**Выбор архитектуры программного обеспечения**

В результате проведенного анализа деятельности предприятия ООО «Доксал-Юг» в части организации продаж алюминиевых вентилируемых фасадов было принято решение о проектировании и разработке программного обеспечения для автоматизации продаж на данном предприятии.

В начале разработки необходимо было выбрать и описать тип архитектуры программного обеспечения.

Для начала была разработана функциональная модель модуля учета продаж, представленная на рисунке 4.

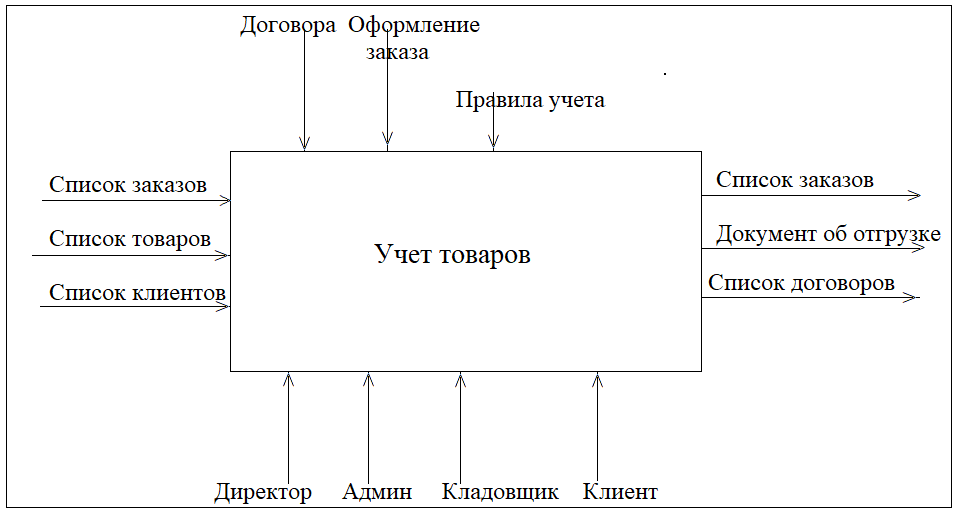


Рисунок 4 – Функциональная схема модуля учета продаж

Разработка функциональной схемы приложения (информационной системы) происходит согласно руководящему документу по стандартизации РД 50-34.698-90 «Автоматизированные системы требования к содержанию документов». В требованиях данного документа описаны нормы и правила при разработке функциональных схем.

На первом этапе создана функциональная схема, описывающая основные процессы учета товаров на предприятии. В системе должны храниться список товаров, список клиентов и список заказов – данная информация является входящей в программный модуль. Выходящей информацией из системы являются: список заказов, документ об отгрузке и список договоров с клиентами. Внутри системы должны выполняться следующие функции: оформление заказов клиентов, создание договора на покупку, проектирование, доставку товаров при соблюдении правил учета продукции предприятия ООО «Доксал-Юг».

Далее разработанную функциональную схему необходимо расширить и дополнить промежуточными связами и функциями, которые позволят описать разрабатываемую систему с точки зрения процессов использования. Расширенная функциональная схема представлена на рисунке 5.

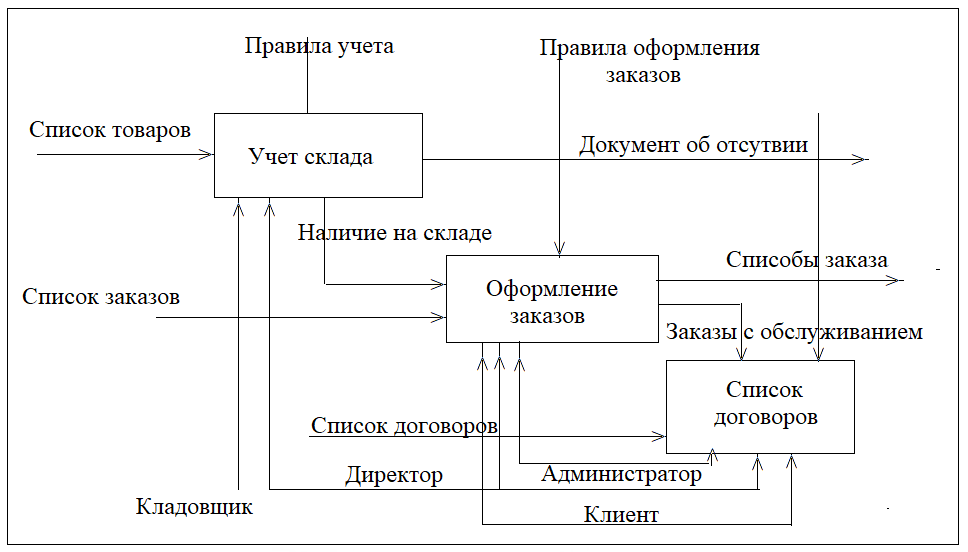


Рисунок 5 – Расширенная функциональная схема

В расширенной функциональной схеме представлены пользователи системы: кладовщик, директор, администратор и клиент и описаны процессы взаимодействия между составляющими системы такими как «Учет склада», «Оформление заказов» и «Список договоров».

Работу системы можно описать следующим образом:

- товары поступают на скала компании и оприходываются кладовщиком согласно правилам ведения складского учета;

- при оформлении заказа для клиента система проверяет наличие необходимого количества товара на складе, если количество необходимое для обеспечения договора с клиентом присутствует на складе предприятия, то сразу происходит заключение договора, иначе формируется документ об отсутствии необходимого количества товара и заключается договор с учетом заказа товара у поставщика (в компании ООО «Доксал-Юг» ведется собственной производство, таким образом заказ предается в отдел производства);

- со стороны клиента и директора компании ООО «Доксал-Юг» подписывается договор на оказание услуг (поставку, проектирование или разработку алюминиевого фасада по согласованному макету).

Далее необходимо разработать диаграмму прецедентов, которая применяется для моделирования вида системы с точки зрения внешнего наблюдателя (актора). На диаграмме прецедентов графически показана совокупность прецедентов и акторов, а также отношения между ними. На рисунке 6 представлена, разработанная диаграмма прецедентов для модуля учета продаж товаров компании ООО «Доксал-Юг».

В процессе проектирования были выделены следующие субъекты (акторы) системы:

- Администратор;

- Пользователь (бухгалтер/менеджер);

- Клиент.

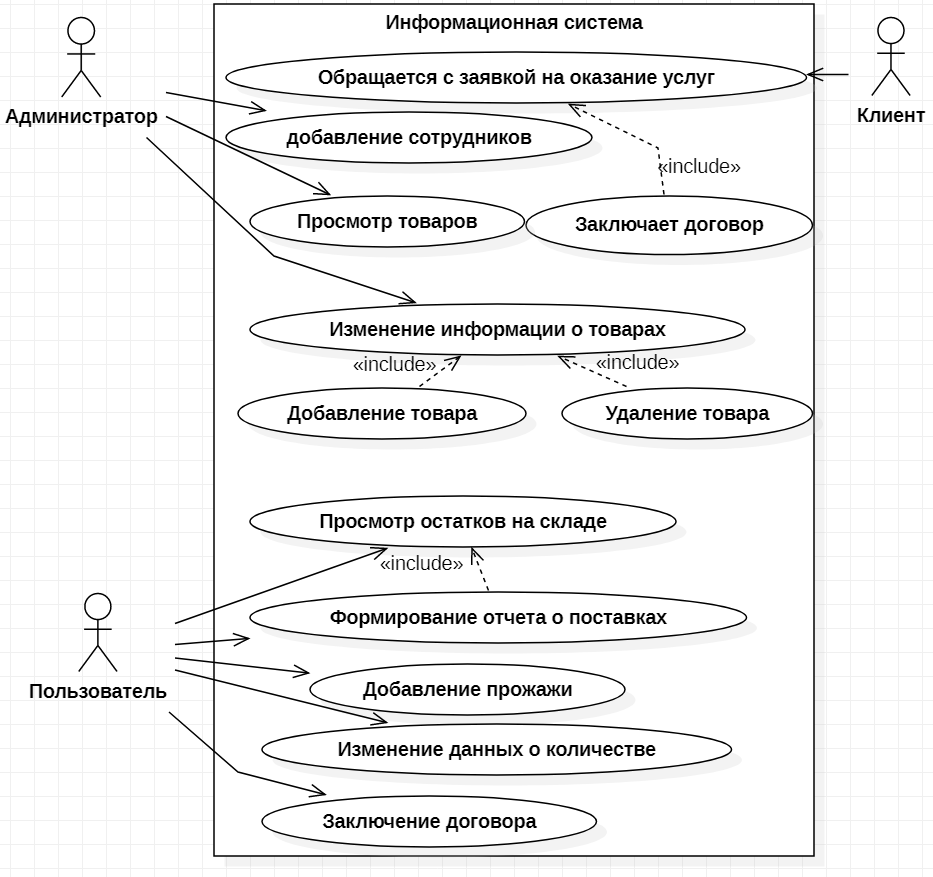


Рисунок 6 – Диаграмма прецедентов

Прецеденты (use case) для каждого субъекта представлены в виде структурных элементов (предоставленных графически в виде овалов) с названием прецедента для каждого субъекта.

В разрабатываемой системе каждый из субъектов имеет следующие прецеденты:

- субъект «Клиент» - прецедент «Обращается с заявкой на оказание услуг», данный прецедент включает в себя (include) прецедент «Заключает договор»;

- субъект «Администратор» - прецеденты: «Добавление сотрудника», «Просмотр товаров», «Изменение информации о товарах» данный прецедент включает в себя (include) прецеденты «Добавление товара» и «Удаление товара»;

- субъект «Пользователь» - прецеденты: «Просмотр остатков на складе» данный прецедент включает в себя (include) прецедент «Формирование отчета о поставках»; «Добавление продажи», «Изменение данных о количестве» и «Заключение договора».

Таким образом из диаграммы прецедентов видно, что с системой напрямую работают только субъекты «Администратор» и «Пользователь», субъект «Клиент» взаимодействует с системой через субъект «Пользователь».

## **Проектирование информационного обеспечения**

Проектирование приложения было начато с разработки форм приложения и организации логики взаимодействия между ними. Разработка логики взаимодействия между структурными элементами должна быть продумана таким образом, чтобы пользователю был интуитивно понятен каждый шаг работы с программным обеспечением.

Для отображения логики взаимодействия между формами приложения была разработана схема логического взаимодействия, представленная на рисунке 7.

Были спроектированы пять самостоятельных окон приложения:

- «Окно авторизация»;

- «Главное окно»;

- «Окно справочники»;

- «Окно добавления/перемещения товаров»;

- «Окно добавления/списания товаров».

Также помимо окон в приложении будут использоваться страницы (frame), которые будут размещены (отрываться) при выполнении соответствующих операций с данными внутри окон (window):

- Страница пополнения товаров на складе;

- Страница добавления товаров;

- Страница продаж товаров;

- Страница добавления продажи;

- Страница остатков товаров;

- Страница перемещений товаров с одного склада на другой при необходимости;

- Страница учета списания товаров;

- Страница с информацией о товарах;

- Страница со справочниками по товарам, единицам измерения, поставщикам комплектующих, группах товаров, контрагентов.

Изображение выглядит как текст, диаграмма, Параллельный, снимок экрана

Автоматически созданное описание

Рисунок 7 - Схема логического взаимодействия между структурными элементами приложения

## **2.4 Проектирование структуры базы данных**

После проектирования приложения и логики взаимодействия между его структурными элементами была спроектирована и разработана база данных «Sales».

Сначала была проработана концептуальная модель данных для будущей базы данных, в которой будет хранится вся необходимая информация для автоматизации процессов продаж алюминиевых вентилируемых фасадов компанией ООО «Доксал-Юг».

На рисунке 8 представлена разработанная концептуальная модель базы данных.

Изображение выглядит как текст, диаграмма, число, План

Автоматически созданное описание

Рисунок 8 – Концептуальная модель базы данных «Sales»

После разработки концептуальной схемы в среде SQL Management Studio была спроектирована и разработана физическая база данных «Sales», диаграмма которой представлена на рисунке 9.

База данных «Sales» состоит из тринадцати таблиц, которые полностью описывают предметную область разработки и позволят хранить всю необходимую информацию для ведения учета продаж в компании ООО «Доксал-Юг».

Изображение выглядит как текст, снимок экрана, внутренний

Автоматически созданное описание

Рисунок 9 - Вид ER-диаграммы базы данных «Sales»

База данных «Sales» содержит следующие таблицы и позволяет хранить в них информацию следующего содержания:

**Таблица «Остатки товаров» (Remaining\_Goods):**

Поле RemainingGoodId – тип int, первичный ключ;

Поле ProductId – тип int, внешний ключ;

Поле WarehouseId – тип int, внешний ключ;

Поле Количество Quantity – тип int.

**Таблица «Товары» (Products):**

Поле ProductId – тип int, первичный ключ;

Поле Код (Code) – тип int;

Поле Артикул (VendorCode) – тип int;

Поле GroupId – тип int, внешний ключ;

Поле Наименование (Name) – тип nvarchar(100);

Поле Ид единицы измерения (UnitId) – тип int, внешний ключ;

Поле Ид поставщика (SupplierId) – тип int, внешний ключ;

Поле Описание (Description) – тип nvarchar(250).

**Таблица «Группы товаров» (Groups):**

Поле Ид группы (GroupId) – тип int, первичный ключ;

Поле Наименование (Name) – тип nvarchar(100);

Поле Описание (Description) – тип nvarchar(250).

**Таблица «Единицы измерения» (Units):**

Поле Ид единицы измерения (UnitId) – тип int, первичный ключ;

Поле Наименование (Name) – тип nvarchar(100);

Поле Описание (Description) – тип nvarchar(250).

**Таблица «Поставщики» (Suppliers):**

Поле Ид поставщика (SupplierId) – тип int, первичный ключ;

Поле Наименование (Name) – тип nvarchar(100);

Поле Описание (Description) – тип nvarchar(250).

**Таблица «Склады» (Warehouses):**

Поле Ид склада (WarehouseId) – тип int, первичный ключ;

Поле Наименование (Name) – тип nvarchar(100);

Поле Описание (Description) – тип nvarchar(250).

**Таблица «Поступившие товары» (SuppliedGoods):**

Поле Ид поступившего товара (SuppliedGoodId) – тип int, первичный ключ;

Поле Ид поступления (SupplyingId) – тип int, внешний ключ;

Поле Ид остатков товара (RemainingGoodId) – тип int, внешний ключ;

Поле Количество (Quantity) – тип int;

Поле Цена покупки (PurchasePrice) – тип float.

**Таблица «Поступления» (Supplyings):**

Поле Ид поступления (SupplyingId) – тип int, первичный ключ;

Поле Дата (Date) – тип Date;

Поле Номер (Number) – тип int;

Поле Ид контрагента (CounterpartyId) – тип int, внешний ключ;

Поле Сумма (Sum) – тип float;

**Таблица «Контрагенты» (Counterparties):**

Поле Ид контрагента (CounterpartyId) – тип int, первичный ключ;

Поле Наименование (Name) – тип nvarchar(100);

Поле Описание (Description) – тип nvarchar(250).

**Таблица «Проданные товары» (ProductsSold):**

Поле Ид проданного товара (ProductSoldId) – тип int, первичный ключ;

Поле Ид продажи (SaleId) – тип int, внешний ключ;

Поле Ид остатков товара (RemainingGoodId) – тип int, внешний ключ;

Поле Количество (Quantity) – тип int;

Поле Цена продажи покупки (SellingPrice) – тип float.

**Таблица «Продажи» (Sales):**

Поле Ид продажи (SaleId) – тип int, первичный ключ;

Поле Дата (Date) – тип Date;

Поле Номер (Number) – тип int;

Поле Реквизиты покупателя (BuyersRequisites) – тип nvarchar(250)$

Поле Сумма (Sum) – тип float.

**Таблица «Передвижения товаров между складами» (MovingGoods):**

Поле Ид перемещения товара (MovingGoodId) – тип int, первичный ключ;

Поле Дата (Date) – тип Date;

Поле Номер (Number) – тип int;

Поле Ид остатков товара отправления (RemainingGoodIdFrom) – тип int, внешний ключ;

Поле Ид остатков товара назначения (RemainingGoodIdTo) – тип int, внешний ключ;

Поле Количество (Quantity) – тип int;

Поле Описание (Description) – тип nvarchar(250).

**Таблица «Списание товаров» (WriteOffs):**

Поле Ид списания (WriteOffId) – тип int, первичный ключ;

Поле Дата (Date) – тип Date;

Поле Номер (Number) – тип int;

Поле Ид остатков товара (RemainingGoodId) – тип int, внешний ключ;

Поле Количество (Quantity) – тип int;

Поле Описание (Description) – тип nvarchar(250).

## **Описание основных элементов программного модуля для организации продаж товаров компании ООО «Доксал-Юг»**

Рассмотрим основные разработанные элементы приложения для организации продаж товаров компании ООО «Доксал-Юг».

При запуске приложения пользователь попадает на окно авторизации, представленное на рисунке 12.

Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание

Рисунок 12 – Вид окна «Авторизация»

Для успешной авторизации необходимо ввести логин и пароль, которые соответствуют логину и паролю пользователя, прописанного в базе данных (процесс создания пользователя базы данных описан выше).

При условии совпадения введенных в поля «Логин» и «Пароль» данных представленной формы с логином и паролем, указанными при создании пользователя в базе данных, пользователь попадет на окно «Главная страница», представленное на рисунке 14.

Если введенные логин и пароль не совпадут с хранящимися логином и паролем созданном для базы данных, пользователю в окне «Авторизация» будет выведено сообщение: «Логин\Пароль введены неверно!» вид окна с сообщением об ошибке представлен на рисунке 13.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 13 – Вид окна «Авторизация» при неверно введенным логине или пароле

При запуске окна программы пользователь видит пустое с тремя вкладками: «Поступления», «Продажи», «Остатки», а также меню «Справочники».

Изображение выглядит как текст, снимок экрана, линия, Прямоугольник

Автоматически созданное описание

Рисунок 14 – Вид окна «Главное окно»

Для дальнейшей работы пользователю системы необходимо выбрать одну из вкладок. При выборе вкладке «Поступления» в окне откроется выбранная страница (рисунок 15).

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

Автоматически созданное описание

Рисунок 15 – Вид окна «Поступления»

При выборе определенной вкладки название окна меняется на соответствующее выполняемому действию. Например, если пользователь откроет вкладку «Поступления» и выберет действие «Добавить поступление», то заголовок открытого окна сменит название на «Добавление поступления товаров» как показано на рисунке 15.

Работа с вкладками «Продажи» и «Остатки» производится аналогичным образом, как и в случае с вкладкой «Поступления».

Для удобной работы с различными видами информации, хранящейся в базе данных в системе предусмотрен ряд справочников, которые позволяют пользователя быстрее сориентироваться при работе с системой.

Система справочников выполнена в виде выпадающего списка меню, как показано на рисунке 16, и содержит следующие справочники: «Товары», «Группы товаров», «Единицы измерения», «Склады», «Поставщики», «Контрагенты».