требования к функции «Обновить статус заказа»

<b>Идентификатор</b>	<b>Функции/требования</b>
<b>F14</b>	Обновить статус заказа
<b>FR-14.1</b>	Система должна предоставить пользователю список не закрытых заказов для выбора обновляемого заказа.
<b>FR-14.2</b>	Система должна предоставить форму для заполнения новых данных о статусе заказа.
<b>FR-14.3</b>	Система должна выдавать уведомление при успешном обновлении статуса заказа.
<b>FR-14.4</b>	Система должна выдавать ошибку в случае, если форма заполнена неверно.

Таблица 1.39 - Требования к функции «Обновить статус оплаты заказа»

<b>Идентификатор</b>	<b>Функции/требования</b>
<b>F15</b>	Обновить статус оплаты заказа
<b>FR-15.1</b>	Система должна предоставить пользователю список заказов для выбора обновляемого заказа.
<b>FR-15.2</b>	Система должна запрашивать подтверждение обновления статуса оплаты заказа.
<b>FR-15.3</b>	Система должна выдавать уведомление при успешном обновлении статуса оплаты заказа.

Таблица 1.40 - Требования к функции «Добавить модель спецтехники»

<b>Идентификатор</b>	<b>Функции/требования</b>
<b>F16</b>	Добавить модель спецтехники

Продолжение таблицы 1.40

<b>FR-16.1</b>	Система должна предоставить пользователю форму для заполнения данных о модели спецтехники.
<b>FR-16.2</b>	Система должна выдавать уведомление при успешном добавлении модели спецтехники.
<b>FR-16.3</b>	Система должна выдавать ошибку в случае, если форма заполнена неверно.

Таблица 1.41 - Требования к функции «Редактировать информацию о модели спецтехники»

<b>Идентификатор</b>	<b>Функции/требования</b>
<b>F17</b>	Редактировать информацию о модели спецтехники
<b>FR-17.1</b>	Система должна предоставить пользователю список доступных в ассортименте моделей спецтехники для выбора редактируемой модели.
<b>FR-17.2</b>	Система должна предоставить форму для заполнения новых данных о модели спецтехники.
<b>FR-17.3</b>	Система должна выдавать уведомление при успешном редактировании модели спецтехники.
<b>FR-17.4</b>	Система должна выдавать ошибку в случае, если форма заполнена неверно.

Таблица 1.42 - Требования к функции «Удалить модель спецтехники»

<b>Идентификатор</b>	<b>Функции/требования</b>
<b>F18</b>	Удалить модель спецтехники
<b>FR-18.1</b>	Система должна предоставить пользователю список доступных в ассортименте моделей спецтехники для выбора удаляемой модели.
<b>FR-18.2</b>	Система должна запрашивать у пользователя подтверждение удаления модели спецтехники.
<b>FR-18.3</b>	Система должна выдавать уведомление при успешном удалении модели спецтехники.

Таблица 1.43 - Требования к функции «Восстановить модель спецтехники»

Идентификатор	Функции/требования
<b>F19</b>	Восстановить модель спецтехники
<b>FR-19.1</b>	Система должна предоставить пользователю список удалённых моделей спецтехники для выбора восстанавливаемой модели.
<b>FR-19.2</b>	Система должна запрашивать у пользователя подтверждение восстановления модели спецтехники.
<b>FR-19.3</b>	Система должна выдавать уведомление при успешном восстановлении модели спецтехники.

Таблица 1.44 - Требования к функции «Добавить поставщика»

Идентификатор	Функции/требования
<b>F20</b>	Добавить поставщика
<b>FR-20.1</b>	Система должна предоставить пользователю форму для заполнения данных о поставщике.
<b>FR-20.2</b>	Система должна выдавать уведомление при успешном добавлении поставщика.
<b>FR-20.3</b>	Система должна выдавать ошибку в случае, если форма заполнена неверно.

Таблица 1.45 - Требования к функции «Редактировать данные о поставщике»

Идентификатор	Функции/требования
<b>F21</b>	Редактировать данные о поставщике
<b>FR-21.1</b>	Система должна предоставить пользователю список доступных поставщиков для выбора редактируемого поставщика.
<b>FR-21.2</b>	Система должна предоставить форму для заполнения новых данных о поставщике.
<b>FR-21.3</b>	Система должна выдавать уведомление при успешном редактировании поставщика.
<b>FR-21.4</b>	Система должна выдавать ошибку в случае, если форма заполнена неверно.



Таблица 1.46 - Требования к функции «Удалить поставщика»

Идентификатор	Функции/требования
<b>F22</b>	Удалить поставщика
<b>FR-22.1</b>	Система должна предоставить пользователю список доступных поставщиков для выбора удаляемого поставщика.
<b>FR-22.2</b>	Система должна запрашивать у пользователя подтверждение удаления поставщика.
<b>FR-22.3</b>	Система должна выдавать уведомление при успешном удалении поставщика.

Таблица 1.47 - Требования к функции «Восстановить поставщика»

Идентификатор	Функции/требования
<b>F23</b>	Восстановить модель спецтехники
<b>FR-23.1</b>	Система должна предоставить пользователю список удалённых поставщиков для выбора восстанавливаемого поставщика.
<b>FR-23.2</b>	Система должна запрашивать у пользователя подтверждение восстановления поставщика.
<b>FR-23.3</b>	Система должна выдавать уведомление при успешном восстановлении поставщика.

Таблица 1.48 - Требования к функции «Редактировать данные о поставщике»

Идентификатор	Функции/требования
<b>F24</b>	Редактировать данные о клиенте
<b>FR-24.1</b>	Система должна предоставить пользователю список клиентов для выбора редактируемого клиента.
<b>FR-24.2</b>	Система должна предоставить форму для заполнения новых данных о клиенте.
<b>FR-24.3</b>	Система должна выдавать уведомление при успешном редактировании клиента.
<b>FR-24.4</b>	Система должна выдавать ошибку в случае, если форма заполнена неверно.

## 1.6 Разработка нефункциональных требований

В ходе анализа были сформированы нефункциональные требования к системе [8].

В таблице 1.49 приведены требования к пользовательским интерфейсам.

Таблица 1.49 - Требования к пользовательским интерфейсам

<b>UI-1</b>	Пользовательский интерфейс должен быть простым и интуитивно понятным.
<b>UI-2</b>	При вызове пользователем какой-либо функции в заголовке нового окна должно быть указано название операции.

В таблице 1.50 приведены требования к производительности.

Таблица 1.50 - Требования к производительности

<b>PER-1</b>	Все веб-страницы, генерируемые системой, должны полностью загружаться не более чем за 4 секунды после запроса их по интернет-подключению со скоростью 20 Мбит/сек.
--------------	--

В таблице 1.51 приведены требования к безопасности.

Таблица 1.51 - Требования к безопасности

<b>SEC-1</b>	Вся личная информация должна передаваться в базу данных по защищённым каналам связи.
<b>SEC-2</b>	Пользователи для выполнения всех операций системы обязательно должны быть авторизованными в системе.

В таблице 1.52 приведены требования к надёжности.

Таблица 1.52 - Требования к надёжности

<b>REL-1</b>	Не более 0,1% данных могут быть потеряны из-за сбоев ПО
<b>REL-2</b>	При возникновении ошибки обработки данных должно быть сформировано уведомление для администратора системы

В таблице 1.53 приведены требования, исходящие из бизнес-правил.

Таблица 1.53 - Требования, исходящие из бизнес-правил

<b>BR-1</b>	Закрытие заказа должно быть доступно только для заказов, статус оплаты которых соответствует полной оплате.
<b>BR-2</b>	Отмена и закрытие заказа, а также изменение статуса заказа и оплаты должны быть доступны только для не закрытых заказов.

### 1.7 Вывод по главе

В первой главе ВКР был проведен анализ требований на разработку. В ходе анализа бизнес-требований были сформированы бизнес-цели, критерии успеха, бизнес-риски, видение решения, была сформирована концептуальная карта, созданы две ER-модели. Далее после анализа предметной области была разработана BPMN-модель и описаны пользовательские, функциональные, нефункциональные требования.

## **2. Проектирование информационной системы**

### **2.1 Функциональная модель системы**

На основе требований были выявлены основные функции информационной системы. Для этих функций были определены входные и выходные данные.

Были выявлены следующие функции системы:

- авторизация;
- оформление заказа;
- обновление статуса заказа;
- обновление статуса оплаты заказа;
- закрытие заказа;
- отмена заказа;
- формирование списка заказов;
- добавление моделей спецтехники, пользователей, поставщиков, клиентов;
- редактирование данных моделей спецтехники, пользователей, поставщиков, клиентов;
- удаление моделей спецтехники, пользователей, поставщиков, заказов;
- восстановление моделей спецтехники, пользователей, поставщиков.

Входными данными являются:

- данные о спецтехнике;
- данные поставщика;
- данные клиента;
- данные пользователя;
- запрос клиента;
- актуальные данные заказа.

Выходными данными являются:

- списки заказов;
- данные о конкретном заказе.

На рисунке 2.1 представлена функциональная модель системы [9].



Рисунок 2.1 – Функциональная модель системы

## 2.2 Разработка алгоритмов системы

Для того, чтобы более подробно рассмотреть функции системы ниже приведены их алгоритмы работы.

### 2.2.1 Алгоритм авторизации

Для функции авторизации разработан алгоритм, приведённый на рисунке 2.2 [10].

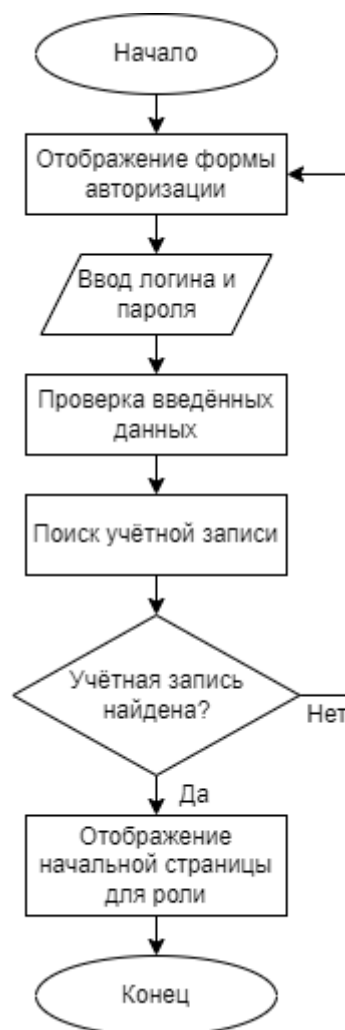


Рисунок 2.2 – Алгоритм авторизации

Процесс авторизации начинается с отображения формы ввода логина и пароля. После ввода данных и подтверждения входа, происходит проверка введенных данных на корректность и поиск пользователя в базе данных. В

случае успешного нахождения пользователя, отображается начальная страница для роли, в которой авторизован пользователь. Если пользователь не найден, пользователь перенаправляется назад на форму ввода логина и пароля.

### 2.2.2 Добавление данных

Для функций оформления заказа, а также добавления клиента, модели спецтехники, поставщика и пользователя разработан алгоритм, приведённый на рисунке 2.3 [11].



Рисунок 2.3 – Алгоритм добавления данных

Добавление моделей спецтехники, пользователей, поставщиков, клиентов, а также оформление заказа происходит по схожему алгоритму, с различием в формах для ввода данных.

Процесс добавления данных начинается с отображения формы ввода данных. После ввода данных и подтверждения добавления, происходит проверка введенных данных на корректность. В случае успешной проверки, данные формируются в соответствующий объект, который затем добавляется в базу данных.

Затем происходит сохранение изменений в базе данных и перенаправление пользователя на страницу, с которой было инициировано действие. Если данные введены некорректно, пользователь перенаправляется назад на форму ввода данных.

### **2.2.3 Редактирование данных**

Редактирование данных спецтехники, пользователей, поставщиков, клиентов, отмена и закрытие заказа, а также обновление статуса оплаты заказа и статуса заказа происходят по схожему алгоритму, с различием в формах для ввода данных.

Процесс редактирования данных начинается с отображения списка объектов требуемого вида. Пользователь выбирает редактируемый объект, после чего происходит отображение формы ввода новых данных. После ввода данных и подтверждения редактирования, происходит проверка введенных данных на корректность. В случае успешной проверки, данные формируются в соответствующий объект, на основе которого происходит обновление уже существующего объекта в базе данных.

Затем происходит сохранение изменений в базе данных и перенаправление пользователя на страницу, с которой было инициировано действие. Если данные введены некорректно, пользователь перенаправляется назад на форму ввода данных.



На рисунке 2.4 приведён алгоритм редактирования данных [12].



Рисунок 2.4 – Алгоритм редактирования данных

#### 2.2.4 Удаление данных

Для функций удаления спецтехники, пользователя, клиентов и поставщиков разработан алгоритм, приведённый на рисунке 2.5.

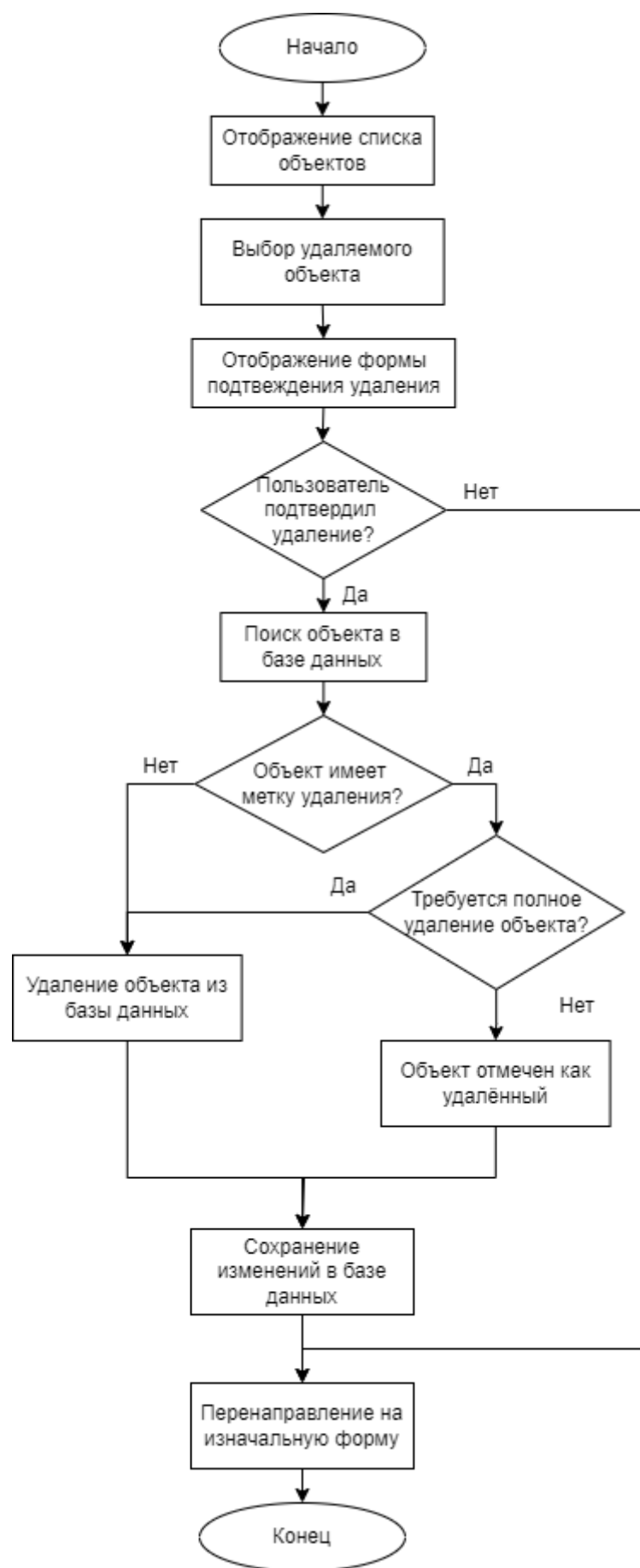


Рисунок 2.5 – Алгоритм удаления данных

Удаление моделей спецтехники, пользователей, поставщиков и клиентов происходит по схожему алгоритму в зависимости от наличия у объекта поля для отметки его как удалённого и необходимости в его полном удалении.

Процесс удаления данных начинается с отображения списка объектов требуемого вида. Пользователь выбирает удаляемый объект, после чего происходит отображение формы подтверждения удаления. В случае подтверждения пользователем удаления, происходит получение соответствующего удаления из базы данных.

Если объект можно пометить как удалённый и его полное удаление не требуется, то он будет помечен как удалённый. Если такой возможности нет или требуется полное удаление, то объект будет удалён из базы данных.

Затем происходит сохранение изменений в базе данных и перенаправление пользователя на страницу, с которой было инициировано действие. Если подтверждение на удаление не было дано, пользователь перенаправляется на страницу, с которой было инициировано действие, сразу.

### **2.2.5 Восстановление данных**

Восстановление моделей спецтехники, пользователей и поставщиков происходит по схожему алгоритму.

Процесс восстановления данных начинается с отображения списка удалённых объектов требуемого вида. Пользователь выбирает восстанавливаемый объект, после чего происходит отображение формы подтверждения восстановления. В случае подтверждения пользователем восстановления, происходит получение соответствующего объекта из базы данных. Полученный объект помечается как не удалённый, после чего происходит обновление соответствующего объекта в базе данных.

Затем происходит сохранение изменений в базе данных и перенаправление пользователя на страницу, с которой было инициировано

действие. Если подтверждение на восстановление не было дано, пользователь перенаправляется на страницу, с которой было инициировано действие, сразу.

На рисунке 2.6 приведён алгоритм восстановления данных.



Рисунок 2.6 – Алгоритм восстановления данных

## 2.3 Проектирование базы данных

Была спроектирована база данных, содержащая 9 таблиц: «Orders», «Employees», «Clients», «Machineries», «Suppliers», «Roles», «OrderStatuses», «PaymentStatuses» и «MachineryClasses» [13]. Схема взаимосвязей таблиц представлена на рисунке 2.7.

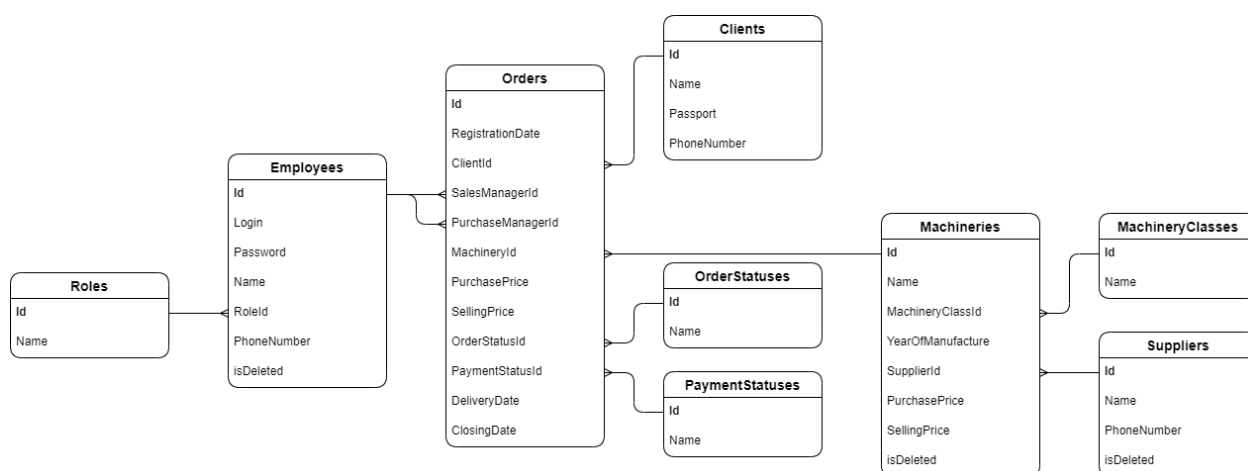


Рисунок 2.7 – Схема взаимосвязей таблиц базы данных

Рассмотрим каждую таблицу подробнее.

Таблица «Orders» хранит информацию о заказах. Наименования её полей и их тип приведены в таблице 2.1.

Таблица 2.1 – Наименования и типы полей таблицы «Orders»

№	Имя поля	Тип поля
1	Id	Int
2	RegistrationDate	Date
3	ClientId	Int
4	SalesManagerId	Int
5	PurchaseManagerId	Int
6	MachineryId	Int
7	PurchasePrice	Double
8	SellingPrice	Double
9	PaymentStatusId	Int

Продолжение таблицы 2.1

<b>10</b>	OrderStatusId	Int
<b>11</b>	DeliveryDate	Date
<b>12</b>	ClosingDate	Date

Таблица «Employees» хранит данные сотрудников, в том числе логин и пароль для авторизации в системе, а также должность, определяющую доступный сотруднику функционал. Наименования её полей и их тип приведены в таблице 2.2.

Таблица 2.2 – Наименования и типы полей таблицы «Employees»

<b>№</b>	<b>Имя поля</b>	<b>Тип поля</b>
<b>1</b>	Id	Int
<b>2</b>	Login	String
<b>3</b>	Password	String
<b>4</b>	Name	String
<b>5</b>	RoleId	Int
<b>6</b>	PhoneNumber	String
<b>7</b>	isDeleted	Bool

Таблица «Clients» хранит данные клиентов. Наименования её полей и их тип приведены в таблице 2.3.

Таблица 2.3 – Наименования и типы полей таблицы «Clients»

<b>№</b>	<b>Имя поля</b>	<b>Тип поля</b>
<b>1</b>	Id	Int
<b>2</b>	Name	String
<b>3</b>	Passport	String
<b>4</b>	PhoneNumber	String

Таблица «Machineries» хранит данные о представленных в ассортименте моделях спецтехники. Наименования её полей и их тип приведены в таблице 2.4.

Таблица 2.4 – Наименования и типы полей таблицы «Machineries»

№	Имя поля	Тип поля
1	Id	Int
2	Name	String
3	MachineryClassId	Int
4	YearOfManufacture	Int
5	SupplierId	Int
6	PurchasePrice	Double
7	SellingPrice	Double
8	isDeleted	Bool

Таблица «Suppliers» хранит данные о поставщиках спецтехники. Наименования её полей и их тип приведены в таблице 2.5.

Таблица 2.5 – Наименования и типы полей таблицы «Suppliers»

№	Имя поля	Тип поля
1	Id	Int
2	Name	String
3	PhoneNumber	String
4	isDeleted	Bool

Таблица «Roles» хранит данные о полях пользователей в системе. Наименования её полей и их тип приведены в таблице 2.6.

Таблица 2.6 – Наименования и типы полей таблицы «Roles»

№	Имя поля	Тип поля
1	Id	Int
2	Name	String

Таблица «MachineryClasses» хранит данные о классах спецтехники, доступных при добавлении моделей спецтехники. Наименования её полей и их тип приведены в таблице 2.7.

Таблица 2.7 – Наименования и типы полей таблицы «MachineryClasses»

№	Имя поля	Тип поля
1	Id	Int
2	Name	String

Таблица «OrderStatuses» хранит данные о возможных состояниях заказа. Наименования её полей и их тип приведены в таблице 2.8.

Таблица 2.8 – Наименования и типы полей таблицы «Roles»

№	Имя поля	Тип поля
1	Id	Int
2	Name	String

Таблица «PaymentStatuses» хранит данные о возможных статусах оплаты заказа. Наименования её полей и их тип приведены в таблице 2.9.

Таблица 2.9 – Наименования и типы полей таблицы «PaymentStatuses»

№	Имя поля	Тип поля
1	Id	Int
2	Name	String

## 2.4 Вывод по главе

В второй главе ВКР была спроектирована информационная система. В ходе анализа были выявлены функции данной информационной системы, необходимые входные и получаемые выходные данные, была разработана функциональная модель системы. Также были разработаны алгоритмы системы и спроектирована база данных.



### **3. Реализация информационной системы**

#### **3.1 Реализация архитектуры информационной системы**

Построение качественной архитектуры является одним из факторов, определяющих как сложность разработки, так и качество итогового программного продукта. Подобная архитектура должна соответствовать следующим критериям:

- Гибкость системы;
- Масштабируемость системы;
- Эффективность системы;
- Тестируемость системы.

Для удовлетворения данных критериев при разработке будет использован шаблон MVC (Mode-View-Controller).

Шаблон подразумевает разделение приложения на 3 части:

- Model – определяет логику хранения и управления данными;
- View – определяет логику визуализации результатов работы программы;
- Controller – обрабатывает пользовательские запросы и входящие данные.

Части имеют чёткое разделение, что упрощает разработку, тестирование и поддержку независимо от размеров приложения [14].

#### **3.2 Интерфейс информационной системы**

##### **3.2.1 Реализация страницы авторизации**

Сотрудник начинает взаимодействие с системой со страницы авторизации (рисунок 3.1).

ООО "ГрандТех"

**Вход на сайт**

Введите логин

admin

Введите пароль

\*\*\*\*\*

Войти

Рисунок 3.1 – Страница авторизации

В системе отсутствует возможность самостоятельной регистрации пользователей, так как это может создать угрозу безопасности системы. Поэтому созданием и управлением пользователями занимается администратор системы. Также администратору доступен весь описанный далее функционал страниц.

После авторизации пользователь попадает на начальную страницу, определённую для каждой роли.

### 3.2.2 Реализация страницы «Оформление/выдача заказов»

После авторизации продавец попадает на страницу «Оформление/выдача заказов» (рисунок 3.2). На этой странице отображаются заказы авторизованного пользователя, находящиеся в компетенции продавца.

Заказы поделены на 3 группы: ожидающие предоплату, ожидающие полную оплату и готовые к выдаче.

При нажатии кнопки «Оформить заказ» открывается страница для ввода данных нового заказа (рисунок 3.3). При оформлении заказа указывается требуемая модель спецтехники из фильтруемого списка, затем выбираются продавец и ответственный за закупку, назначенные на заказ, а также вводятся данные клиента.

При обновлении статуса оплаты, удалении или закрытии заказа открывается страница для подтверждения данного действия (рисунок 3.4).

ООО "ГрандТех"

Оформление/выдача заказов

Закупка спецтехники

Клиенты

Ассортимент спецтехники

Поставщики

Выход

Оформить заказ

Активные заказы

Ожидающие предоплату

Id	Клиент	Продавец	Ответственный за закупку	Состав заказа	Дата оформления	Действия
4	Лялюшкин Ростислав Кузьмич	Скворцов Андрей Николаевич	Романов Сергей Викторович	Беларус 82.1	03.06.2022 11:55:28	<a href="#">Подробнее</a> <a href="#">Обновить статус оплаты</a> <a href="#">Удалить заказ</a>

Ожидающие полную оплату

Id	Клиент	Продавец	Ответственный за закупку	Состав заказа	Дата оформления	Действия
1	Остапов Константин Сергеевич	Скворцов Андрей Николаевич	Романов Сергей Викторович	Беларус 82.1	01.06.2022 14:52:18	<a href="#">Подробнее</a> <a href="#">Обновить статус оплаты</a>

Готовые к выдаче

Id	Клиент	Продавец	Ответственный за закупку	Состав заказа	Дата оформления	Действия
3	Койтасбаев Венедикт Леонтьевич	Скворцов Андрей Николаевич	Романов Сергей Викторович	SDLG E6210F	03.06.2022 11:53:51	<a href="#">Подробнее</a> <a href="#">Заккрыть заказ</a>

Рисунок 3.2 – Страница «Оформление/выдача заказов»

ООО "ГрандТех"

Оформление/выдача заказов

Закупка спецтехники

Клиенты

Ассортимент спецтехники

Поставщики

Выход

Состав заказа

Поставщик: MTZ-Tatarstan

Класс спецтехники: Трактор

Фильтр

Модель спецтехники

Беларус 82.1

Ответственный персонал

Продавец

Скворцов Андрей Николаевич

Ответственный за закупку

Романов Сергей Викторович

Данные клиента

Имя клиента

Остапов Константин Сергеевич

Паспортные данные

9201857439

Номер телефона

+79038520538

Добавить заказ

Рисунок 3.3 – Страница оформления заказа

ООО "ГрандТех"

Оформление/выдача заказов

Закупка спецтехники

Клиенты

Ассортимент спецтехники

Поставщики

Выход

Подтвердите закрытие заказа.

Подтвердить

Рисунок 3.4 – Страница подтверждения действия на примере закрытия заказа

При нажатии кнопки «Подробнее» напротив заказа открывается страница с детальной информацией этого заказа (рисунки 3.5, 3.6).

ООО "ГрандТех"	Оформление/выдача заказов	Закупка спецтехники	Клиенты	Ассортимент спецтехники	Поставщики	Выход
----------------	---------------------------	---------------------	---------	-------------------------	------------	-------

### Заказ №4

#### Основное

**Продавец**  
Скворцов Андрей Николаевич

**Ответственный за закупку**  
Романов Сергей Викторович

**Дата оформления**  
03.06.2022 11:55:28

**Статус заказа**  
ПроверкаДоступностиЗакупки

**Статус оплаты**  
ОжиданиеПредоплаты

Рисунок 3.5 – Страница с детальной информацией о заказе

### Состав заказа

**Модель спецтехники**  
Беларус 82.1

**Год выпуска**  
2008

**Класс спецтехники**  
Трактор

**Цена закупки (руб.)**  
1500000

**Цена продажи (руб.)**  
1700000

### Данные поставщика

**Поставщик**  
MTZ-Tatarstan

**Контактный телефон**  
+79057405734

### Данные клиента

**Имя клиента**  
Лялюшкин Ростислав Кузьмич

**Паспортные данные**  
4613465554

**Контактный телефон**  
+79105243351

© 2022 - ООО "ГрандТех"

Рисунок 3.6 – Продолжение страницы с детальной информацией о заказе

Функционал страницы «Оформление/выдача заказов» предназначен главным образом для работы пользователей в роли продавца. При открытии страницы пользователем иной роли, доступный на ней функционал будет ограничен (рисунок 3.7).

ООО "ГрандТех"   Оформление/выдача заказов <b>Закупка спецтехники</b> Клиенты   Ассортимент спецтехники   Поставщики   Выход						
<b>Активные заказы</b>						
<b>Ожидающие предоплату</b>						
Id	Клиент	Продавец	Ответственный за закупку	Состав заказа	Дата оформления	Действия
4	Лялюшкин Ростислав Кузьмич	Скворцов Андрей Николаевич	Романов Сергей Викторович	Беларус 82.1	03.06.2022 11:55:28	<a href="#">Подробнее</a>
<b>Ожидающие полную оплату</b>						
Id	Клиент	Продавец	Ответственный за закупку	Состав заказа	Дата оформления	Действия
1	Остапов Константин Сергеевич	Скворцов Андрей Николаевич	Романов Сергей Викторович	Беларус 82.1	01.06.2022 14:52:18	<a href="#">Подробнее</a>
<b>Готовые к выдаче</b>						
Id	Клиент	Продавец	Ответственный за закупку	Состав заказа	Дата оформления	Действия
3	Койтасбаев Венедикт Леонтьевич	Скворцов Андрей Николаевич	Романов Сергей Викторович	SDLG E6210F	03.06.2022 11:53:51	<a href="#">Подробнее</a>

Рисунок 3.7 – Страница «Оформление/выдача заказов» с ограниченным функционалом

### 3.2.3 Реализация страницы «Закупка спецтехники»

После авторизации ответственный за закупку попадает на страницу «Закупка спецтехники» (рисунок 3.8). На этой странице отображаются заказы авторизованного продавца, находящиеся на том этапе выполнения, который требует действий ответственного за закупку.

ООО "ГрандТех"   Оформление/выдача заказов <b>Закупка спецтехники</b> Клиенты   Ассортимент спецтехники   Поставщики   Выход						
<b>Активные заказы</b>						
<b>Ожидают подтверждения</b>						
Id	Клиент	Продавец	Состав заказа	Дата оформления	Действия	
2	Рощин Даниил Семенович	Скворцов Андрей Николаевич	SDLG E6210F	03.06.2022 11:50:09	<a href="#">Подробнее</a>	<a href="#">Подтвердить закупку</a> <a href="#">Отменить</a>
<b>Ожидают прибытия</b>						
Id	Клиент	Продавец	Состав заказа	Дата оформления	Действия	
5	Остапов Константин Сергеевич	Скворцов Андрей Николаевич	SDLG E6210F	03.06.2022 11:56:53	<a href="#">Подробнее</a>	<a href="#">Подтвердить прибытие</a>

Рисунок 3.8 – Страница «Закупка спецтехники»

Процесс подтверждения отмены и прибытия заказа аналогичен подтверждениям действий на странице «Оформление/выдача заказов», а также просмотр подробной информации о заказе. Для подтверждения закупки дополнительно требуется ввод даты ожидаемой поставки (рисунок 3.9).

Рисунок 3.9 – Страница подтверждения закупки

Функционал страницы «Закупка спецтехники» предназначен главным образом для работы пользователей в роли ответственного за закупку. При открытии страницы пользователем иной роли, доступный на ней функционал будет ограничен (рисунок 3.10).

Активные заказы					
Ожидают подтверждения					
Id	Клиент	Продавец	Состав заказа	Дата оформления	Действия
2	Рощин Даниил Семенович	Скворцов Андрей Николаевич	SDLG E6210F	03.06.2022 11:50:09	<a href="#">Подробнее</a>

Ожидают прибытия					
Id	Клиент	Продавец	Состав заказа	Дата оформления	Действия
5	Остапов Константин Сегреевич	Скворцов Андрей Николаевич	SDLG E6210F	03.06.2022 11:56:53	<a href="#">Подробнее</a>

Рисунок 3.10 – Страница «Закупка спецтехники» с ограниченным функционалом

### 3.2.4 Реализация страницы «Клиенты»

Страница «Клиенты» отображает список клиентов дилерского центра (рисунок 3.11).

<div> <div>ООО "ГрандТех"</div> <div> Оформление/выдача заказов Закупка спецтехники Клиенты Ассортимент спецтехники Поставщики Выход </div> </div>					
<div> Добавить клиента Клиенты </div>					
Имя	Паспортные данные	Телефон	Действия		
Горшков Семён Андреевич	9204638529	+79063359537		Редактировать	Удалить
Остапов Константин Сегреевич	9201857439	+79038520538	История заказов	Редактировать	
Рощин Даниил Семенович	4359694085	+79296239476	История заказов	Редактировать	
Койтасбаев Венедикт Леонтьевич	4368751437	+79334943574	История заказов	Редактировать	
Лялюшкин Ростислав Кузьмич	4613465554	+79105243351	История заказов	Редактировать	

Рисунок 3.11 – Страница «Клиенты»

Добавление клиента происходит либо при оформлении заказа, либо при нажатии кнопки «Добавить клиента», открывающей страницу для заполнения данных нового клиента (рисунок 3.12).

<div> <div>ООО "ГрандТех"</div> <div> Оформление/выдача заказов Закупка спецтехники Клиенты Ассортимент спецтехники Поставщики Выход </div> </div>					
Имя клиента					
<input type="text"/>					
Паспортные данные					
<input type="text"/>					
Номер телефона					
<input type="text"/>					
Добавить клиента					

Рисунок 3.12 – Страница для заполнения данных нового клиента

Редактирование данных клиента происходит после нажатия напротив нужного клиента кнопки «Редактировать», открывающей страницу для заполнения новых данных клиента (рисунок 3.13).

<div> <div>ООО "ГрандТех"</div> <div> Оформление/выдача заказов Закупка спецтехники Клиенты Ассортимент спецтехники Поставщики Выход </div> </div>					
Имя клиента					
<input type="text" value="Горшков Семён Андреевич"/>					
Паспортные данные					
<input type="text" value="9204638529"/>					
Номер телефона					
<input type="text" value="+79063359537"/>					
Сохранить изменения					

Рисунок 3.13 – Страница для редактирования данных клиента

После нажатия кнопки «История заказов» напротив клиента, открывается страница со списком заказов этого клиента, поделённых на активные, закрытые и отменённые (рисунок 3.14).

ООО "ГрандТех"   Оформление/выдача заказов   Закупка спецтехники   Клиенты   Ассортимент спецтехники   Поставщики   Выход							
<b>Заказы клиента Остапов Константин Сегреевич</b>							
<b>Активные заказы</b>							
Id	Продавец	Ответственный за закупку	Состав заказа	Дата оформления	Дата поставки	Статус заказа	Статус оплаты
1	Скворцов Андрей Николаевич	Романов Сергей Викторович	Беларус 82.1	01.06.2022 14:52:18	03.06.2022 11:58:28	ГотовКПередаче	ОжиданиеПолнойОплаты
5	Скворцов Андрей Николаевич	Романов Сергей Викторович	SDLG E6210F	03.06.2022 11:56:53	03.06.2022 11:58:00	ОжиданиеДоставки	ОжиданиеПолнойОплаты
<b>Закрытые заказы</b>							
Id	Продавец	Ответственный за закупку	Состав заказа	Дата оформления	Дата поставки	Дата закрытия	
<b>Отменённые заказы</b>							
Id	Продавец	Ответственный за закупку	Состав заказа				

Рисунок 3.14 – Страница «История заказов»

Процесс подтверждения удаления клиента аналогичен процессам подтверждения действий на прошлых страницах.

Функционал добавления, редактирования и удаления предназначен главным образом для работы пользователей в роли продавца. При открытии страницы пользователем иной роли, доступный на ней функционал будет ограничен (рисунок 3.15).

ООО "ГрандТех"   Оформление/выдача заказов   Закупка спецтехники   Клиенты   Ассортимент спецтехники   Поставщики   Выход			
<b>Клиенты</b>			
Имя	Паспортные данные	Телефон	Действия
Горшков Семён Андреевич	9204638529	+79063359537	
Остапов Константин Сегреевич	9201857439	+79038520538	<a href="#">История заказов</a>
Рошин Даниил Семенович	4359694085	+79296239476	<a href="#">История заказов</a>
Койтасбаев Венедикт Леонтьевич	4368751437	+79334943574	<a href="#">История заказов</a>
Лялюшкин Ростислав Кузьмич	4613465554	+79105243351	<a href="#">История заказов</a>

Рисунок 3.15 – Страница «Клиенты» с ограниченным функционалом

### 3.2.5 Реализация страницы «Ассортимент спецтехники»

Страница «Ассортимент спецтехники» отображает списки как доступной в ассортименте, так и недоступной спецтехники (рисунок 3.16). Для



списков имеется возможность отобразить модели спецтехники определённого класса или от конкретного поставщика [15].

Добавить модель спецтехники

**Модели спецтехники**

Поставщик:  Класс спецтехники:  [Фильтр](#)

**Доступные модели**

Наименование	Класс	Год выпуска	Поставщик	Цена закупки (руб.)	Цена продажи (руб.)	Действия
Беларус 82.1	Трактор	2008	MTZ-Tatarstan	1500000	1700000	<a href="#">История заказов</a> <a href="#">Редактировать</a> <a href="#">Удалить</a>
Caterpillar D4	Бульдозер	2011	DEUTZ	2500000	2750000	<a href="#">История заказов</a> <a href="#">Редактировать</a> <a href="#">Удалить</a>
SDLG E6210F	Экскаватор	2010	DEUTZ	9500000	10000000	<a href="#">История заказов</a> <a href="#">Редактировать</a> <a href="#">Удалить</a>

**Удалённые модели**

Наименование	Класс	Год выпуска	Поставщик	Цена закупки (руб.)	Цена продажи (руб.)	Действия
SOLIS-90 4WD	Трактор	2014	Solis	2290000	2500000	<a href="#">История заказов</a> <a href="#">Восстановить</a>

Рисунок 3.16 – Страница «Ассортимент спецтехники»

Добавление новой модели происходит при нажатии кнопки «Добавить модель спецтехники», открывающей страницу для заполнения данных о новой модели спецтехники (рисунок 3.17).

ООО "ГрандТех" [Оформление/выдача заказов](#) [Закупка спецтехники](#) [Клиенты](#) [Ассортимент спецтехники](#) [Поставщики](#) [Выход](#)

Наименование

Класс спецтехники

Поставщик

Год выпуска

Стоимость закупки (руб.)

Стоимость продажи (руб.)

[Добавить модель спецтехники](#)

Рисунок 3.17 – Страница для заполнения данных о новой модели спецтехники

Редактирование данных о модели спецтехники происходит после нажатия напротив нужной модели кнопки «Редактировать», открывающей страницу для заполнения новых данных о модели спецтехники (рисунок 3.18).

ООО "ГрандТех"   Оформление/выдача заказов   Закупка спецтехники   Клиенты   Ассортимент спецтехники   Поставщики   Выход

---

Наименование

Класс спецтехники

Поставщик

Год выпуска

Стоимость закупки (руб.)

Стоимость продажи (руб.)

[Сохранить изменения](#)

Рисунок 3.18 – Страница для редактирования данных о модели спецтехники

Процесс подтверждения удаления и восстановления модели спецтехники аналогичен процессам подтверждения действий на прошлых страницах.

Функционал добавления, редактирования, удаления и восстановления моделей спецтехники предназначен главным образом для работы пользователей в роли ответственного за закупку. При открытии страницы пользователем иной роли, доступный на ней функционал будет ограничен (рисунок 3.19).

ООО "ГрандТех"   Оформление/выдача заказов   Закупка спецтехники   Клиенты   Ассортимент спецтехники   Поставщики   Выход

---

**Модели спецтехники**

Поставщик:  Класс спецтехники:  [Фильтр](#)

**Доступные модели**

Наименование	Класс	Год выпуска	Поставщик	Цена закупки (руб.)	Цена продажи (руб.)	Действия
Беларус 82.1	Трактор	2008	MTZ-Tatarstan	1500000	1700000	<a href="#">История заказов</a>
Caterpillar D4	Бульдозер	2011	DEUTZ	2500000	2750000	<a href="#">История заказов</a>
SDLG E6210F	Экскаватор	2010	DEUTZ	9500000	10000000	<a href="#">История заказов</a>

**Удалённые модели**

Наименование	Класс	Год выпуска	Поставщик	Цена закупки (руб.)	Цена продажи (руб.)	Действия
SOLIS-90 4WD	Трактор	2014	Solis	2290000	2500000	<a href="#">История заказов</a>

Рисунок 3.19 – Страница «Ассортимент спецтехники» с ограниченным функционалом

Процесс просмотра истории заказов на конкретную модель спецтехники аналогичен процессу просмотра истории заказов конкретного клиента.

### 3.2.6 Реализация страницы «Поставщики»

Страница «Поставщики» отображает списки как доступных, так и недоступных поставщиков (рисунок 3.20).

ООО "ГрандТех"

Оформление/выдача заказов

Закупка спецтехники

Клиенты

Ассортимент спецтехники

Поставщики

Выход

Добавить поставщика

Поставщики

Действующие поставщики

Имя	Телефон	Действия		
MTZ-Tatarstan	+79057405734	<a href="#">Подробнее</a>	<a href="#">Редактировать</a>	<a href="#">Удалить</a>
DEUTZ	+79306066918	<a href="#">Подробнее</a>	<a href="#">Редактировать</a>	<a href="#">Удалить</a>

Удалённые поставщики

Имя	Телефон	Действия	
Solis	+79573194283	<a href="#">Подробнее</a>	<a href="#">Восстановить</a>

Рисунок 3.20 – Страница «Поставщики»

Добавление нового поставщика происходит при нажатии кнопки «Добавить поставщика», открывающей страницу для заполнения данных о новом поставщике (рисунок 3.21).

ООО "ГрандТех"   Оформление/выдача заказов   Закупка спецтехники   Клиенты   Ассортимент спецтехники   Поставщики   Выход					
Имя поставщика					
<input type="text"/>					
Номер телефона					
<input type="text"/>					
<a href="#">Добавить поставщика</a>					

Рисунок 3.21 – Страница для заполнения данных о новом поставщике

Редактирование данных о поставщике происходит после нажатия напротив нужного поставщика кнопки «Редактировать», открывающей страницу для заполнения новых данных о поставщике (рисунок 3.22).

ООО "ГрандТех"
Оформление/выдача заказов
Закупка спецтехники
Клиенты
Ассортимент спецтехники
Поставщики
Выход

Имя поставщика

MTZ-Tatarstan

Номер телефона

+79057405734

Сохранить изменения

Рисунок 3.22 – Страница для редактирования данных о поставщике

Процесс подтверждения удаления и восстановления поставщика аналогичен процессам подтверждения действий на прошлых страницах.

После нажатия кнопки «Подробнее» напротив поставщика, открывается страница со списком спецтехники от этого поставщика, поделённой на доступные и недоступные в ассортименте (рисунок 3.23).

ООО "ГрандТех"
Оформление/выдача заказов
Закупка спецтехники
Клиенты
Ассортимент спецтехники
Поставщики
Выход

### Модели спецтехники от DEUTZ

#### Доступные модели

Наименование	Класс	Год выпуска	Цена закупки (руб.)	Цена продажи (руб.)	Действия
Caterpillar D4	Бульдозер	2011	2500000	2750000	<a href="#">Подробнее</a>
SDLG E6210F	Экскаватор	2010	9500000	10000000	<a href="#">Подробнее</a>

#### Удалённые модели

Наименование	Класс	Год выпуска	Цена закупки (руб.)	Цена продажи (руб.)	Действия
--------------	-------	-------------	---------------------	---------------------	----------

Рисунок 3.23 – Страница со списком спецтехники от выбранного поставщика

Функционал добавления, редактирования, удаления и восстановления поставщиков предназначен главным образом для работы пользователей в роли ответственного за закупку. При открытии страницы пользователем иной роли, доступный на ней функционал будет ограничен (рисунок 3.24).

ООО "ГрандТех"    Оформление/выдача заказов    Закупка спецтехники    Клиенты    Ассортимент спецтехники    Поставщики    Выход		
<b>Поставщики</b>		
<b>Действующие поставщики</b>		
Имя	Телефон	Действия
MTZ-Tatarstan	+79057405734	<a href="#">Подробнее</a>
DEUTZ	+79306066918	<a href="#">Подробнее</a>
<b>Удалённые поставщики</b>		
Имя	Телефон	Действия
Solis	+79573194283	<a href="#">Подробнее</a>

Рисунок 3.24 – Страница «Поставщики» с ограниченным функционалом

### 3.2.7 Реализация страницы «Сотрудники»

После авторизации администратор попадает на страницу «Сотрудники» (рисунок 3.25). На этой странице отображаются списки активных и удалённых пользователей.

"ГрандТех"    Сотрудники    Оформление/выдача заказов    Закупка спецтехники    Клиенты    Ассортимент спецтехники    Поставщики    Выход						
<a href="#">Добавить пользователя</a>						
<b>Список сотрудников</b>						
<b>Действующие сотрудники</b>						
Имя	Телефон	Должность	Логин	Пароль	Действия	
Администратор	+79306066918	Администратор	admin	admin	<a href="#">Редактировать</a>	
Скворцов Андрей Николаевич	+79874624729	Продавец	seller	seller	<a href="#">Редактировать</a>	<a href="#">История заказов</a> <a href="#">Удалить</a>
Романов Сергей Викторович	+79056839657	ОтветственныйЗаЗакупку	purchaser	purchaser	<a href="#">Редактировать</a>	<a href="#">История заказов</a> <a href="#">Удалить</a>
<b>Удалённые сотрудники</b>						
Имя	Телефон	Должность	Логин	Пароль	Действия	
Анисимов Николай Михайлович	+79535892051	Продавец	hiretofire	hiretofire	<a href="#">История заказов</a> <a href="#">Восстановить</a>	

Рисунок 3.25 – Страница «Сотрудники»

Добавление нового пользователя происходит при нажатии кнопки «Добавить пользователя», открывающей страницу для заполнения данных о новом пользователе (рисунок 3.26).

The screenshot shows a web interface for 'GrandTech' with a navigation bar at the top containing links: 'ГрандТех', 'Сотрудники', 'Оформление/выдача заказов', 'Закупка спецтехники', 'Клиенты', 'Ассортимент спецтехники', 'Поставщики', and 'Выход'. Below the navigation bar is a registration form with the following fields: 'Логин' (Login), 'Пароль' (Password), 'Имя пользователя' (User Name), 'Должность' (Job Title) with a dropdown menu currently showing 'Продавец' (Salesman), and 'Номер телефона' (Phone Number). At the bottom of the form is a button labeled 'Добавить пользователя' (Add user).

Рисунок 3.26 – Страница для заполнения данных о новом пользователе

Редактирование данных о пользователе происходит после нажатия напротив нужного пользователя кнопки «Редактировать», открывающей страницу для заполнения новых данных о пользователе (рисунок 3.27).

The screenshot shows the same 'GrandTech' web interface, but the registration form is now an editing form. The fields are pre-filled with the following data: 'Логин' (seller), 'Пароль' (seller), 'Имя пользователя' (Скворцов Андрей Николаевич), 'Должность' (Продавец), and 'Номер телефона' (+79874624729). At the bottom of the form is a button labeled 'Сохранить изменения' (Save changes).

Рисунок 3.27 – Страница для редактирования данных о пользователе

Процесс подтверждения удаления и восстановления пользователей аналогичен процессам подтверждения действий на прошлых страницах.

Процесс просмотра истории заказов конкретного пользователя аналогичен процессам просмотра истории заказов на прошлых страницах.

В отличие от прошлых страниц, страница «Сотрудники» предназначена только для администратора, так как на ней представлены данные и функционал, позволяющие стороннему человеку получить доступ к системе, что может представлять угрозу для безопасности данных.

### **3.3 Вывод по главе**

В третьей главе ВКР была разработана информационная система. Была описана реализация архитектура системы. Была описана реализация страницы авторизации, а также страниц «Заказы продавца», «Закупка заказов», «Клиенты», «Спецтехника», «Поставщики» и «Сотрудники».

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения ВКР была разработана информационная система для учёта заказов спецтехники.

В первой главе был проведён анализ бизнес-требований. Было выявлено, что дилерский центр спецтехники «ГрандТех» испытывает трудности с обслуживанием клиентов из-за плохо согласованной работы отдела по продажам и отдела по управлению закупками. Продавцам приходится звонить в отдел по управлению закупками, чтобы узнать о статусе заказа. В свою очередь ответственному за закупки приходится лично отвечать на каждую заявку. Было принято решение разработать информационную систему для упрощения взаимодействия между отделами и для снижения среднего времени обслуживания клиентов. Были сформированы бизнес-требования, разработаны модели данных и BPMN2 модель бизнес процессов. В итоге были разработаны пользовательские, функциональные и нефункциональные требования к системе.

Во второй главе была спроектирована информационная система. На основании выявленных функций, а также входных и выходных данных системы была разработана функциональная модель системы. Были разработаны алгоритмы авторизации, а также добавления, редактирования, удаления и восстановления данных. Также была спроектирована база данных.

В третьей главе была разработана информационная система. Была описана реализация архитектура системы. Была описана реализация страницы авторизации, а также страниц «Заказы продавца», «Закупка заказов», «Клиенты», «Спецтехника», «Поставщики» и «Сотрудники».

Таким образом при выполнении работы были решены все поставленные задачи, работа выполнена в полном объёме и в заданный срок.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Вигерс К., Битти Д. Разработка требований к программному обеспечению. 3-е изд., дополненное / Пер. с англ. — М.: Издательство «Русская редакция»; СПб. : БХВ-Петербург, 2014. — 736 стр.:ил.
- 2 Концептуальные карты или концепт-карты [Электронный ресурс] // NitForYou.com: [сайт] URL: <https://nitforyou.com/konceptualnye-karty/> (дата обращения 20.05.2022)
- 3 Нотации модели сущность-связь (ER диаграммы) [Электронный ресурс] // Блог Программиста: [сайт] URL: <https://pro-prof.com/archives/8126> (дата обращения: 21.05.2022)
- 4 Хамадеев Ш.А. Методология моделирования бизнес-процессов BPMN2. Учебно-методическое пособие по дисциплине «Проектирование АСОИУ». – Набережные Челны: ИПЦ НЧИ К(П)ФУ, 2017. – 36 с.
- 5 Арлоу Д., Нейштадт И. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование, 2е издание. – Пер. с англ. – СПб: СимволПлюс, 2007. – 624 с.:ил.
- 6 Хамадеев Ш.А. Методология описания пользовательских требований. Учебно-методическое пособие по дисциплине «Проектирование АСОИУ». – Набережные Челны: ИПЦ НЧИ К(П)ФУ, 2017. – 28 с.
- 7 Коберн А. Современные методы описания функциональных требований к системе / Пер. с англ. – Москва: Издательство «Лори», 2002.
- 8 О нефункциональных требованиях [Электронный ресурс] // SchoolForAnalyst: [сайт] URL: <https://schoolforanalyst.ru/aboutnonfuncreq> (дата обращения: 22.05.2022)
- 9 Создание диаграмм [Электронный ресурс] // Draw.io: [сайт] URL: <https://app.diagrams.net/> (дата обращения: 23.05.2022)
- 10 Авторизация по ролям [Электронный ресурс] // Metanit.com: [сайт] URL: <https://metanit.com/sharp/aspnet5/15.5.php> (дата обращения 24.05.2022)

11 Основные операции с моделями. Создание и вывод [Электронный ресурс] // Metanit.com: [сайт] URL: <https://metanit.com/sharp/aspnet5/12.2.php> (дата обращения: 25.05.2022)

12 Основные операции с моделями. Редактирование и удаление [Электронный ресурс] // Metanit.com: [сайт] URL: <https://metanit.com/sharp/aspnet5/12.3.php> (дата обращения: 26.05.2022)

13 Введение в проектирование баз данных [Электронный ресурс] // База знаний по бизнес-анализу: [сайт] URL: <https://analytics.infozone.pro/vvedenie-v-proektirovanie-baz-dannih/> (дата обращения: 27.05.2022)

14 Фримен, А. ASP.NET MVC 3 для профессионалов /А. Фримен, С. Сандерсон. - Москва: Вильямс, 2012. - 52 с.

15 Фильтрация [Электронный ресурс] // Metanit.com: [сайт] URL: <https://metanit.com/sharp/aspnet5/12.6.php> (дата обращения: 28.05.2022)

## ПРИЛОЖЕНИЕ А

### Код контроллеров на языке C#

Код контроллера AccountController:

```
using System.Collections.Generic;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using DealerCenter.Models;
using Microsoft.EntityFrameworkCore;
using System.Security.Claims;
using Microsoft.AspNetCore.Authentication;
using Microsoft.AspNetCore.Authentication.Cookies;

namespace ClaimsApp.Controllers
{
    public class AccountController : Controller
    {
        private DealerCenterContext db;
        public AccountController(DealerCenterContext context)
        {
            db = context;
        }

        [HttpGet]
        public IActionResult Login()
        {
            if (User.IsInRole(RoleEnum.Администратор.ToString()))
            {
                return RedirectToAction("Employees", "Employee");
            }
            if (User.IsInRole(RoleEnum.Продавец.ToString()))
            {
                return RedirectToAction("SalesManagerOrders", "SalesManagerOrder");
            }
            if (User.IsInRole(RoleEnum.ОтветственныйЗаЗакупку.ToString()))
            {
                return RedirectToAction("PurchaseManagerOrders", "PurchaseManagerOrder");
            }
            return View();
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<IActionResult> Login(LoginModel model)
        {
            if (ModelState.IsValid)
            {
                Employee user = await db.Employees.Include(u => u.Role)
                    .FirstOrDefaultAsync(e => e.Login == model.Login && e.Password ==
model.Password);
                if (user != null && !user.isDeleted)
                {
                    await Authenticate(user);

                    if (user.Role.Name == RoleEnum.Администратор.ToString())
                    {
                        return RedirectToAction("Employees", "Employee");
                    }
                    if (user.Role.Name == RoleEnum.ОтветственныйЗаЗакупку.ToString())
                    {

```

```

        return RedirectToAction("PurchaseManagerOrders",
"PurchaseManagerOrder");
    }
    if (user.Role.Name == RoleEnum.Продавец.ToString())
    {
        return RedirectToAction("SalesManagerOrders",
"SalesManagerOrder");
    }
    ModelState.AddModelError("", "Некорректные логин и(или) пароль");
}
return View();
}

public IActionResult Logout()
{
    HttpContext.SignOutAsync(CookieAuthenticationDefaults.AuthenticationScheme);
    return RedirectToAction("Login", "Account");
}

private async Task Authenticate(Employee user)
{
    List<Claim> claims = new List<Claim>
    {
        new Claim(ClaimsIdentity.DefaultNameClaimType, user.Login),
        new Claim(ClaimsIdentity.DefaultRoleClaimType, user.Role.Name.ToString())
    };
    ClaimsIdentity claimIdentity = new ClaimsIdentity(claims,
"ApplicationCookie", ClaimsIdentity.DefaultNameClaimType,
ClaimsIdentity.DefaultRoleClaimType);
    await
HttpContext.SignInAsync(CookieAuthenticationDefaults.AuthenticationScheme, new
ClaimsPrincipal(claimIdentity));
}
}
}

```

### Код контроллера ClientController:

```

using DealerCenter.Models;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;

namespace DealerCenter.Controllers
{
    public class ClientController : Controller
    {
        DealerCenterContext db;

        public ClientController(DealerCenterContext context)
        {
            db = context;
        }

        public IActionResult Clients()
        {
            IQueryable<Client> clients = db.Clients.Include(c => c.Orders);
            return View(clients.ToList());
        }
    }
}

```

```

[Authorize(Roles = "Администратор, Продавец")]
public IActionResult Create()
{
    return View();
}

[HttpPost]
public async Task<IActionResult> Create(Client client)
{
    db.Clients.Add(client);
    await db.SaveChangesAsync();
    return RedirectToAction("Clients");
}

[Authorize(Roles = "Администратор, Продавец")]
public async Task<IActionResult> Edit(int? id)
{
    if (id != null)
    {
        Client client = await db.Clients.FirstOrDefaultAsync(p => p.Id == id);
        if (client != null)
        {
            return View(client);
        }
    }
    return NotFound();
}

[HttpPost]
public async Task<IActionResult> Edit(Client client)
{
    db.Clients.Update(client);
    await db.SaveChangesAsync();
    return RedirectToAction("Clients");
}

[Authorize(Roles = "Администратор, Продавец")]
[HttpGet]
[ActionName("Delete")]
public async Task<IActionResult> ConfirmDelete(int? id)
{
    if (id != null)
    {
        Client client = await db.Clients.FirstOrDefaultAsync(p => p.Id == id);
        if (client != null)
        {
            return View(client);
        }
    }
    return NotFound();
}

[HttpPost]
public async Task<IActionResult> Delete(int? id)
{
    if (id != null)
    {
        Client client = await db.Clients.FirstOrDefaultAsync(p => p.Id == id);
        if (client != null)
        {
            db.Clients.Remove(client);
            await db.SaveChangesAsync();
            return RedirectToAction("Clients");
        }
    }
}

```

```

        return NotFound();
    }

    public async Task<IActionResult> Details(int? id)
    {
        if (id != null)
        {
            List<Order> orders = await db.Orders.Include(o => o.Client)
                                                .Include(o => o.SalesManager)
                                                .Include(o => o.PurchaseManager)
                                                .Include(o => o.OrderStatus)
                                                .Include(o => o.PaymentStatus)
                                                .Include(o => o.Machinery)
                                                .Include(o => o.Machinery.MachineryClass)
                                                .Include(o => o.Machinery.Supplier)
                                                .Where(o => o.ClientId == id)
                                                .ToListAsync();

            if (orders != null)
                return View(orders);
        }
        return NotFound();
    }

    [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore =
true)]
    public IActionResult Error()
    {
        return View(new ErrorViewModel { RequestId = Activity.Current?.Id ??
HttpContext.TraceIdentifier });
    }
}

```

### Код контроллера EmployeeController:

```

using DealerCenter.Models;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;

namespace DealerCenter.Controllers
{
    public class EmployeeController : Controller
    {
        DealerCenterContext db;

        public EmployeeController(DealerCenterContext context)
        {
            db = context;
        }

        [Authorize(Roles = "Администратор")]
        public IActionResult Employees()
        {
            IQueryable<Employee> employees = db.Employees.Include(e => e.Role);
            return View(employees.ToList());
        }
    }
}

```

```

[Authorize(Roles = "Администратор")]
public IActionResult Create()
{
    ViewBag.Roles = new SelectList(db.Roles.Where(r => r.Name !=
RoleEnum.Администратор.ToString()), "Id", "Name");
    return View();
}

[HttpPost]
public async Task<IActionResult> Create(Employee employee)
{
    db.Employees.Add(employee);
    await db.SaveChangesAsync();
    return RedirectToAction("Employees");
}

[Authorize(Roles = "Администратор")]
public async Task<IActionResult> Edit(int? id)
{
    if (id != null)
    {
        Employee employee = await db.Employees.Include(e =>
e.Role).FirstOrDefaultAsync(p => p.Id == id);
        if (employee != null)
        {
            if (employee.Role.Name == RoleEnum.Администратор.ToString())
            {
                ViewBag.Roles = new SelectList(db.Roles.Where(r => r.Name ==
RoleEnum.Администратор.ToString()), "Id", "Name");
            }
            else
            {
                ViewBag.Roles = new SelectList(db.Roles.Where(r => r.Name !=
RoleEnum.Администратор.ToString()), "Id", "Name");
            }
            return View(employee);
        }
    }
    return NotFound();
}

[HttpPost]
public async Task<IActionResult> Edit(Employee employee)
{
    db.Employees.Update(employee);
    await db.SaveChangesAsync();
    return RedirectToAction("Employees");
}

[Authorize(Roles = "Администратор")]
[HttpGet]
[ActionName("FlagDelete")]
public async Task<IActionResult> ConfirmFlagDelete(int? id)
{
    if (id != null)
    {
        Employee employee = await db.Employees.FirstOrDefaultAsync(p => p.Id ==
id);
        if (employee != null)
            return View(employee);
    }
    return NotFound();
}

```

```

[HttpPost]
public async Task<IActionResult> FlagDelete(int? id)
{
    if (id != null)
    {
        Employee employee = await db.Employees.FirstOrDefaultAsync(p => p.Id ==
id);

        if (employee != null)
        {
            employee.isDeleted = true;
            db.Employees.Update(employee);
            await db.SaveChangesAsync();
            return RedirectToAction("Employees");
        }
    }
    return NotFound();
}

[Authorize(Roles = "Администратор")]
[HttpGet]
[ActionName("Restore")]
public async Task<IActionResult> ConfirmRestore(int? id)
{
    if (id != null)
    {
        Employee employee = await db.Employees.FirstOrDefaultAsync(p => p.Id ==
id);

        if (employee != null)
            return View(employee);
    }
    return NotFound();
}

[HttpPost]
public async Task<IActionResult> Restore(int? id)
{
    if (id != null)
    {
        Employee employee = await db.Employees.FirstOrDefaultAsync(p => p.Id ==
id);

        if (employee != null)
        {
            employee.isDeleted = false;
            db.Employees.Update(employee);
            await db.SaveChangesAsync();
            return RedirectToAction("Employees");
        }
    }
    return NotFound();
}

public async Task<IActionResult> Details(int? id)
{
    if (id != null)
    {
        List<Order> orders = await db.Orders.Include(o => o.Client)
                                             .Include(o => o.SalesManager)
                                             .Include(o => o.PurchaseManager)
                                             .Include(o => o.OrderStatus)
                                             .Include(o => o.PaymentStatus)
                                             .Include(o => o.Machinery)
                                             .Include(o =>
o.Machinery.MachineryClass)
                                             .Include(o => o.Machinery.Supplier)

```



```

        o.PurchaseManagerId == id)
        .Where(o => o.SalesManagerId == id ||
        .ToListAsync();

        if (orders != null)
            return View(orders);
    }
    return NotFound();
}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore =
true)]
public IActionResult Error()
{
    return View(new ErrorViewModel { RequestId = Activity.Current?.Id ??
HttpContext.TraceIdentifier });
}
}
}

```

### Код контроллера MachineryController:

```

using DealerCenter.Models;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;

namespace DealerCenter.Controllers
{
    public class MachineryController : Controller
    {
        DealerCenterContext db;

        public MachineryController(DealerCenterContext context)
        {
            db = context;
        }

        public IActionResult Machineries(int? machineryClassId, int? supplierId)
        {
            IQueryable<Machinery> machineries = db.Machineries.Include(m => m.Supplier)
                .Include(m =>
m.MachineryClass);

            List<MachineryClass> machineryClasses = db.MachineryClasses.ToList();
            machineryClasses.Insert(0, new MachineryClass { Id = 0, Name = "Bce" });
            ViewBag.MachineryClasses = new SelectList(machineryClasses, "Id", "Name",
machineryClassId);

            List<Supplier> suppliers = db.Suppliers.ToList();
            suppliers.Insert(0, new Supplier { Id = 0, Name = "Bce" });
            ViewBag.Suppliers = new SelectList(suppliers, "Id", "Name", supplierId);

            if (machineryClassId != null && machineryClassId != 0)
            {
                machineries = machineries.Where(m => m.MachineryClassId ==
machineryClassId);
            }
            if (supplierId != null && supplierId != 0)
            {
                machineries = machineries.Where(m => m.SupplierId == supplierId);
            }
        }
    }
}

```

```

    }
    return View(machineries.ToList());
}

[Authorize(Roles = "Администратор, ОтветственныйЗаЗакупку")]
public IActionResult Create()
{
    ViewBag.Suppliers = new SelectList(db.Suppliers.ToList(), "Id", "Name");
    ViewBag.Classes = new SelectList(db.MachineryClasses.ToList(), "Id", "Name");
    return View();
}

[HttpPost]
public async Task<IActionResult> Create(Machinery machinery)
{
    db.Machineries.Add(machinery);
    await db.SaveChangesAsync();
    return RedirectToAction("Machineries");
}

[Authorize(Roles = "Администратор, ОтветственныйЗаЗакупку")]
public async Task<IActionResult> Edit(int? id)
{
    if (id != null)
    {
        Machinery machinery = await db.Machineries.Include(m =>
m.MachineryClass).FirstOrDefaultAsync(p => p.Id == id);
        if (machinery != null)
        {
            ViewBag.Suppliers = new SelectList(db.Suppliers.ToList(), "Id",
"Name");
            ViewBag.Classes = new SelectList(db.MachineryClasses.ToList(), "Id",
"Name");
            return View(machinery);
        }
    }
    return NotFound();
}

[HttpPost]
public async Task<IActionResult> Edit(Machinery machinery)
{
    db.Machineries.Update(machinery);
    await db.SaveChangesAsync();
    return RedirectToAction("Machineries");
}

[Authorize(Roles = "Администратор, ОтветственныйЗаЗакупку")]
[HttpGet]
[ActionName("FlagDelete")]
public async Task<IActionResult> ConfirmFlagDelete(int? id)
{
    if (id != null)
    {
        Machinery machinery = await db.Machineries.FirstOrDefaultAsync(p => p.Id
== id);
        if (machinery != null)
            return View(machinery);
    }
    return NotFound();
}

[HttpPost]
public async Task<IActionResult> FlagDelete(int? id)

```

```

{
    if (id != null)
    {
        Machinery machinery = await db.Machineries.FirstOrDefaultAsync(p => p.Id
== id);

        if (machinery != null)
        {
            machinery.isDeleted = true;
            db.Machineries.Update(machinery);
            await db.SaveChangesAsync();
            return RedirectToAction("Machineries");
        }
    }
    return NotFound();
}

[Authorize(Roles = "Администратор, ОтветственныйЗаЗакупку")]
[HttpGet]
[ActionName("Delete")]
public async Task<IActionResult> ConfirmDelete(int? id)
{
    if (id != null)
    {
        Machinery machinery = await db.Machineries.FirstOrDefaultAsync(p => p.Id
== id);

        if (machinery != null)
            return View(machinery);
    }
    return NotFound();
}

[HttpPost]
public async Task<IActionResult> Delete(int? id)
{
    if (id != null)
    {
        Machinery machinery = await db.Machineries.FirstOrDefaultAsync(p => p.Id
== id);

        if (machinery != null)
        {
            db.Machineries.Remove(machinery);
            await db.SaveChangesAsync();
            return RedirectToAction("Machineries");
        }
    }
    return NotFound();
}

[Authorize(Roles = "Администратор, ОтветственныйЗаЗакупку")]
[HttpGet]
[ActionName("Restore")]
public async Task<IActionResult> ConfirmRestore(int? id)
{
    if (id != null)
    {
        Machinery machinery = await db.Machineries.FirstOrDefaultAsync(p => p.Id
== id);

        if (machinery != null)
            return View(machinery);
    }
    return NotFound();
}

[HttpPost]

```

```

        public async Task<IActionResult> Restore(int? id)
        {
            if (id != null)
            {
                Machinery machinery = await db.Machineries.FirstOrDefaultAsync(p => p.Id
== id);

                if (machinery != null)
                {
                    machinery.isDeleted = false;
                    db.Machineries.Update(machinery);
                    await db.SaveChangesAsync();
                    return RedirectToAction("Machineries");
                }
            }
            return NotFound();
        }

        public async Task<IActionResult> Details(int? id)
        {
            if (id != null)
            {
                List<Order> orders = await db.Orders.Include(o => o.Client)
                                                        .Include(o => o.SalesManager)
                                                        .Include(o => o.PurchaseManager)
                                                        .Include(o => o.OrderStatus)
                                                        .Include(o => o.PaymentStatus)
                                                        .Include(o => o.Machinery)
                                                        .Include(o =>
o.Machinery.MachineryClass)
                                                        .Include(o => o.Machinery.Supplier)
                                                        .Where(o => o.MachineryId == id)
                                                        .ToListAsync();

                if (orders != null)
                    return View(orders);
            }
            return NotFound();
        }

        [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore =
true)]
        public IActionResult Error()
        {
            return View(new ErrorViewModel { RequestId = Activity.Current?.Id ??
HttpContext.TraceIdentifier });
        }
    }
}

```

Код контроллера PurchaseManagerOrderController:

```

using DealerCenter.Models;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using System;
using System.Diagnostics;
using System.Linq;
using System.Security.Claims;
using System.Threading.Tasks;

namespace DealerCenter.Controllers
{
    public class PurchaseManagerOrderController : Controller
    {
        DealerCenterContext db;
    }
}

```

```

public PurchaseManagerOrderController(DealerCenterContext context)
{
    db = context;
}

public IActionResult PurchaseManagerOrders()
{
    IQueryable<Order> order = db.Orders.Where(o => o.PurchaseManager.Login ==
User.Identity.Name ||
                                o.SalesManager.Login ==
User.Identity.Name ||
User.HasClaim(ClaimsIdentity.DefaultRoleClaimType, RoleEnum.Администратор.ToString()))
    .Where(o => o.PaymentStatusId ==
(int)PaymentStatusEnum.ОжиданиеПолнойОплаты)
    .Include(o => o.Client)
    .Include(o => o.PurchaseManager)
    .Include(o => o.SalesManager)
    .Include(o => o.Machinery)
    .Include(o => o.Machinery.MachineryClass)
    .Include(o => o.Machinery.Supplier)
    .Include(o => o.OrderStatus)
    .Include(o => o.PaymentStatus);

    return View(order.ToList());
}

[Authorize(Roles = "Администратор, ОтветственныйЗаЗакупку")]
[HttpGet]
[ActionName("ChangeOrderStatusToWaitingForDelivery")]
public async Task<IActionResult>
ConfirmChangeOrderStatusToWaitingForDelivery(int? id)
{
    if (id != null)
    {
        Order order = await db.Orders.FirstOrDefaultAsync(p => p.Id == id);
        if (order != null)
        {
            return View(order);
        }
    }
    return NotFound();
}

[HttpPost]
public async Task<IActionResult>
ChangeOrderStatusToWaitingForDelivery(OrderSetDateViewModel orderModel)
{
    Order order = db.Orders.Find(orderModel.Id);
    order.OrderStatus =
db.OrderStatuses.Find((int)OrderStatusEnum.ОжиданиеДоставки);
    order.DeliveryDate = orderModel.DeliveryDate;
    db.Orders.Update(order);
    await db.SaveChangesAsync();
    return RedirectToAction("PurchaseManagerOrders");
}

[Authorize(Roles = "Администратор, ОтветственныйЗаЗакупку")]
[HttpGet]
[ActionName("ChangeOrderStatusToReadyToTransfer")]
public async Task<IActionResult> ConfirmChangeOrderStatusToReadyToTransfer(int?
id)
{
    if (id != null)

```

```

        {
            Order order = await db.Orders.FirstOrDefaultAsync(p => p.Id == id);
            if (order != null)
            {
                return View(order);
            }
        }
        return NotFound();
    }

    [HttpPost]
    public async Task<IActionResult>
ChangeOrderStatusToReadyToTransfer(OrderSetDateViewModel orderModel)
    {
        Order order = db.Orders.Find(orderModel.Id);
        order.OrderStatus =
db.OrderStatuses.Find((int)OrderStatusEnum.ГотовКПередаче);
        order.DeliveryDate = DateTime.Now;
        db.Orders.Update(order);
        await db.SaveChangesAsync();
        return RedirectToAction("PurchaseManagerOrders");
    }

    [Authorize(Roles = "Администратор, ОтветственныйЗаЗакупку")]
    [HttpGet]
    [ActionName("Cancel")]
    public async Task<IActionResult> ConfirmCancel(int? id)
    {
        if (id != null)
        {
            Order order = await db.Orders.FirstOrDefaultAsync(p => p.Id == id);
            if (order != null)
            {
                return View(order);
            }
        }
        return NotFound();
    }

    [HttpPost]
    public async Task<IActionResult> Cancel(int? id)
    {
        if (id != null)
        {
            Order order = await db.Orders.FirstOrDefaultAsync(p => p.Id == id);
            if (order != null)
            {
                order.ClosingDate = DateTime.Now;
                order.PaymentStatus =
db.PaymentStatuses.Find((int)PaymentStatusEnum.ЗаказОтменён);
                order.OrderStatus =
db.OrderStatuses.Find((int)OrderStatusEnum.ЗаказОтменён);
                db.Orders.Update(order);
                await db.SaveChangesAsync();
                return RedirectToAction("PurchaseManagerOrders");
            }
        }
        return NotFound();
    }

    public async Task<IActionResult> Details(int? id)
    {
        if (id != null)
        {
            Order order = await db.Orders.Include(o => o.Client)
                .Include(o => o.SalesManager)

```

```

        .Include(o => o.PurchaseManager)
        .Include(o => o.OrderStatus)
        .Include(o => o.PaymentStatus)
        .Include(o => o.Machinery)
        .Include(o => o.Machinery.MachineryClass)
        .Include(o => o.Machinery.Supplier)
        .FirstOrDefaultAsync(p => p.Id == id);

        if (order != null)
            return View(order);
    }
    return NotFound();
}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore =
true)]
public IActionResult Error()
{
    return View(new ErrorViewModel { RequestId = Activity.Current?.Id ??
HttpContext.TraceIdentifier });
}
}
}

```

Код контроллера SalesManagerOrderController:

```

using DealerCenter.Models;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;

namespace DealerCenter.Controllers
{
    public class ClientController : Controller
    {
        DealerCenterContext db;

        public ClientController(DealerCenterContext context)
        {
            db = context;
        }

        public IActionResult Clients()
        {
            IQueryable<Client> clients = db.Clients.Include(c => c.Orders);
            return View(clients.ToList());
        }

        [Authorize(Roles = "Администратор, Продавец")]
        public IActionResult Create()
        {
            return View();
        }

        [HttpPost]
        public async Task<IActionResult> Create(Client client)
        {
            db.Clients.Add(client);
            await db.SaveChangesAsync();
            return RedirectToAction("Clients");
        }
    }
}

```

```

[Authorize(Roles = "Администратор, Продавец")]
public async Task<IActionResult> Edit(int? id)
{
    if (id != null)
    {
        Client client = await db.Clients.FirstOrDefaultAsync(p => p.Id == id);
        if (client != null)
        {
            return View(client);
        }
    }
    return NotFound();
}

[HttpPost]
public async Task<IActionResult> Edit(Client client)
{
    db.Clients.Update(client);
    await db.SaveChangesAsync();
    return RedirectToAction("Clients");
}

[Authorize(Roles = "Администратор, Продавец")]
[HttpGet]
[ActionName("Delete")]
public async Task<IActionResult> ConfirmDelete(int? id)
{
    if (id != null)
    {
        Client client = await db.Clients.FirstOrDefaultAsync(p => p.Id == id);
        if (client != null)
        {
            return View(client);
        }
    }
    return NotFound();
}

[HttpPost]
public async Task<IActionResult> Delete(int? id)
{
    if (id != null)
    {
        Client client = await db.Clients.FirstOrDefaultAsync(p => p.Id == id);
        if (client != null)
        {
            db.Clients.Remove(client);
            await db.SaveChangesAsync();
            return RedirectToAction("Clients");
        }
    }
    return NotFound();
}

public async Task<IActionResult> Details(int? id)
{
    if (id != null)
    {
        List<Order> orders = await db.Orders.Include(o => o.Client)
                                             .Include(o => o.SalesManager)
                                             .Include(o => o.PurchaseManager)
                                             .Include(o => o.OrderStatus)
                                             .Include(o => o.PaymentStatus)
                                             .Include(o => o.Machinery)
                                             .Include(o =>
o.Machinery.MachineryClass)

```



```

        .Include(o => o.Machinery.Supplier)
        .Where(o => o.ClientId == id)
        .ToListAsync();

        if (orders != null)
            return View(orders);
    }
    return NotFound();
}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore =
true)]
public IActionResult Error()
{
    return View(new ErrorViewModel { RequestId = Activity.Current?.Id ??
HttpContext.TraceIdentifier });
}
}
}

```

### Код контроллера SupplierController:

```

using DealerCenter.Models;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;

namespace DealerCenter.Controllers
{
    public class SupplierController : Controller
    {
        DealerCenterContext db;

        public SupplierController(DealerCenterContext context)
        {
            db = context;
        }

        public IActionResult Suppliers()
        {
            IQueryable<Supplier> supplier = db.Suppliers.Include(s => s.Machineries);
            return View(supplier.ToList());
        }

        [Authorize(Roles = "Администратор, ОтветственныйЗаЗакупку")]
        public IActionResult Create()
        {
            return View();
        }

        [HttpPost]
        public async Task<IActionResult> Create(Supplier supplier)
        {
            db.Suppliers.Add(supplier);
            await db.SaveChangesAsync();
            return RedirectToAction("Suppliers");
        }

        [Authorize(Roles = "Администратор, ОтветственныйЗаЗакупку")]
        public async Task<IActionResult> Edit(int? id)
        {

```

```

        if (id != null)
        {
            Supplier supplier = await db.Suppliers.FirstOrDefaultAsync(p => p.Id ==
id);
            if (supplier != null)
            {
                return View(supplier);
            }
        }
        return NotFound();
    }

    [HttpPost]
    public async Task<IActionResult> Edit(Supplier supplier)
    {
        db.Suppliers.Update(supplier);
        await db.SaveChangesAsync();
        return RedirectToAction("Suppliers");
    }

    [Authorize(Roles = "Администратор, ОтветственныйЗаЗакупку")]
    [HttpGet]
    [ActionName("FlagDelete")]
    public async Task<IActionResult> ConfirmFlagDelete(int? id)
    {
        if (id != null)
        {
            Supplier supplier = await db.Suppliers.FirstOrDefaultAsync(p => p.Id ==
id);
            if (supplier != null)
            {
                return View(supplier);
            }
        }
        return NotFound();
    }

    [HttpPost]
    public async Task<IActionResult> FlagDelete(int? id)
    {
        if (id != null)
        {
            Supplier supplier = await db.Suppliers.Include(s =>
s.Machineries).FirstOrDefaultAsync(p => p.Id == id);
            if (supplier != null)
            {
                supplier.IsDeleted = true;
                db.Suppliers.Update(supplier);
                await db.SaveChangesAsync();
                foreach (var machinery in supplier.Machineries)
                {
                    machinery.IsDeleted = true;
                    await db.SaveChangesAsync();
                }
                return RedirectToAction("Suppliers");
            }
        }
        return NotFound();
    }

    [Authorize(Roles = "Администратор, ОтветственныйЗаЗакупку")]
    [HttpGet]
    [ActionName("Delete")]
    public async Task<IActionResult> ConfirmDelete(int? id)
    {

```

```

        if (id != null)
        {
            Supplier supplier = await db.Suppliers.FirstOrDefaultAsync(p => p.Id ==
id);
            if (supplier != null)
                return View(supplier);
        }
        return NotFound();
    }

    [HttpPost]
    public async Task<IActionResult> Delete(int? id)
    {
        if (id != null)
        {
            Supplier suppliers = await db.Suppliers.FirstOrDefaultAsync(p => p.Id ==
id);
            if (suppliers != null)
            {
                db.Suppliers.Remove(suppliers);
                await db.SaveChangesAsync();
                return RedirectToAction("Suppliers");
            }
        }
        return NotFound();
    }

    [Authorize(Roles = "Администратор, ОтветственныйЗаЗакупку")]
    [HttpGet]
    [ActionName("Restore")]
    public async Task<IActionResult> ConfirmRestore(int? id)
    {
        if (id != null)
        {
            Supplier supplier = await db.Suppliers.FirstOrDefaultAsync(p => p.Id ==
id);
            if (supplier != null)
                return View(supplier);
        }
        return NotFound();
    }

    [HttpPost]
    public async Task<IActionResult> Restore(int? id)
    {
        if (id != null)
        {
            Supplier supplier = await db.Suppliers.FirstOrDefaultAsync(p => p.Id ==
id);
            if (supplier != null)
            {
                supplier.isDeleted = false;
                db.Suppliers.Update(supplier);
                await db.SaveChangesAsync();
                return RedirectToAction("Suppliers");
            }
        }
        return NotFound();
    }

    public async Task<IActionResult> Details(int? id)
    {
        if (id != null)
        {

```

```

        List<Machinery> machineries = await db.Machineries.Include(m =>
m.Supplier)
                                .Include(m =>
m.MachineryClass)
                                .Where(m =>
m.SupplierId == id)
                                .ToListAsync();
        if (machineries != null)
            return View(machineries);
    }
    return NotFound();
}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore =
true)]
public IActionResult Error()
{
    return View(new ErrorViewModel { RequestId = Activity.Current?.Id ??
HttpContext.TraceIdentifier });
}
}
}

```