# **Руководство Администратора**

Оглавление

[**Руководство Администратора** 1](#_Toc199279256)

[**1. Общие сведения о системе** 1](#_Toc199279257)

[**2. Детальное описание пользовательского интерфейса** 3](#_Toc199279258)

[**2.1 Структура главного окна администратора** 3](#_Toc199279259)

[**2.2 Спецификация столбцов таблицы продукции** 5](#_Toc199279260)

[**3. Подробные алгоритмы работы с системой** 5](#_Toc199279261)

[**3.1 Процедуры управления продуктами** 5](#_Toc199279262)

[**3.2 Функциональность работы с материалами** 8](#_Toc199279263)

[**4. Математические модели и алгоритмы расчетов** 8](#_Toc199279264)

[**4.1 Модель расчета себестоимости продукции** 8](#_Toc199279265)

[**4.2 Алгоритм расчета потребности в материалах** 10](#_Toc199279266)

[**5. Технические требования и архитектура системы** 11](#_Toc199279267)

[**5.1 Технические требования к программно-аппаратной среде** 11](#_Toc199279268)

[**5.2 Архитектура базы данных** 12](#_Toc199279269)

[**6. Приложения и сопроводительная документация** 14](#_Toc199279270)

[**6.1 Скрипт инициализации базы данных** 14](#_Toc199279271)

[**6.2 Диаграмма сущность-связь (ER-диаграмма)** 15](#_Toc199279272)

[**6.3 Конфигурационные настройки** 16](#_Toc199279273)

[**Дополнительные рекомендации:** 17](#_Toc199279274)

# **1. Общие сведения о системе**

Система представляет собой комплексное настольное приложение, разработанное для автоматизации процессов управления продукцией производственной компании «Наш декор». Основные функции системы включают:

* Учет и каталогизация продукции с возможностью добавления, редактирования и удаления товаров.
* Управление справочной информацией о материалах, включая их характеристики, цены и доступность.
* Автоматический расчет себестоимости продукции на основе стоимости материалов, технологических затрат и накладных расходов.
* Формирование аналитической отчетности для контроля за производственными и финансовыми показателями.

Приложение разработано с использованием современных технологий:

* Платформа: Microsoft .NET Framework 4.8 (или .NET 6.0), язык программирования C#.
* Интерфейс: Windows Forms для создания интуитивного пользовательского интерфейса.
* База данных: Microsoft SQL Server (включая Express-версию) для хранения данных с поддержкой транзакций, хранимых процедур и триггеров.

Архитектура системы реализована по принципу клиент-серверного взаимодействия:

* Клиентская часть отвечает за отображение данных, обработку пользовательского ввода и реализацию бизнес-логики (например, валидацию данных перед сохранением).
* Серверная часть обеспечивает хранение данных, их защиту и целостность через механизмы реляционной базы данных. Все операции с данными выполняются через параметризованные SQL-запросы, что минимизирует риск инъекций и других уязвимостей.

Система предназначена для административного персонала предприятия и позволяет централизованно управлять информацией, обеспечивая:

* Единое хранилище данных с возможностью оперативного обновления.
* Ролевой доступ для пользователей с разными уровнями привилегий.
* Интеграцию с внешними системами (например, ERP или CRM) через API или импорт/экспорт данных.

# **2. Детальное описание пользовательского интерфейса**

# **2.1 Структура главного окна администратора**

Главное окно приложения выполнено в классическом стиле Windows-приложений и состоит из следующих функциональных зон:

1. Заголовочная область:
   * Расположена в верхней части окна.
   * Содержит логотип компании «Наш декор» в левом верхнем углу. Логотип служит не только элементом фирменного стиля, но и индикатором успешного запуска приложения.
   * Справа от логотипа может быть размещено меню (например, «Файл», «Справка») для доступа к дополнительным функциям.
2. Панель инструментов:
   * Горизонтальная панель под заголовочной областью с кнопками, обеспечивающими быстрый доступ к основным операциям.
   * Кнопки:
     + «Добавить» (иконка + текст): открывает форму для создания новой записи о продукте.
     + «Изменить» (иконка + текст): активируется после выбора строки в таблице, позволяет редактировать данные.
     + «Удалить» (иконка + текст): удаляет выбранную запись с подтверждением.
     + «Материалы» (иконка + текст): отображает список материалов для выбранного продукта.
     + «Поиск» (текстовое поле): фильтрует данные по артикулу, наименованию или другим критериям.
     + «Выход» (иконка + текст): завершает работу приложения с сохранением текущего состояния.
3. Центральная рабочая область:
   * Занимает основное пространство окна.
   * Содержит таблицу данных (DataGridView), отображающую информацию о продукции.
   * Функционал таблицы:
     + Сортировка по столбцам (например, по возрастанию/убыванию цены).
     + Выделение строк для последующего редактирования или удаления.
     + Контекстное меню (вызывается правой кнопкой мыши) с действиями: копирование данных, экспорт в Excel, печать.
     + Возможность изменения ширины столбцов и порядка их отображения.
4. Строка состояния:
   * Расположена в нижней части окна.
   * Отображает общее количество записей (например, «Всего продуктов: 150»).
   * Может включать дополнительную информацию: текущее время, статус подключения к базе данных.

# **2.2 Спецификация столбцов таблицы продукции**

Таблица содержит следующие столбцы с соответствующими характеристиками:

Артикул - текстовое поле переменной длины, содержащее уникальный идентификатор продукта. Поле не допускает пустых значений и дублирования записей. Формат артикула определяется внутренними стандартами предприятия.

Тип продукта - справочное поле, значения которого выбираются из предопределенного списка категорий продукции. Связано с таблицей типов продуктов через внешний ключ.

Наименование - текстовое поле для полного названия продукта, поддерживающее кириллические и латинские символы, цифры и специальные знаки.

Расчетная цена - вычисляемое поле, отображающее автоматически рассчитанную стоимость продукта на основе входящих материалов и накладных расходов. Значение представлено в денежном формате с двумя знаками после запятой.

Минимальная стоимость - редактируемое числовое поле для установки минимальной цены реализации партнерам. Принимает только положительные значения.

Ширина рулона - числовое поле с плавающей точкой, определяющее размер продукта в метрах с точностью до сотых долей (например, 0.91 м).

# **3. Подробные алгоритмы работы с системой**

# **3.1 Процедуры управления продуктами**

Алгоритм добавления нового продукта:

1. Пользователь нажимает кнопку «Добавить».
2. Открывается модальное окно с формой, содержащей поля:
   * Артикул: текстовое поле с валидацией на уникальность.
   * Тип продукта: выпадающий список из справочника **ProductTypes**.
   * Наименование: многострочное поле с поддержкой длинных описаний.
   * Минимальная стоимость: числовой элемент с ограничением на положительные значения.
   * Ширина рулона: поле с плавающей точкой (например, 1.2 м).
3. Перед отправкой данных на сервер выполняется клиентская валидация:
   * Проверка заполнения обязательных полей.
   * Проверка уникальности артикула через SQL-запрос (**SELECT COUNT(\*) FROM Products WHERE Article = @input**).
   * Проверка корректности числовых значений (например, минимальная стоимость > 0).
4. При успешной валидации данные отправляются на сервер через параметризованный SQL-запрос **INSERT INTO Products (...) VALUES (...)**.
5. После сохранения:
   * Обновляется таблица в интерфейсе.
   * Закрывается модальное окно.
   * Отображается уведомление об успешном добавлении.

Алгоритм редактирования продукта:

1. Пользователь выбирает строку в таблице.
2. Активируется кнопка «Изменить».
3. Открывается форма редактирования, предварительно заполненная текущими данными.
4. Возможные ограничения:
   * Артикул может быть заблокирован для изменения, если включен режим защиты от случайных правок.
   * Проверка зависимостей (например, нельзя изменить тип продукта, если он используется в заказах).
5. Валидация данных аналогична процедуре добавления.
6. Сохранение изменений через **UPDATE Products SET ... WHERE ProductID = @id**.
7. Обновление таблицы и отображение уведомления.

Алгоритм удаления продукта:

1. Пользователь выбирает строку в таблице.
2. Нажимает кнопку «Удалить».
3. Система отображает диалог подтверждения с деталями:
   * Наименование удаляемого продукта.
   * Список связанных записей (например, материалы, заказы).
4. Перед удалением проверяются зависимости:
   * Если продукт используется в заказах, предлагаются варианты:
     + Каскадное удаление (удалить вместе с зависимостями).
     + Запрет операции с уведомлением о конфликте.
5. При подтверждении выполняется **DELETE FROM Products WHERE ProductID = @id**.
6. Обновление таблицы и отображение уведомления.

# **3.2 Функциональность работы с материалами**

Подсистема управления материалами интегрирована с основным модулем и доступна через кнопку «Материалы» после выбора продукта.

Функционал окна материалов:

* Таблица материалов:
  + Наименование материала: ссылается на справочник **Materials**, включая артикул поставщика и единицы измерения.
  + Количество: автоматически рассчитывается по формуле:

Количество = Базовая норма × (1 + Коэффициент брака) × Планируемый объем производства

* + Цена за единицу: текущая закупочная цена с учетом НДС и доставки.
  + Стоимость: произведение количества на цену.
* Редактирование рецептуры:
  + Доступно только для пользователей с правами администратора.
  + Возможность добавления/удаления материалов, изменения коэффициентов брака.
* Интеграция со складом:
  + Отображение остатков материалов на складе.
  + Предупреждения о недостаточном количестве для производства.

# **4. Математические модели и алгоритмы расчетов**

# **4.1 Модель расчета себестоимости продукции**

Себестоимость рассчитывается по формуле:

Расчетная цена = (Σ(Стоимость материалов × Коэффициент типа продукта)) + Накладные расходы + Плановая прибыль

Компоненты модели:

1. Стоимость материалов:
   * Для каждого материала: **Количество × Цена за единицу** (с учетом НДС, доставки).
   * Пример: для продукта с 3 материалами:
     + Материал A: 10 м × 50 руб. = 500 руб.
     + Материал B: 5 кг × 200 руб. = 1000 руб.
     + Материал C: 2 л × 150 руб. = 300 руб.
     + Итого: 500 + 1000 + 300 = 1800 руб.
2. Коэффициент типа продукта:
   * Хранится в таблице **ProductTypes**.
   * Учитывает сложность производства и рыночное позиционирование.
   * Пример: для типа «Обои» коэффициент = 1.1, для «Ламинат» = 1.25.
3. Накладные расходы:
   * Включают амортизацию, зарплаты, коммунальные платежи.
   * Распределяются пропорционально трудозатратам:

Накладные расходы = (Общие накладные / Общее время производства) × Время на продукт

1. Плановая прибыль:
   * Процент от себестоимости, задаваемый администратором (например, 20%).

# **4.2 Алгоритм расчета потребности в материалах**

Формула:

Потребность = Базовая норма × (1 + Коэффициент брака) × Планируемый объем производства

Пример расчета:

* Базовая норма для материала X: 2 м² на единицу.
* Коэффициент брака: 5% (0.05).
* Планируемый объем: 1000 единиц.
* Потребность = 2 × (1 + 0.05) × 1000 = 2100 м².

Учет складских остатков:

* Если на складе есть 500 м², то:
  + Необходимо закупить: 2100 − 500 = 1600 м².
  + Система проверяет минимальную партию поставки (например, 200 м²) и корректирует заказ.

Дополнительные параметры:

* Коэффициент использования материала (например, 0.95 для сложных технологий).
* Сезонные колебания цен (автоматическое обновление через API поставщиков).

# **5. Технические требования и архитектура системы**

# **5.1 Технические требования к программно-аппаратной среде**

Для корректного функционирования системы необходимы следующие параметры оборудования и программного обеспечения:

Операционная система:

* Минимальная: Windows 10 (версия 1903 и выше) или Windows 11 с установленными обновлениями безопасности.
* Рекомендуемая: Windows 11 Pro (64-bit) с актуальными пакетами обновлений.

Аппаратные требования:

* Процессор: Intel Core i3 или AMD эквивалент (рекомендуется i5 или выше)
* Оперативная память: минимум 4 ГБ (рекомендуется 8 ГБ)
* Дисковое пространство: 500 МБ для приложения + пространство для базы данных
* Разрешение экрана: минимум 1024×768 (рекомендуется 1920×1080)

Программные зависимости:

1. Microsoft .NET Framework 4.8 или .NET 6.0 Runtime (в зависимости от версии приложения).
2. Microsoft SQL Server 2017 и выше (поддерживается Express-версия).
3. Microsoft Visual C++ Redistributable Package (2015–2022).
4. SQL Server Management Studio (SSMS) — для администрирования базы данных.

Дополнительные рекомендации:

* Настройка автоматического резервного копирования базы данных через SQL Server Agent.
* Использование SSD-накопителей для ускорения работы с БД.
* Настройка брандмауэра для разрешения портов 1433 (SQL Server) и 1434 (SQL Browser).

# **5.2 Архитектура базы данных**

База данных системы спроектирована по принципам реляционной модели с нормализацией до третьей нормальной формы (3NF). Структура включает таблицы, представления, хранимые процедуры и триггеры для обеспечения целостности данных и производительности.

**Основные таблицы:**

1. **Products** (Продукты)
   * Поля: **ProductID** (PK), **Article** (уникальный артикул), **Name**, **ProductTypeID** (FK), **MinPrice**, **RollWidth**.
   * Ограничения:
     + Уникальность артикула (**UNIQUE CONSTRAINT**).
     + Проверка положительных значений для **MinPrice** и **RollWidth**.
2. **ProductTypes** (Типы продуктов)
   * Поля: **ProductTypeID** (PK), **TypeName**, **CostCoefficient** (коэффициент для расчета себестоимости).
3. **Materials** (Материалы)
   * Поля: **MaterialID** (PK), **Name**, **SupplierID** (FK), **UnitPrice**, **UnitOfMeasure**.
   * Индексы: Созданы для полей **Name** и **SupplierID** для ускорения поиска.
4. **ProductMaterials** (Материалы продуктов)
   * Поля: **ProductID** (FK), **MaterialID** (FK), **Quantity**, **WasteCoefficient**.
   * Связь: Реализует отношение многие-ко-многим между продуктами и материалами.
5. **Suppliers** (Поставщики)
   * Поля: **SupplierID** (PK), **CompanyName**, **ContactInfo**, **DeliveryTerms**.

**Связи и целостность:**

* Все внешние ключи (FK) поддерживают каскадное обновление и ограничение удаления для предотвращения орфанных записей.
* Пример: При удалении поставщика из таблицы **Suppliers** система проверяет наличие связанных материалов в **Materials**. Если связи существуют, операция блокируется с уведомлением.

**Хранимые процедуры:**

1. **CalculateProductCost**
   * Рассчитывает себестоимость продукта на основе материалов и коэффициентов.
   * Пример вызова:

EXEC CalculateProductCost @ProductID = 101;

1. **UpdateMaterialStock**
   * Обновляет остатки материалов на складе после завершения заказа.

**Триггеры:**

1. **trg\_UpdateCalculatedPrice**
   * Автоматически пересчитывает **CalculatedPrice** в таблице **Products** при изменении рецептуры в **ProductMaterials**.

# **6. Приложения и сопроводительная документация**

# **6.1 Скрипт инициализации базы данных**

Файл **decorDB.sql** содержит полный набор команд для создания и настройки базы данных. Пример структуры:

-- Создание таблиц

CREATE TABLE ProductMaterials (

Id INT PRIMARY KEY IDENTITY(1,1),

ProductId INT NOT NULL,

MaterialId INT NOT NULL,

Quantity DECIMAL(10,4) NOT NULL,

FOREIGN KEY (ProductId) REFERENCES Products(Id) ON DELETE CASCADE,

FOREIGN KEY (MaterialId) REFERENCES Materials(Id)

);

-- Создание индексов

CREATE INDEX idx\_ProductName ON Products(Name);

-- Хранимая процедура

CREATE PROCEDURE CalculateProductCost

@ProductID INT

AS

BEGIN

SELECT SUM(pm.Quantity \* m.UnitPrice \* pt.CostCoefficient) +

(SELECT OverheadRate FROM OverheadSettings) AS TotalCost

FROM ProductMaterials pm

JOIN Materials m ON pm.MaterialID = m.MaterialID

JOIN ProductTypes pt ON pt.ProductTypeID = p.ProductTypeID

WHERE pm.ProductID = @ProductID;

END;

# **6.2 Диаграмма сущность-связь (ER-диаграмма)**

Диаграмма, представленная в файле **ER.pdf**, визуализирует структуру базы данных в нотации "воронья лапка". Основные элементы:

* Сущности: **Products**, **ProductTypes**, **Materials**, **Suppliers**, **ProductMaterials**.
* Связи:
  + **Products** ↔ **ProductTypes**: один-ко-многим.
  + **Products** ↔ **Materials**: многие-ко-многим через **ProductMaterials**.
  + **Materials** ↔ **Suppliers**: один-ко-многим.
* Кардинальность: Пример: один тип продукта может быть связан с несколькими продуктами.

# **6.3 Конфигурационные настройки**

Файл **App.config** содержит параметры подключения к базе данных и другие настройки приложения:

<configuration>

<connectionStrings>

<add name="DecorDBConnection"

connectionString="Server=.\SQLEXPRESS;Database=DecorDB;Trusted\_Connection=True;"/>

</connectionStrings>

<appSettings>

<add key="LogPath" value="C:\Logs\DecorApp\" />

<add key="MaxRetries" value="3" />

<add key="Timeout" value="30" />

</appSettings>

</configuration>

Рекомендации по настройке:

1. Строка подключения:
   * Замените **Server=.\SQLEXPRESS** на имя вашего SQL Server (например, **Server=192.168.1.100\SQL2019**).
   * Для использования SQL-логина добавьте параметры **User ID** и **Password**:

Server=myServer;Database=DecorDB;User ID=admin;Password=secret;

1. Параметры безопасности:
   * Включите шифрование соединения (SSL/TLS) для защиты данных.
   * Ограничьте права доступа к базе данных: пользователь приложения должен иметь только необходимые привилегии (SELECT, INSERT, UPDATE).
2. Логирование:
   * Укажите путь к папке логов в параметре **LogPath**.
   * Настройте ротацию логов (ежедневно/еженедельно) для предотвращения переполнения диска.

# **Дополнительные рекомендации:**

* Резервное копирование: Настройте регулярное резервное копирование БД через SQL Server Agent (ежедневно в 2:00 AM).
* Мониторинг производительности: Используйте инструменты вроде SQL Profiler для анализа медленных запросов.
* Обновления: Регулярно обновляйте SQL Server и .NET Runtime для устранения уязвимостей.