

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)

АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ
ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ
(АКТ (ф) СПбГУТ)

КУРСОВОЙ ПРОЕКТ
НА ТЕМУ

РАЗРАБОТКА ПОДСИСТЕМЫ УЧЕТА
ДОРОЖНО-СТРОИТЕЛЬНЫХ МАТЕРИАЛОВ
ДЛЯ АВТОДОРОЖНЫХ ПРОЕКТОВ

Л109. 25КП01. 013 ПЗ

(Обозначение документа)

МДК.02.01 Технология разработки
программного обеспечения

Студент

ИСПП-21

(Группа) (Подпись)

08.12.2025

(Дата)

М.А. Кореев

(И.О. Фамилия)

Преподаватель

(Подпись)

09.12.2025

(Дата)

Ю.С. Маломан

(И.О. Фамилия)

Архангельск 2025

СОДЕРЖАНИЕ

Перечень сокращений и обозначений.....	3
Введение.....	4
1 Анализ и разработка требований.....	6
1.1 Назначение и область применения.....	6
1.2 Постановка задачи	6
1.3 Выбор состава программных и технических средств	8
2 Проектирование программного обеспечения	10
2.1 Проектирование интерфейса пользователя.....	10
2.2 Разработка архитектуры программного обеспечения.....	12
2.3 Проектирование базы данных	13
3 Разработка и интеграция модулей программного обеспечения.....	15
3.1 Разработка программных модулей.....	15
3.2 Реализация интерфейса пользователя.....	18
3.3 Разграничение прав доступа пользователей	19
3.4 Экспорт и импорт данных.....	20
4 Тестирование и отладка программного обеспечения.....	22
4.1 Структурное тестирование.....	22
4.2 Функциональное тестирование	23
5 Инструкция по эксплуатации программного обеспечения	24
5.1 Установка программного обеспечения.....	24
5.2 Инструкция по работе	25
Заключение	29
Список использованных источников	30

СПИСОК ИСПОЛЬЗОВАННЫХ СОКРАЩЕНИЙ

В настоящем курсовом проекте применяют следующие сокращения и обозначения:

БД – база данных

ПК – персональный компьютер

ПО – программное обеспечение

СУБД – система управления базами данных

MSSQL – система управления базами данных

SQL – язык структурированных запросов

DOCX – формат документа

ORM – технология программирования

WPF – графическая подсистема

XAML – язык разметки

.NET – платформа для программирования

ВВЕДЕНИЕ

В современном мире автоматизация бизнес-процессов стала неотъемлемой частью успешного управления предприятием. Особенно это актуально для строительной отрасли, где учет и контроль материалов играют ключевую роль в обеспечении эффективности и рентабельности проектов. Дорожно-строительные компании сталкиваются с необходимостью точного учета строительных материалов, контроля их расхода и своевременного пополнения запасов.

Современные информационные системы позволяют автоматизировать процессы учета, что значительно сокращает временные затраты, минимизирует человеческие ошибки и повышает точность данных. Использование специализированного программного обеспечения для учета дорожно-строительных материалов позволяет не только упростить процесс управления запасами, но и обеспечить прозрачность и контроль на всех этапах строительства.

Актуальность разрабатываемого курсового проекта заключается в создании информационной системы, которая позволит автоматизировать учет дорожно-строительных материалов и своевременное пополнение запасов. Это позволит дорожно-строительным компаниям повысить эффективность использования материалов, снизить издержки и улучшить планирование проектов.

Целью курсового проектирования является разработка программного обеспечения для учета дорожно-строительных материалов, которое обеспечит возможность добавления, редактирования и удаления информации о материалах, контроля их движения, а также генерации отчетов о наличии и расходе материалов.

Для достижения поставленной цели требуется решить следующие задачи:

- провести сбор требований целевой аудитории,

- проанализировать информационные источники по предметной области,
- спроектировать архитектуру приложения,
- спроектировать диаграмму вариантов использования приложения,
- выбрать состав программных и технических средств для реализации приложения,
- спроектировать БД,
- спроектировать интерфейс оконного приложения,
- создать БД в выбранной СУБД,
- разработать интерфейс оконного приложения,
- реализовать разграничение прав доступа пользователей,
- разработать оконное приложение,
- реализовать экспорт данных в виде файлов Excel,
- выполнить структурное тестирование ПО,
- выполнить функциональное тестирование ПО,
- разработать программную и эксплуатационную документацию.

В результате выполнения поставленных задач будет разработано оконное приложение для учета и контроля строительных материалов, которое обеспечит эффективный учет запасов и генерацию отчетов. Это позволит строительным компаниям оптимизировать процессы закупок и использования материалов, а также улучшить планирование и отчетность по строительным проектам.

1 Сбор и анализ требований

1.1 Назначение и область применения

Областью применения разрабатываемой подсистемы являются дорожно-строительные организации, занимающиеся строительством, ремонтом и содержанием автомобильных дорог. Назначением использования является упрощение учёта дорожно-строительных материалов путем контроля их остатков на складах, распределения материалов по проектам и автоматического формирования необходимой отчётности. Основными категориями пользователей будут инженеры дорожно-строительных организаций.

1.2 Постановка задачи

Необходимо разработать оконное приложение, которое предоставит доступ к следующей функциональности:

- авторизация пользователей;
- просмотр, добавление, редактирование и удаление информации о материалах;
- контроль остатков материалов на складах;
- привязка материалов к проектам;
- просмотр, добавление, редактирование и удаление информации о проектах;
- просмотр и управление информации о материалах, связанных с проектом;
- фиксация поставок материалов на склад;
- генерация отчётов по остаткам и расходу материалов в формате *.xlsx.

При запуске приложения «Подсистема учёта дорожно-строительных материалов» отображается начальная страница, на которой присутствует

возможность авторизации. После авторизации открывается главное окно приложения.

Инженер может добавлять и редактировать информацию о материалах, контролировать остатки и учёт поставок, просматривать и редактировать информацию о материалах в проектах, формировать и просматривать отчётность.

Администратор имеет те же возможности что и инженер, а также возможность добавлять или удалять информацию о пользователях.

Гость может только просматривать каталог материалов и список проектов.

На рисунке 1 изображена диаграмма вариантов использования приложения.

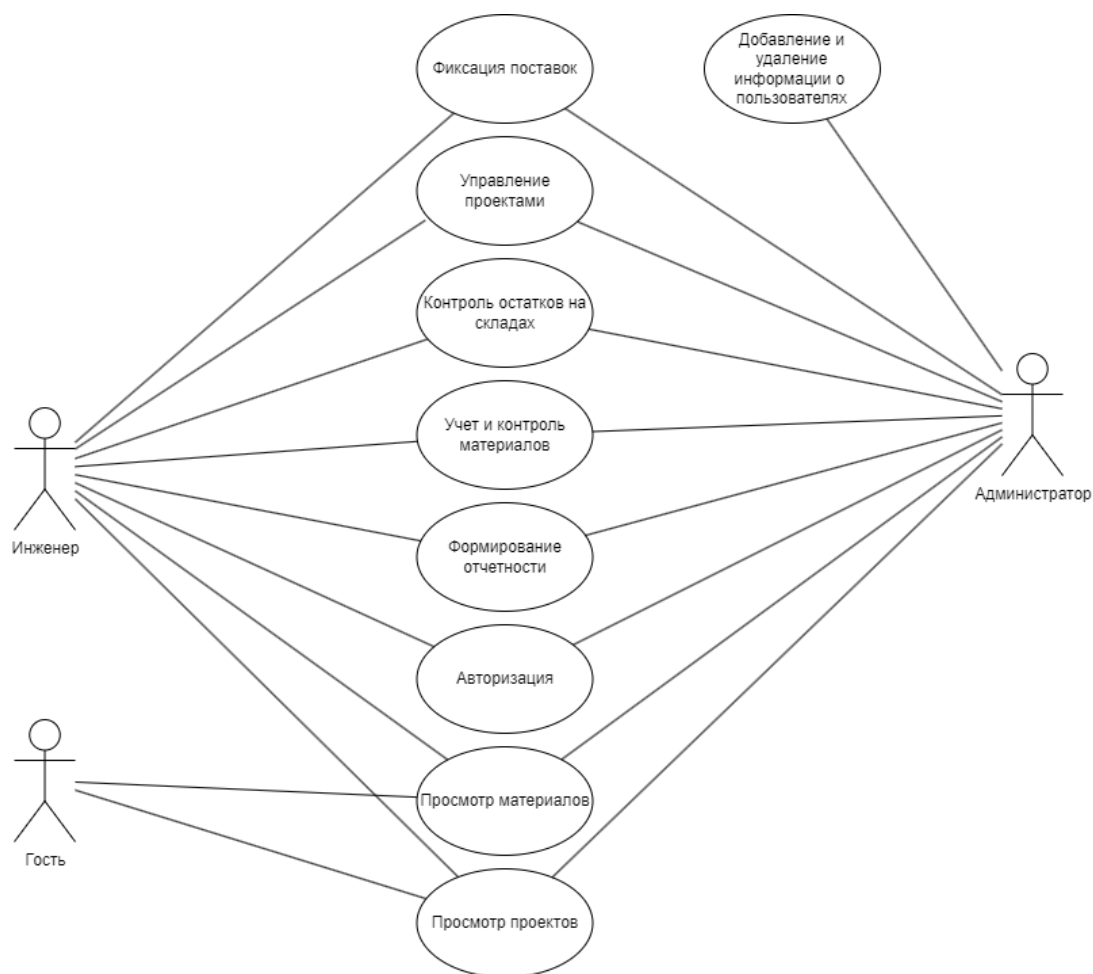


Рисунок 1 – Диаграмма вариантов использования

1.3 Выбор состава программных и технических средств

Требуется создать оконное приложение для учёта дорожно-строительных материалов.

Работа с приложением будет осуществляться на персональных компьютерах и ноутбуках с установленной ОС Windows 10 или новее.

В качестве СУБД выбрана Microsoft SQL Server 2022. Эта СУБД выбрана за её надёжность, высокую производительность и удобство в использовании. MSSQL обеспечивает безопасность данных, поддерживает резервное копирование и восстановление, а также легко интегрируется с приложениями на платформе .NET.

Приложение будет написано на языке программирования C# с использованием фреймворка WPF, который предоставляет удобные инструменты для создания современных оконных приложений, позволяет разрабатывать гибкие и динамичные интерфейсы с использованием XAML, поддерживает привязку данных и визуальное проектирование в IDE Visual Studio 2022. Выбор IDE обусловлен её широкофункциональными инструментами для отладки, профилирования и удобного предпросмотра интерфейсов.

Для функционирования системы на стороне сервера достаточны следующие программные и технические средства:

- ОС Windows 10 и выше;
- сервер: SQL Server 2022;
- процессор x64 с частотой 2 ГГц или выше (рекомендуется 4 ядра);
- оперативная память объемом 4 ГБ;
- место на диске минимум 15 ГБ.

Для функционирования системы на стороне клиента достаточны следующие программные и технические средства:

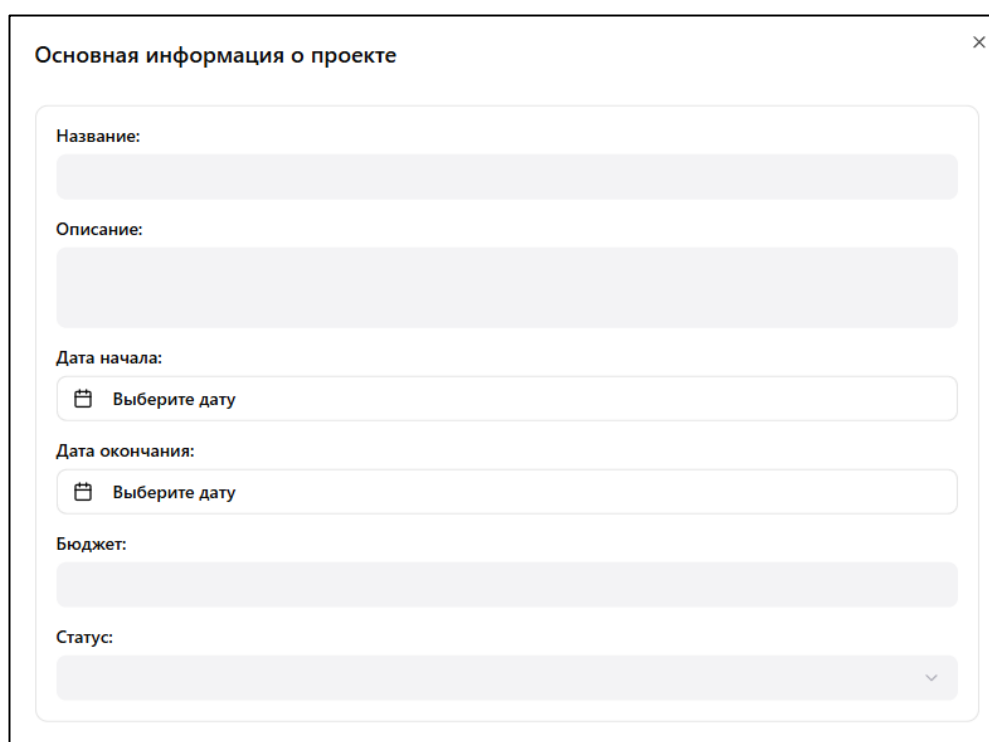
- ОС Windows 10 и выше;
- .NET 8 Desktop Runtime;

- процессор с частотой 1,6 ГГц или выше,
- оперативная память в объеме 4 ГБ (рекомендуется 8 ГБ для повышения производительности);
- свободное место в хранилище 500 МБ.

2 Проектирование ПО

2.1 Проектирование интерфейса пользователя

В ходе проектирования оконного приложения разработаны wireframe [4] интерфейсов пользователя для следующих страниц: добавление основной информации о дорожном проекте (представлен на рисунке 2), добавление материала (представлен на рисунке 3) и просмотра материалов, используемых в проекте (представлен на рисунке 4). Наличие визуальных прототипов позволяет более четко определить структуру взаимодействия с системой и способствует согласованности между этапами проектирования и реализации. Это существенно ускоряет разработку визуальной части ПО.



The wireframe shows a window titled "Основная информация о проекте" (Basic project information) with a close button (X) in the top right corner. The form contains the following fields:

- Название:** A single-line text input field.
- Описание:** A multi-line text input field.
- Дата начала:** A date selection field with a calendar icon and the text "Выберите дату" (Select a date).
- Дата окончания:** A date selection field with a calendar icon and the text "Выберите дату" (Select a date).
- Бюджет:** A single-line text input field.
- Статус:** A dropdown menu with a downward arrow icon.

Рисунок 2 – Wireframe для страницы добавления основной информации о дорожном проекте

Информация о материале

×

Название:

Единица измерения:

Стоимость за единицу:

Поставщик:

Текущий запас:

Отмена

Сохранить

Рисунок 3 – Wireframe для страницы добавления материала

Материалы проекта

×

Добавить

Удалить

Материал	Планируемое количество	Использованное количество
----------	------------------------	---------------------------

Закреть

Рисунок 4 – Wireframe для просмотра материалов, используемых в проекте

В процессе проектирования приложения для учета строительных материалов разработана и применена следующая цветовая палитра: #4361EE в качестве основного цвета, которым оформлены заголовки и кнопки, что позволяет акцентировать внимание пользователей на ключевых элементах управления.; #F5F5F5 в качестве вторичного цвета, который применяется для фона полей ввода и обеспечивает необходимый контраст с основным цветом и текстом; #FFFFFF в качестве цвета для текста на темном фоне, который обеспечивает максимальную читаемость и контрастность.

Для визуального оформления приложения выбраны иконки из библиотеки Google Material Icons, что обеспечивает единый и современный визуальный стиль интерфейса.

Логотип в виде дорожного катка отображает основную тематику программы, связанную со строительством и дорожными работами (представлен на рисунке 5).



Рисунок 5 – Логотип приложения

2.2 Разработка архитектуры ПО

Подсистема построена на основе клиент-серверной модели [5] и рассчитана на работу в корпоративной сети. Ее компоненты размещены на следующих физических устройствах: клиентских рабочих станциях и отдельном сервере БД. Диаграмма развертывания подсистемы показана на рисунке 6

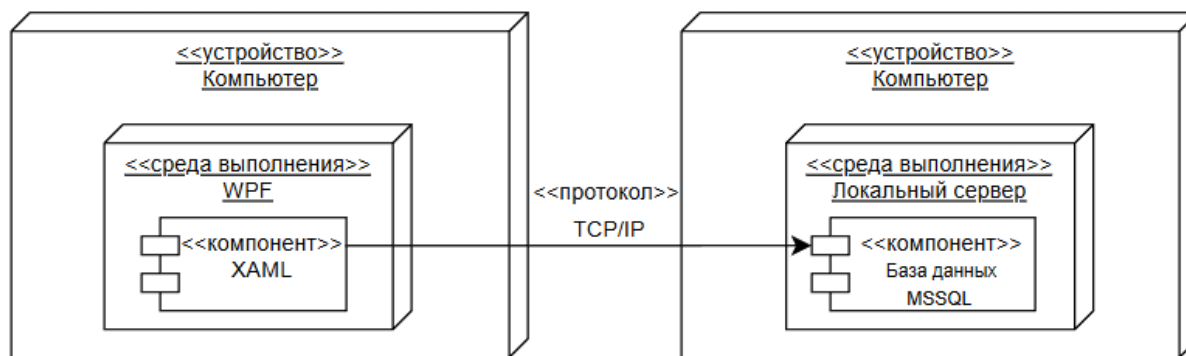


Рисунок 6 – Диаграмма развертывания

2.3 Проектирование БД

Подсистема учета строительных материалов представляет собой комплексное решение для эффективного управления данными, связанными со строительными материалами, их поставками, проектами, поставщиками и пользователями.

База данных системы содержит разнообразную информацию, необходимую для полноценного учета и контроля. Она включает в себя данные о характеристиках строительных материалов, такие как название, единица измерения, стоимость за единицу, текущий запас и поставщик. Кроме того, в базе данных хранятся данные о поставщиках, включая контактную информацию, что позволяет быстро получать необходимые сведения для взаимодействия. Также система содержит параметры строительных проектов, включая описание, сроки начала и окончания, бюджет и статус выполнения, что позволяет эффективно планировать и контролировать выполнение проектов.

Физическая модель БД [3], разработанная для СУБД Microsoft SQL Server 2022, изображена на рисунке 7.

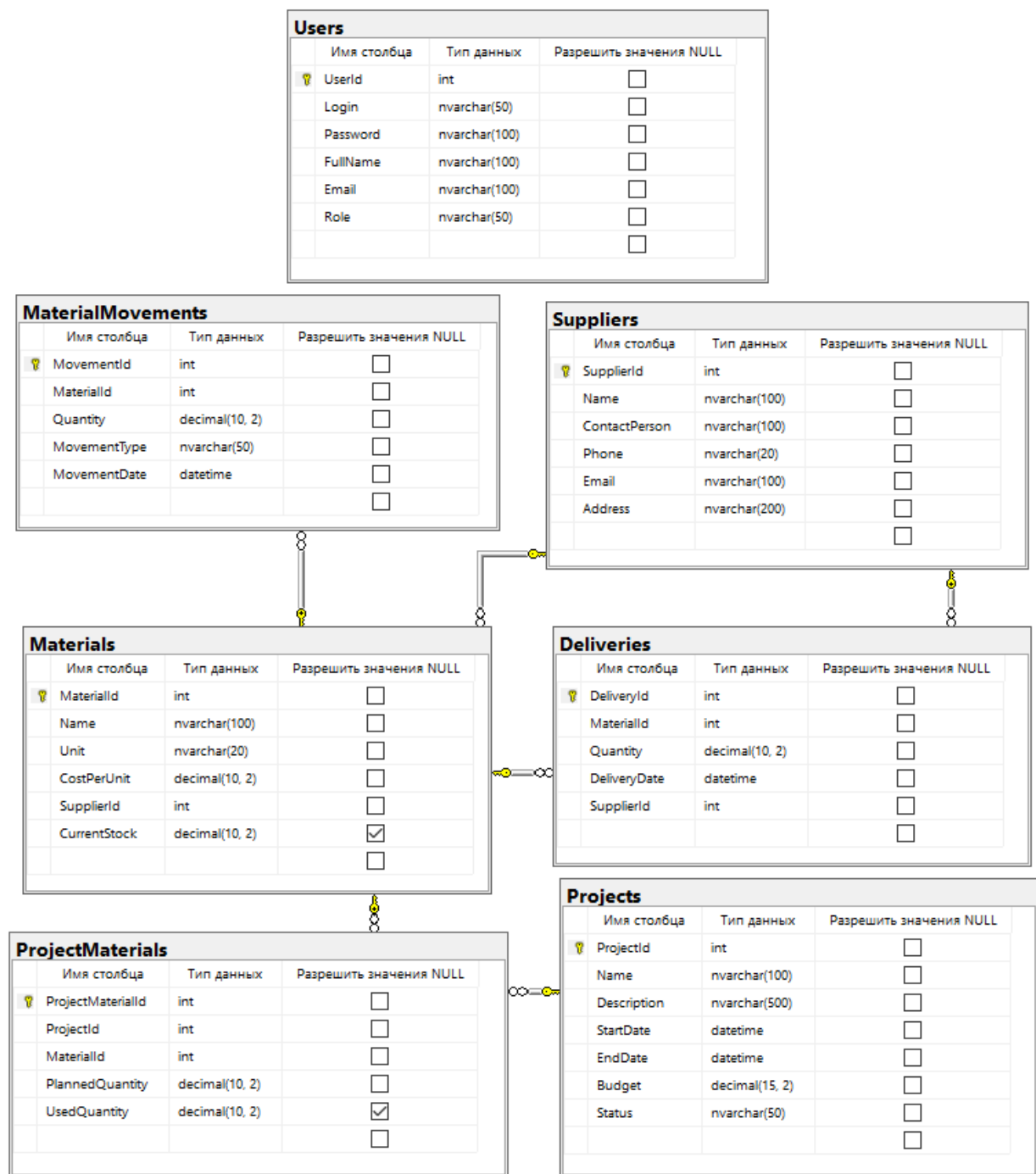


Рисунок 7 – Физическая модель БД

3 Разработка и интеграция модулей программного обеспечения

3.1 Разработка программных модулей

В рамках курсового проекта разработано оконное приложение на языке программирования C# с использованием архитектурного паттерна code-behind [2]. Приложение предназначено для учета строительных материалов и управления проектами.

Для реализации получения данных из БД использован Entity Framework Core. Код метода получения данных о материалах представлен листингом 1.

Листинг 1 – Код метода для получения списка материалов из БД

```
/// <summary>
/// Возвращает список всех материалов с информацией о
поставщиках.
/// </summary>
/// <returns>Список объектов Material с заполненными данными о
поставщиках.</returns>
public List<Material> GetMaterials()
{
    try
    {
        // Используем Include для загрузки связанных данных о
поставщиках
        var materials = _context.Materials.Include(m =>
m.Supplier).ToList();

        if (materials == null)
            return null; // Возвращаем null, если данные не
найденны

        return materials; // Возвращаем список материалов
    }
    catch (DbUpdateException)
    {
        // В случае ошибки БД пробрасываем исключение дальше
// для обработки на более высоком уровне
        throw;
    }
}
```

Одной из ключевых сущностей системы является модель Material, которая описывает строительные материалы, их характеристики и связи с другими сущностями. Реализация модели представлен листингом 2.

Листинг 2 – Модель Material

```
/// <summary>
/// Модель материала для таблицы Materials
/// </summary>
public partial class Material
{
    public int MaterialId { get; set; }           // Первичный
    ключ
    public string Name { get; set; } = null!;    // Наименование
    материала
    public string Unit { get; set; } = null!;    // Единица
    измерения
    public decimal CostPerUnit { get; set; }     // Стоимость за
    единицу
    public int SupplierId { get; set; }          // Внешний ключ
    к поставщику
    public decimal? CurrentStock { get; set; }   // Текущий запас

    // Навигационные свойства
    public virtual ICollection<Delivery> Deliveries { get; set; }
} = new List<Delivery>();
    public virtual ICollection<MaterialMovement>
MaterialMovements { get; set; } = new List<MaterialMovement>();
    public virtual ICollection<ProjectMaterial> ProjectMaterials
{ get; set; } = new List<ProjectMaterial>();
    public virtual Supplier Supplier { get; set; } = null!;
}
```

Для корректной работы с БД выполнена настройка модели Material в классе ApplicationDbContext. Эта настройка определяет параметры хранения данных, ограничения и связи между таблицами. Реализация представлен листингом 3.

Листинг 3 – Настройка модели Material в ApplicationDbContext

```
/// <summary>
/// Настройка сущности Material для отображения в БД
```



```

/// </summary>
protected override void OnModelCreating(ModelBuilder
modelBuilder)
{
    modelBuilder.Entity<Material>(entity =>
    {
        // Установка первичного ключа с именем ограничения
        entity.HasKey(e =>
e.MaterialId).HasName("PK__Material__C50610F7D11276D9");

        // Настройка свойств с указанием типов данных и
ограничений
        entity.Property(e =>
e.CostPerUnit).HasColumnType("decimal(10, 2)"); // Точность до 2
знаков после запятой
        entity.Property(e => e.CurrentStock)
            .HasDefaultValue(0m) // Значение по
умолчанию
            .HasColumnType("decimal(10, 2)"); // Точность до 2
знаков после запятой
        entity.Property(e => e.Name).HasMaxLength(100); //
Ограничение длины названия
        entity.Property(e => e.Unit).HasMaxLength(20); //
Ограничение длины единицы измерения

        // Настройка связи с таблицей Suppliers (поставщики)
        entity.HasOne(d => d.Supplier) // Один
поставщик
            .WithMany(p => p.Materials) // Многие
материалы
            .HasForeignKey(d => d.SupplierId) // Внешний
ключ
            .OnDelete(DeleteBehavior.ClientSetNull) //
Поведение при удалении поставщика

        .HasConstraintName("FK__Materials__Suppl__403A8C7D"); // Имя
ограничения
    });

    // Вызов частичного метода для дополнительной настройки
модели
    OnModelCreatingPartial(modelBuilder);
}

```

Для реализации добавления данных о поставках разработан метод, который добавляет новую запись в БД. Код метода добавления поставки представлен листингом 4.

Листинг 4 – Код метода для добавления поставки

```

/// <summary>
/// Добавляет новую поставку в БД и регистрирует движение
материала.
/// </summary>
/// <param name="delivery">Объект поставки, содержащий
информацию о материале, количестве и поставщике.</param>
public void AddDelivery(Delivery delivery)
{
    try
    {
        // Добавляем новую поставку в контекст БД
        _context.Deliveries.Add(delivery);
        _context.SaveChanges(); // Сохраняем изменения в БД
        // Создаем запись о движении материала при его
поступлении
        var movement = new MaterialMovement
        {
            MaterialId = delivery.MaterialId,
            Quantity = delivery.Quantity,
            MovementDate = DateTime.Now,
            MovementType = "Поступление" // Тип движения -
поступление на склад
        };
        // Добавляем движение материала в БД
        _context.MaterialMovements.Add(movement);
        _context.SaveChanges(); // Сохраняем изменения
    }
    catch (DbUpdateException)
    {
        // При возникновении ошибки БД пробрасываем исключение
дальше
        throw;
    }
}

```

3.2 Реализация интерфейса пользователя

Интерфейс разработан с использованием фреймворка WPF, что позволяет пользователю легко перемещаться между различными разделами приложения. Для реализации интерфейса управления строительными материалами разработана страница `MaterialsPage` с интуитивно понятной системой фильтрации данных. Основное внимание уделено созданию эффективного механизма поиска и сортировки материалов. Код реализации панели фильтрации материалов представлен листингом 5.

Листинг 5 – Код панели фильтрации материалов на странице MaterialsPage

```
<!-- Контейнер панели фильтрации с визуальным оформлением -->
<Border Grid.Row="1" BorderBrush="#E0E0E0" BorderThickness="1"
Padding="10" Margin="0,0,0,10">
    <StackPanel Orientation="Horizontal">
        <!-- Комбинированное поле поиска с подсказкой -->
        <Grid>
            <TextBox x:Name="MaterialFilterTextBox" Width="250"
TextChanged="MaterialFilterTextBox_TextChanged"
                Margin="0,0,10,0"/>
            <TextBlock x:Name="MaterialFilterWatermark" Text="🔍
Поиск по названию..."
                Margin="5,0,0,0" Foreground="#888"
                VerticalAlignment="Center"
IsHitTestVisible="False"/>
        </Grid>

        <!-- Выпадающий список выбора поставщика -->
        <ComboBox x:Name="SupplierFilter" Width="200"
                DisplayMemberPath="Name"
SelectedValuePath="SupplierId"
SelectionChanged="SupplierFilter_SelectionChanged"/>
    </StackPanel>
</Border>
```

3.3 Разграничение прав доступа пользователей

В приложении реализовано разграничение прав доступа пользователей на основе ролей пользователей, хранящихся в БД.

Код метода проверки прав доступа пользователей представлен листингом 6.

Листинг 6 – Код метода проверки прав доступа пользователей

```
/// <summary>
/// Обновляет видимость элементов интерфейса в зависимости от
роли текущего пользователя.
/// Администраторы видят все кнопки, инженеры – все кроме
удаления, гости – только просмотр.
```

```

/// </summary>
private void UpdateUI()
{
    try
    {
        // Проверяем роль текущего пользователя
        bool isGuest = App.CurrentUser?.Role == "Гость";
        bool isAdmin = App.CurrentUser?.Role == "Администратор";

        // Настраиваем видимость кнопок в зависимости от роли
        AddMaterialBtn.Visibility = isGuest ?
Visibility.Collapsed : Visibility.Visible;
        EditMaterialBtn.Visibility = isGuest ?
Visibility.Collapsed : Visibility.Visible;

        // Кнопка удаления доступна только администраторам
        DeleteMaterialBtn.Visibility = !isAdmin ?
Visibility.Collapsed : Visibility.Visible;
    }
    catch
    {
        // Игнорируем ошибки обновления UI, чтобы не прерывать
        // работу приложения
        // в случае проблем с доступом к свойствам пользователя
    }
}

```

3.4 Экспорт данных

В приложении реализован экспорт данных в формат *.xlsx при помощи библиотеки ClosedXML. Код метода экспорта данных о строительных материалах в Excel представлен листингом 7.

Листинг 7 – Код метода экспорта данных

```

/// <summary>
/// Генерирует Excel-отчет по материалам с использованием
/// библиотеки ClosedXML.
/// Создает лист с заголовками и заполняет его данными о
/// материалах.
/// </summary>
/// <param name="materials">Коллекция материалов для включения в
/// отчет.</param>
/// <param name="filePath">Путь для сохранения сгенерированного
/// файла Excel.</param>

```

```

public void GenerateMaterialsReport(IEnumerable<Material>
materials, string filePath)
{
    // Создаем новую книгу Excel
    using (var workbook = new XLWorkbook())
    {
        // Добавляем лист для материалов
        var worksheet = workbook.Worksheets.Add("Материалы");

        // Задаем заголовки столбцов
        worksheet.Cell(1, 1).Value = "Наименование";
        worksheet.Cell(1, 2).Value = "Ед. изм.";
        worksheet.Cell(1, 3).Value = "Стоимость";
        worksheet.Cell(1, 4).Value = "Поставщик";
        worksheet.Cell(1, 5).Value = "Остаток";

        // Заполняем лист данными о материалах, начиная со
второй строки
        int row = 2;
        foreach (var material in materials)
        {
            worksheet.Cell(row, 1).Value = material.Name;
            worksheet.Cell(row, 2).Value = material.Unit;
            worksheet.Cell(row, 3).Value = material.CostPerUnit;

            // Проверяем наличие поставщика, чтобы избежать
NullReferenceException
            worksheet.Cell(row, 4).Value =
material.Supplier?.Name ?? "Нет поставщика";
            worksheet.Cell(row, 5).Value =
material.CurrentStock;

            row++; // Переходим к следующей строке
        }

        // Сохраняем книгу в указанный файл
        workbook.SaveAs(filePath);
    }
}

```

4 Тестирование и отладка программного обеспечения

4.1 Структурное тестирование

В процессе разработки курсового проекта проведено структурное тестирование [1] ключевых компонентов приложения для учета строительных материалов. Для тестирования использовалась библиотека xUnit. Код для тестирования метода AddMaterial представлен листингом 8.

Результат тестирования представлен в рисунке 8.

Листинг 8 – Код тестирования сервиса работы с материалами

```
[Fact]
public void AddMaterial_ShouldAddMaterialToDatabase()
{
    // Создаем новый объект материала с заданными свойствами
    var material = new Material
    {
        Name = [translate:"Тестовый материал"],
        Unit = [translate:"шт"],
        CostPerUnit = 100.50m,
        SupplierId = 1,
        CurrentStock = 50
    };

    // Добавляем материал в контекст БД и сохраняем изменения
    _context.Materials.Add(material);
    _context.SaveChanges();

    // Проверяем, что материал успешно добавлен и его свойства
    // совпадают с ожидаемыми
    var materialInDb = _context.Materials.FirstOrDefault(m =>
        m.Name == [translate:"Тестовый материал"]);
    Assert.NotNull(materialInDb);
    Assert.Equal([translate:"Тестовый материал"],
        materialInDb.Name);
    Assert.Equal(100.50m, materialInDb.CostPerUnit);
    Assert.Equal(1, materialInDb.SupplierId);
    Assert.Equal(50, materialInDb.CurrentStock);
}
```

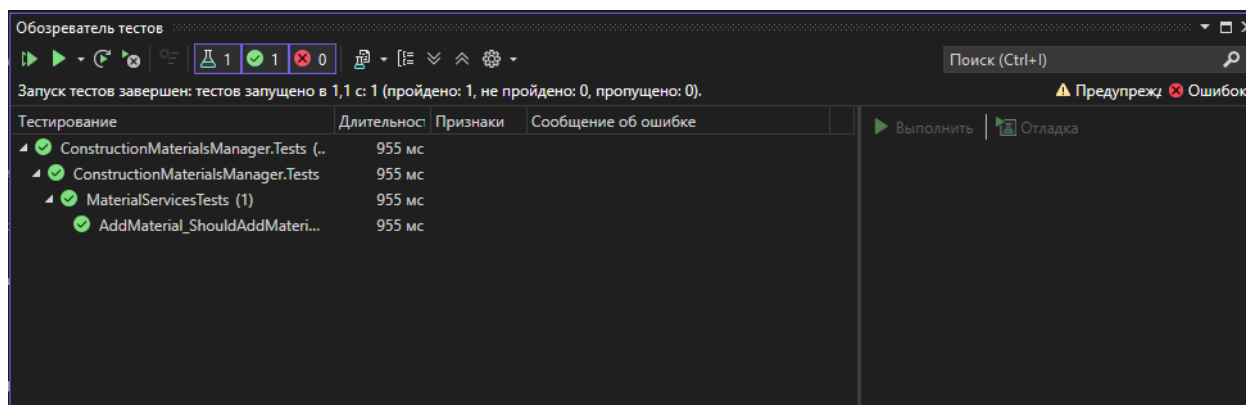


Рисунок 8 – Visual Studio. Вид окна «Обозреватель тестов»

4.2 Функциональное тестирование

Во время курсового проектирования проведено функциональное тестирование методом «чёрного ящика», результаты тестирования представлены в таблице 1.

Таблица 1 – Набор тестов

Действие	Ожидаемый результат	Фактический результат
Удаление материала	Материал исчезает из списка	Совпадает с ожидаемым
Добавление нового материала с корректными данными	Материал появляется в списке материалов	Совпадает с ожидаемым
Редактирование существующего материала	Изменения сохраняются и отображаются в списке	Совпадает с ожидаемым
Фильтрация материалов по поставщику	Отображаются только материалы выбранного поставщика	Совпадает с ожидаемым
Генерация отчета по материалам	Создание и сохранение Excel-файла с данными	Совпадает с ожидаемым

После проведения тестирования можно сделать вывод: приложение работает корректно и все функции выполняются правильно.

5 Инструкция по эксплуатации программного обеспечения

5.1 Установка программного обеспечения

Для функционирования системы на стороне сервера достаточны следующие программные и технические средства:

- ОС Windows 10 и выше;
- сервер: SQL Server 2022;
- процессор x64 с частотой 2 ГГц или выше (рекомендуется 4 ядра);
- оперативная память объемом 4 ГБ;
- место на диске минимум 15 ГБ.

Для развёртывания БД нужно установить MSSQL 2022 на сервер, подключиться к серверу БД с помощью SQL Server Management Studio, вставить и выполнить SQL-скрипт из репозитория проекта.

Для функционирования системы на стороне клиента достаточны следующие программные и технические средства:

- ОС Windows 10 и выше;
- .NET 8 Desktop Runtime;
- процессор с частотой 1,6 ГГц или выше,
- оперативная память в объеме 4 ГБ (рекомендуется 8 ГБ для повышения производительности);
- свободное место в хранилище 500 МБ.

Для установки оконного приложения требуется распаковать архив ConstructionMaterialsManager.zip из репозитория и разархивировать его в любую удобную папку.

В приложении используются следующие учётные данные:

- логин: ivanov_ii,
- пароль: securepass123.

5.2 Инструкция по работе

При запуске приложения, пользователя встречает окно авторизации. Для авторизации требуется ввести учетные данные в поля ввода логина и пароля и нажать кнопку «Войти». Окно авторизации представлено на рисунке 9.

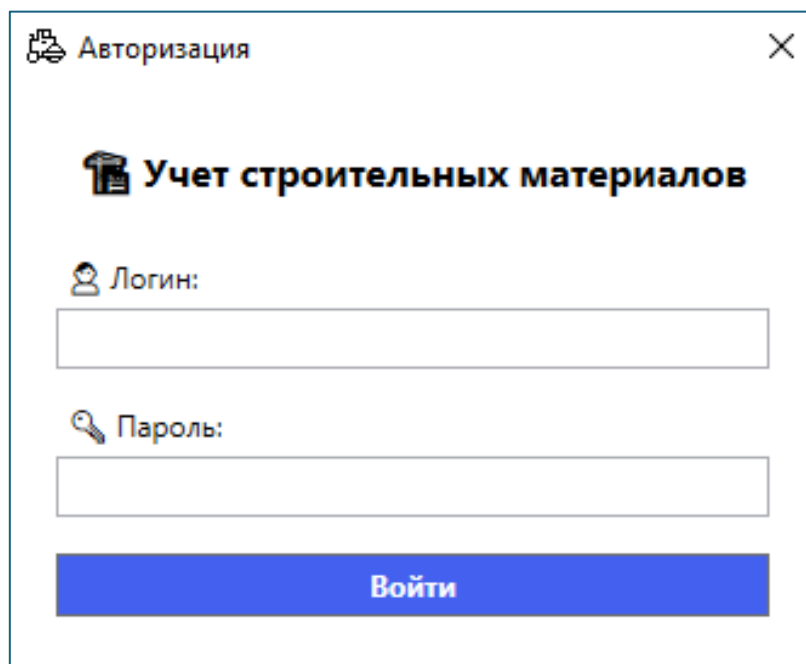


Рисунок 9 – ConstructionMaterialsManager. Вид окна «Авторизация»

После успешной авторизации пользователь попадает на главную страницу приложения, где доступны основные функции управления материалами, проектами, поставщиками, поставками и отчетами. По умолчанию открывается страница «Материалы», где отображается список всех материалов, доступных в системе. Пользователь может добавить новый материал, нажав на кнопку «Добавить материал», или отредактировать существующий, нажав на кнопку «Редактировать». Главное окно представлено на рисунке 10.

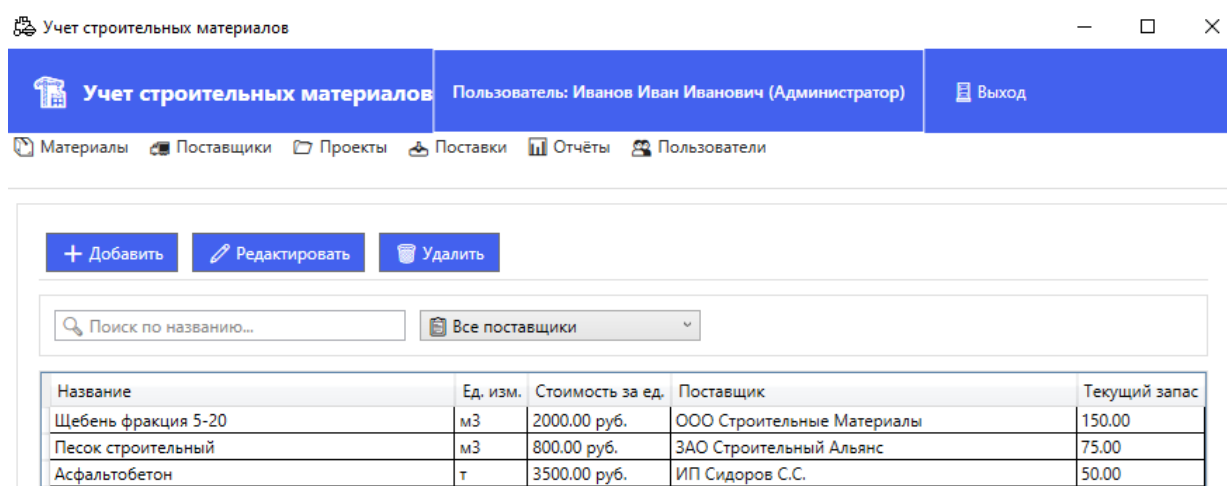


Рисунок 10 – ConstructionMaterialsManager. Главное окно

На странице «Проекты» пользователь может управлять дорожно-строительными проектами: просматривать список проектов, добавлять новые проекты, редактировать и удалять существующие. Страница «Проекты» представлена рисунком 11.

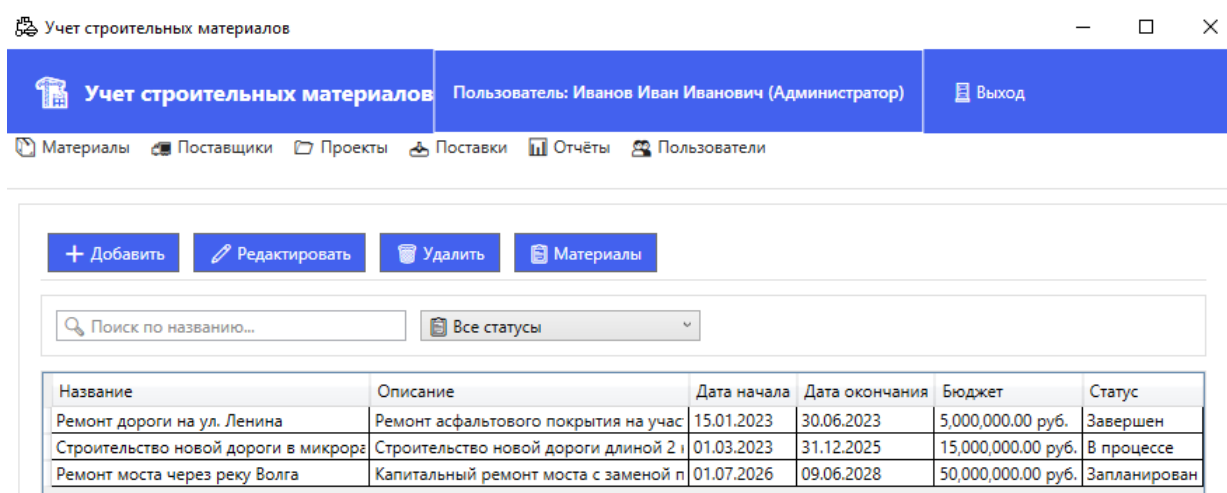


Рисунок 11 – ConstructionMaterialsManager. Страница «Проекты»

На странице «Поставщики» пользователь может управлять информацией о поставщиках: просматривать список, добавлять новых

поставщиков, редактировать и удалять существующих. Страница «Поставщики» представлена рисунком 12.

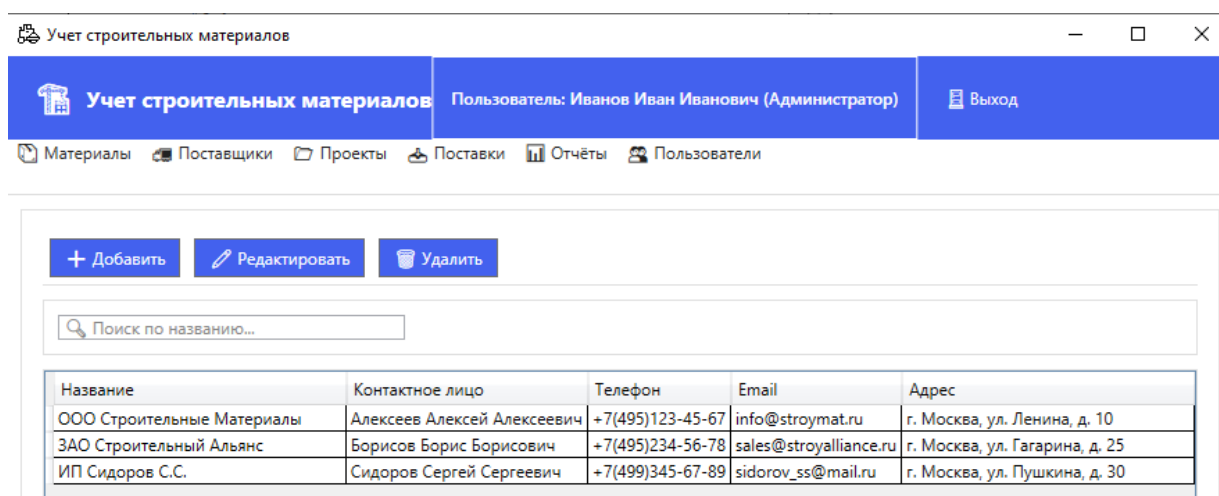


Рисунок 12 – ConstructionMaterialsManager. Страница «Поставщики»

На странице «Поставки» пользователь может управлять информацией о поставках: просматривать список поставок, добавлять новые поставки, редактировать и удалять существующие. Страница «Поставки» представлена рисунком 13.

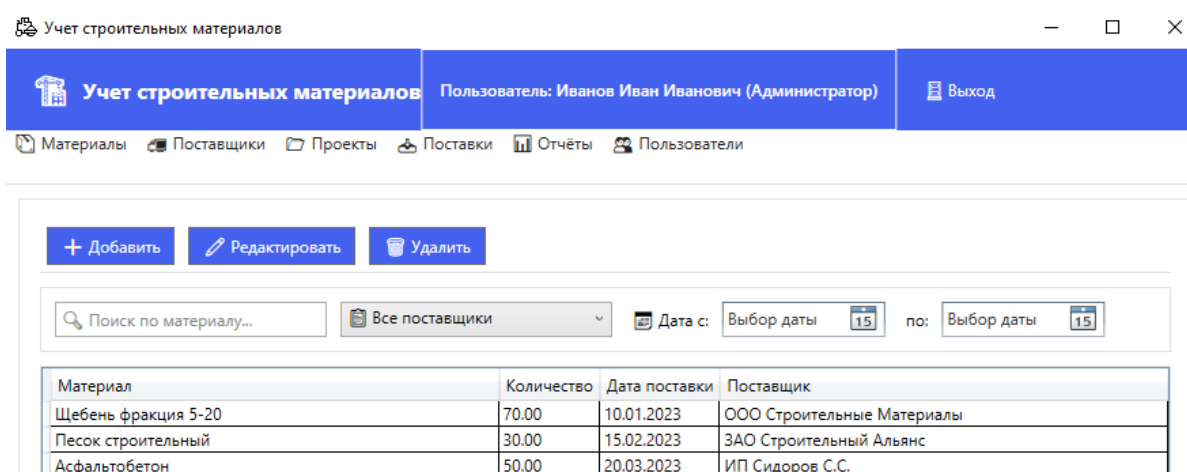


Рисунок 13 – ConstructionMaterialsManager. Страница «Поставки»

На странице «Отчёты» пользователь может просматривать сводную информацию о проектах, материалах и поставках, а также экспортировать отчеты в удобные форматы для дальнейшего анализа. Страница «Отчёты» представлена рисунком 14.

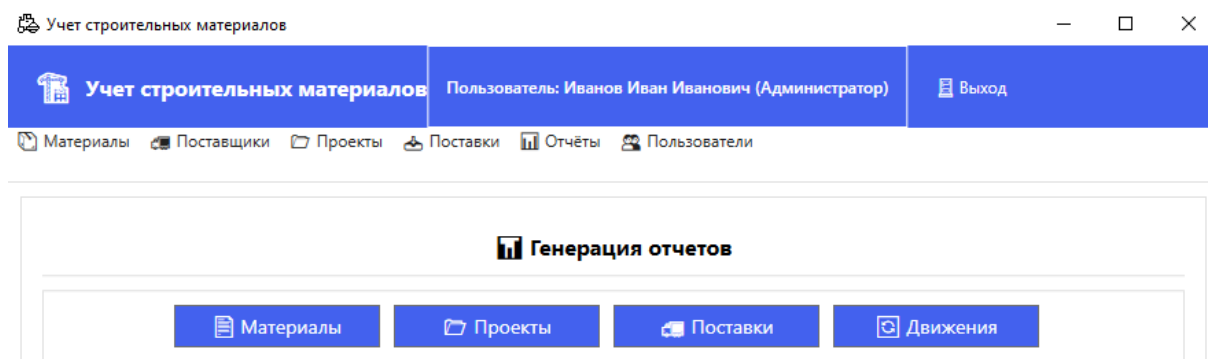


Рисунок 14 – ConstructionMaterialsManager. Страница «Отчёты»

ЗАКЛЮЧЕНИЕ

В ходе курсового проектирования достигнута поставленная цель: разработана система учета строительных материалов, обеспечивающая ведение точного учета запасов, контроль движения материалов и генерацию отчетов о наличии и расходе. Также, решены все поставленные задачи:

- проведён сбор требований целевой аудитории;
- проанализированы информационные источники по предметной области;
- спроектирована архитектура приложения;
- спроектирована диаграмма вариантов использования приложения;
- выбран состав программных и технических средств для реализации приложения;
- спроектирована БД;
- спроектирован интерфейс оконного приложения;
- создана БД в выбранной СУБД;
- разработан интерфейс оконного приложения;
- реализовано разграничение прав доступа пользователей;
- разработано оконное приложение;
- реализован экспорт данных в виде файлов Excel;
- выполнено структурное тестирование ПО;
- выполнено функциональное тестирование ПО;
- разработана программная и эксплуатационная документация.

В результате выполнения поставленных задач разработано оконное приложение для учета и контроля строительных материалов, обеспечивающее эффективный учет запасов и генерацию отчетов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бек, К. Экстремальное программирование: разработка через тестирование. – Санкт-Петербург : Питер, 2021. – 224 с. – URL: <https://ibooks.ru/bookshelf/376974/reading> (дата обращения: 30.11.2025). – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

2. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Сидорова-Виснадул ; под ред. Л. Г. Гагариной. – Москва : ФОРУМ : ИНФРА-М, 2025. – 400 с. – URL: <https://znanium.ru/catalog/product/2178802> (дата обращения: 13.11.2025). – Режим доступа: по подписке. – Текст : электронный.

3. Мартишин, С. А. Базы данных. Практическое применение СУБД SQL и NoSQL-типа для проектирования информационных систем : учебное пособие / С. А. Мартишин, В. Л. Симонов, М. В. Храпченко. – Москва : ФОРУМ : ИНФРА-М, 2024. – 368 с. – URL: <https://znanium.ru/catalog/product/2096940> (дата обращения: 6.11.2025). – Режим доступа: по подписке. – Текст : электронный.

4. Тидвелл, Д. Разработка интерфейсов. Паттерны проектирования. 3-е изд. – Санкт-Петербург : Питер, 2022. – 560 с. – URL: <https://ibooks.ru/bookshelf/386796/reading> (дата обращения: 4.11.2025). – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

5. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности : учебное пособие. – Москва : КУРС : ИНФРА-М, 2024. – 336 с. – URL: <https://znanium.ru/catalog/product/2083407> (дата обращения: 16.11.2025). – Режим доступа: по подписке. – Текст : электронный.