**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)**

Допущен к защите

Зав. отделением

**\_\_\_\_\_\_\_\_\_\_\_\_\_** Ю.В. Солодкая

(Подпись) (И.О. Фамилия)

« 05 » июня 2024 г.

**ДИПЛОМНЫЙ ПРОЕКТ**

НА ТЕМУ

|  |
| --- |
| РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ |
| ДЛЯ ООО «КОСМОДОМ» |
| Л109. 24ДП00. 010 ПЗ |

(Обозначение документа)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Рецензент |  | |  | 05.06.2024 | А.М. Березко |
|  |  | | (Подпись) | (Дата) | (И.О. Фамилия) |
| Руководитель |  | |  | 05.06.2024 | Ю.С. Маломан |
|  |  | | (Подпись) | (Дата) | (И.О. Фамилия) |
| Дипломник |  | |  | 05.06.2024 | А.А. Корхов |
|  | |  | (Подпись) | (Дата) | (И.О. Фамилия) |
| Консультант по оформлению | |  |  | 05.06.2024 | Ю.В. Рубашнева |
|  | | | (Подпись) | (Дата) | (И.О. Фамилия) |
| Консультант по экономической части | | |  | 05.06.2024 | С.Н. Короткова |
|  | | | (Подпись) | (Дата) | (И.О. Фамилия) |

Архангельск 2024

Дипломный проект написан мною самостоятельно и не содержит неправомерных заимствований

*(дата) (подпись дипломника)*

Коэффициент оригинальности ДП %

Проверил:

*(ФИО руководителя ДП)*

*(дата) (подпись)*

**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)**

**О Т З Ы В**

**РЕЦЕНЗЕНТА**

**О ДИПЛОМНОМ ПРОЕКТЕ**

Студент Корхов Артём Анатольевич

(Ф И О)

Специальность 09.02.07 Информационные системы и программирование

Тема ДП Разработка информационной системы для ООО «Космодом»

Рецензент Березко Андрей Михайлович, ООО «Космодом», генеральный директор

( Фамилия И.О., место работы, должность)

**ОЦЕНКА ДИПЛОМНОГО ПРОЕКТА**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **№** | Показатели оценки | **Оценка** | | | | |
| **5** | **4** | **3** | **2** | **0\*** |
| Справочно–информационная | 1 | Соответствие представленного материала теме ДП |  |  |  |  |  |
| 2 | Раскрытие актуальности темы проекта |  |  |  |  |  |
| 3 | Уровень использования в проекте современных методов и технологий |  |  |  |  |  |
| 4 | Корректность использования в проекте технико–экономических  расчетов |  |  |  |  |  |
| 5 | Использование различных информационных ресурсов |  |  |  |  |  |
| 6 | Использование последних достижений науки и техники |  |  |  |  |  |
| Творческая | 7 | Ясность, четкость, последовательность и обоснованность изложения |  |  |  |  |  |
| 8 | Способность самостоятельно решать производственные и конструкторские задачи |  |  |  |  |  |
| 9 | Практическая значимость ДП |  |  |  |  |  |
| Оформитель– ская | 10 | Уровень оформления пояснительной записки: |  |  |  |  |  |
| * общий уровень грамотности |  |  |  |  |  |
| * стиль изложения |  |  |  |  |  |
| * качество иллюстративного материала |  |  |  |  |  |
| 11 | Уровень технической грамотности |  |  |  |  |  |
| ИТОГОВАЯ ОЦЕНКА | | |  | | | | |

\* – не оценивается (трудно оценить)

**Отмеченные достоинства:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Отмеченные недостатки:**

**Заключение**:

Дипломный проект Корхова Артёма Анатольевича

(Ф И О)

соответствует требованиям программы подготовки специалистов среднего звена по специальности 09.02.07 Информационные системы и программирование и может быть допущен к защите.

Считаю, что данный дипломный проект заслуживает оценки \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Оценка прописью)

Рецензент « 05 » июня 2024 г.

(Подпись)

**СОДЕРЖАНИЕ**

[Перечень сокращений и обозначений 3](#_Toc168622337)

[Введение 4](#_Toc168622338)

[1 Анализ и разработка требований 6](#_Toc168622339)

[1.1 Назначение и область применения 6](#_Toc168622340)

[1.2 Постановка задачи 6](#_Toc168622341)

[1.3 Описание алгоритма функционирования системы 7](#_Toc168622342)

[1.4 Выбор состава программных и технических средств 9](#_Toc168622343)

[2 Разработка программного обеспечения 12](#_Toc168622344)

[2.1 Разработка интерфейса пользователя 12](#_Toc168622345)

[2.2 Разграничение прав доступа пользователей 13](#_Toc168622346)

[2.3 Разработка программных модулей 15](#_Toc168622347)

[3 Тестирование и отладка программного обеспечения 21](#_Toc168622348)

[3.1 Модульное тестирование 21](#_Toc168622349)

[3.2 Функциональное тестирование 22](#_Toc168622350)

[4 Инструкция по эксплуатации программного обеспечения 25](#_Toc168622351)

[5 Определение затрат на разработку 31](#_Toc168622352)

[6 Охрана труда и техника безопасности при работе на ПК 44](#_Toc168622353)

[6.1 Общие требования охраны труда 44](#_Toc168622354)

[6.2 Требования охраны труда перед началом работы 44](#_Toc168622355)

[6.3 Требования охраны труда во время работы 45](#_Toc168622356)

[6.4 Требования охраны труда по окончании работы 46](#_Toc168622357)

[Заключение 47](#_Toc168622358)

[Список использованных источников 48](#_Toc168622359)

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящем дипломном проекте применяют следующие сокращения и обозначения:

БД – база данных

ИС – информационная система

МВ – машинное время

МРОТ – минимальный размер оплаты труда

ОИС – объект информационной системы

ООО – общество с ограниченной ответственностью

ОС – операционная система

ПК – персональный компьютер

ПО – программное обеспечение

СУБД – система управления базами данных

ЭВМ – электронная вычислительная машина

IDE – интегрированная среда разработки

MSSQL – система управления базами данных

ORM – отображение данных на реальные объекты

SQL – язык структурированных запросов

SSMS – утилита для управления СУБД

UML – унифицированный язык моделирования

VCS – система контроля версий

WPF – платформа для разработки приложений

## ВВЕДЕНИЕ

В настоящее время информационные системы играют важную роль в современном мире, обеспечивая предприятиям возможность автоматизировать процессы, улучшать управление данными и повысить общую эффективность работы. Они являются незаменимым инструментом для решения различных задач, начиная от учета и анализа данных и заканчивая планированием и оптимизацией бизнес–процессов.

Внедрение ИС позволяет компаниям значительно сократить временные и финансовые затраты, повысить точность и достоверность данных, а также улучшить качество предоставляемых услуг.

Актуальность разрабатываемого дипломного проекта заключается в необходимости оптимизации и автоматизации процессов управления и учета в ООО «Космодом». Внедрение разрабатываемой ИС обеспечит:

* более точный и своевременный учет данных о фасадных материалах,
* автоматизацию рутинных задач по расчету необходимого количества фасадных материалов и подбору необходимых комплектующих, что сократит временные затраты сотрудников и уменьшит вероятность появления ошибок в расчетах,
* улучшение качества обслуживания клиентов за счет быстрого доступа к актуальной информации.

Целью дипломного проектирования является разработка ИС для ООО «Космодом» с использованием современных инструментов и технологий.

Для достижения поставленной цели требуется решить следующие задачи:

* проанализировать функциональные и эксплуатационные требования к ПО,
* выработать и оформить требования к разрабатываемым программным модулям,
* спроектировать архитектуру ПО,
* спроектировать и реализовать БД,
* создать таблицы БД и загрузить в них данные,
* реализовать добавление и редактирование данных о фасадных материалах в клиентском приложении,
* реализовать модуль для расчета количества требуемых фасадных материалов,
* реализовать экспорт данных о расчетах,
* реализовать разграничение прав доступа пользователей,
* выполнить интеграцию модулей ПО,
* составить тестовые наборы для разрабатываемого ПО,
* выполнить тестирование и отладку разработанного ПО с использованием специализированных программных средств,
* разработать программную и эксплуатационную документацию.

## 1 Анализ и разработка требований

### 1.1 Назначение и область применения

Разрабатываемая ИС предназначена для автоматизации процесса работы менеджеров с клиентами, что позволит значительно упростить и ускорить выполнение рутинных задач. Основная цель системы – устранение необходимости ручного труда при выполнении расчетов и оформлении документации, что снизит вероятность ошибок и повысит общую производительность.

ПО может быть применено в компаниях, занимающихся продажей строительных материалов. Благодаря использованию разрабатываемой ИС, такие компании смогут оптимизировать процесс взаимодействия с клиентами, обеспечив точность и оперативность в обработке заказов и расчетов. Потенциальными пользователями системы будут являться менеджеры и клиенты компаний по продаже строительных материалов, в частности ООО «Космодом», а также администраторы приложения.

### 1.2 Постановка задачи

Требуется разработать ИС с оконным интерфейсом пользователя, предназначенную для автоматизации расчетов и вычисления необходимого количества фасадных материалов на основе предоставляемых заказчиком данных и хранения информации о доступных к выбору фасадных материалах.

В приложении должны быть следующие функциональные возможности:

* расчет необходимого количества материалов для заказа на основе вводимых данных (ширины и высоты объекта и др.),
* предоставление итоговой информации о расчете в удобном для пользователя виде,
* оформление результатов расчетов в виде файла в формате .txt,
* просмотр и редактирование списка фасадных материалов,
* отображение фотографий имеющихся фасадных материалов.

К эксплуатационным возможностям можно отнести интуитивно–понятный интерфейс пользователя с реализацией разграничения прав доступа. В ИС должно быть предусмотрено ограничение доступа к функциональным возможностям в зависимости от роли пользователя.

Пользователь с ролью администратор должен иметь следующие возможности: возможность добавлять и редактировать информацию о новых пользователях и фасадных материалах, добавлять и редактировать существующие расчеты, сохранять результаты расчета в БД и файл в формате .txt, просматривать список пользователей и редактировать информацию о них.

Пользователь с ролью менеджер должен иметь следующие возможности: возможность создавать и редактировать расчеты, сохранять результаты расчета в БД и файл в формате .txt, просматривать и редактировать список материалов.

Пользователь с ролью клиент должен иметь следующие возможности: возможность создавать расчеты, просматривать список материалов и сохранять результаты расчета в файл в формате .txt.

### 1.3 Описание алгоритма функционирования системы

При запуске системы открывается окно авторизации с возможностью работы неавторизованного пользователя как пользователя с ролью «Клиент». При необходимости пользователи могут войти в систему, используя свой логин и пароль. После успешной авторизации пользователи получают доступ к определенным функциям системы.

На рисунке 1 представлена диаграмма вариантов использования ПО администратором приложения, менеджером и клиентом, разработанная с использованием draw.io [13].

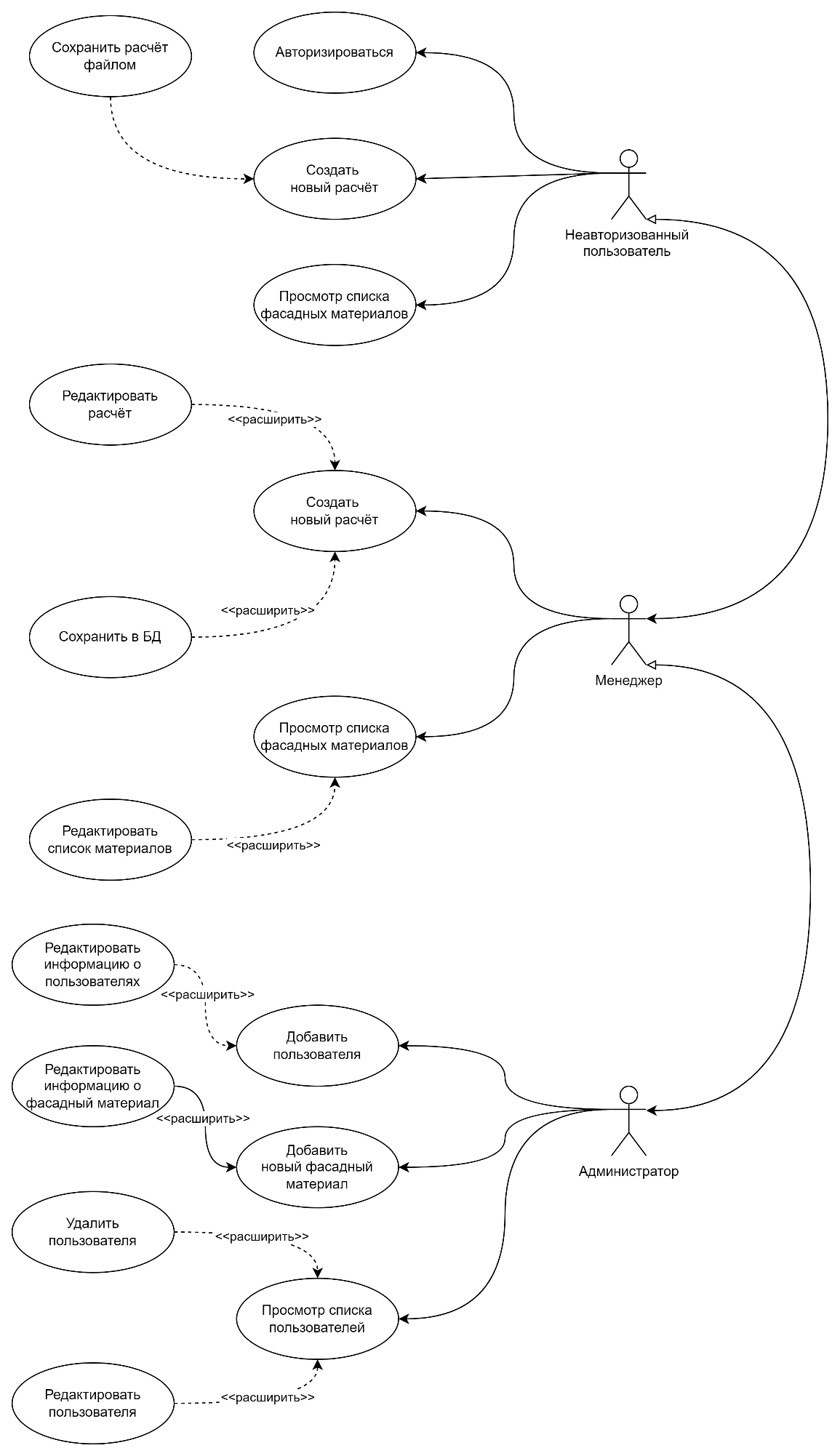


Рисунок 1– Диаграмма вариантов использования

### 1.4 Выбор состава программных и технических средств

Согласно цели проекта требуется разработать информационную систему для ООО «Космодом».

Работа с ИС будет осуществляться на ПК с установленной ОС Windows версии не ниже 10, объединенных в локальную сеть.

В качестве СУБД выбрана MSSQL Developer 2022 [12], так как она обладает возможностью масштабируемости и автоматизации рутинных процессов и задач.

Приложение будет написано на языке программирования С#, так как он является простым, надежным и масштабируемым языком программирования.

Для разработки приложения будет использоваться среда разработки JetBrains Rider 2023. Выбор Rider обусловлен многими преимуществами этой IDE. В частности, Rider предоставляет обширный инструментарий для разработки на C# [19] и обеспечивает высокую производительность и удобство в использовании, что делает его отличным выбором для разработки клиентской части приложения.

Для разработки использованы следующие технологии, библиотеки и пакеты:

* WPF: фреймворк для создания графических пользовательских интерфейсов на платформе Windows [9],
* Entity Framework Core: ORM–фреймворк для работы с БД [16],
* Microsoft.EntityFrameworkCore.SqlServer: библиотека, предоставляющая драйвер для работы с БД MSSQL вместе с Entity Framework Core [18],
* Microsoft.EntityFrameworkCore.Tools,.Microsoft.EntityFrameworkCore.Design: библиотеки, предоставляющие инструменты командной строки для работы с Entity Framework Core [14].

2 Проектирование программного обеспечения

2.1 Проектирование БД

В связи с задачей разработки ИС для расчета фасадных материалов возникла необходимость в создании БД для хранения информации о фасадных материалах, расчетах, работниках и связанных с ними данных [8].

На рисунке 2 выделены таблицы, представляющие собой сущности и соответствующие атрибуты, необходимые для реализации функциональности [6].

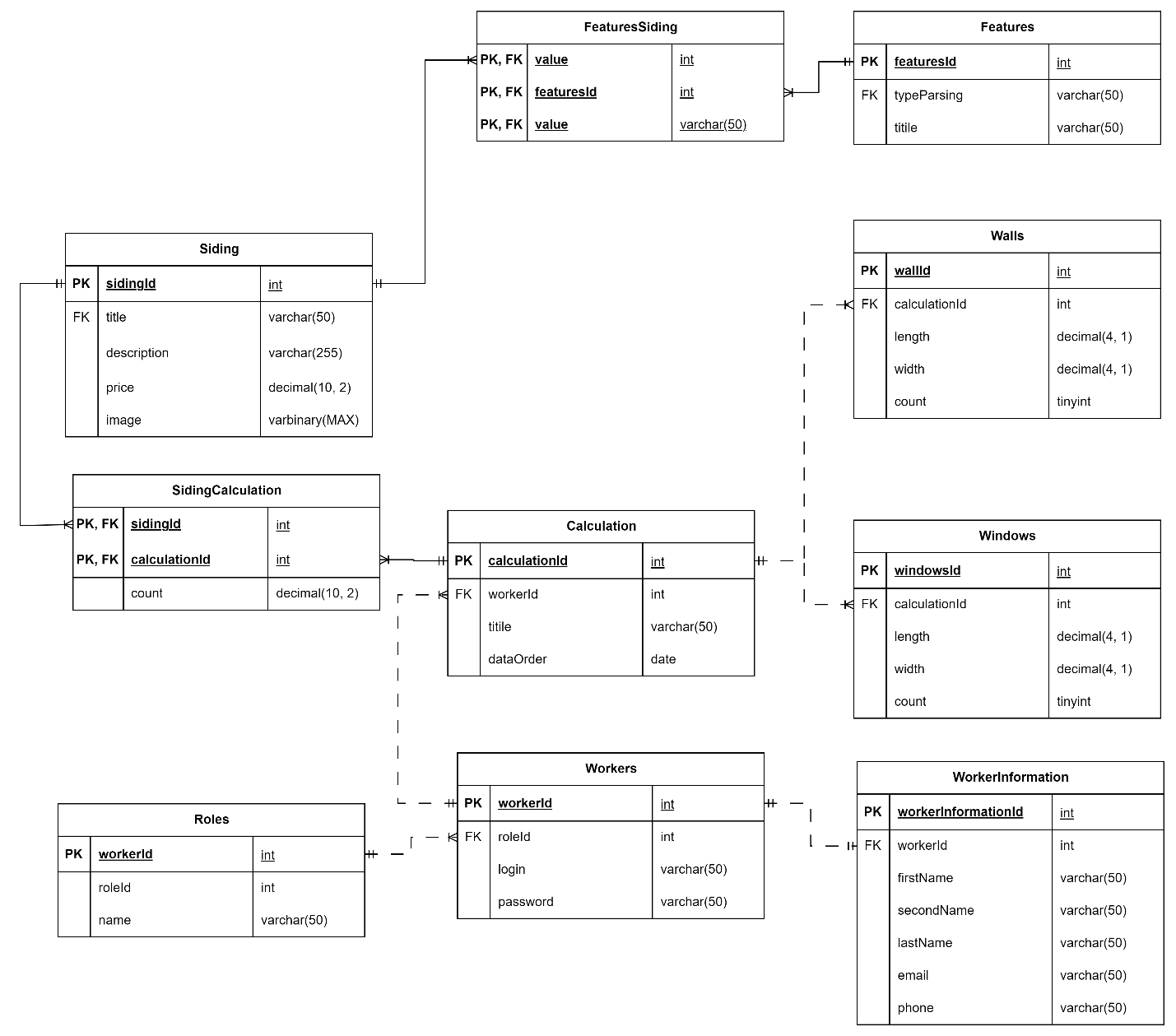


Рисунок 2 – Физическая модель БД

3.2 Разработка архитектуры программного обеспечения

В качестве архитектуры ПО выбрана архитектура «клиент–сервер». Серверная часть будет представлена в виде СУБД MSSQL [20], клиентская – в виде оконного приложения, созданного с использованием технологии WPF.

В ходе проектирования архитектуры ПО создана диаграмма развертывания, которая иллюстрирует взаимодействие двух основных составляющих системы: БД и клиентского приложения. Диаграмма развертывания представлена на рисунке 3 и изображает физическую конфигурацию системы.

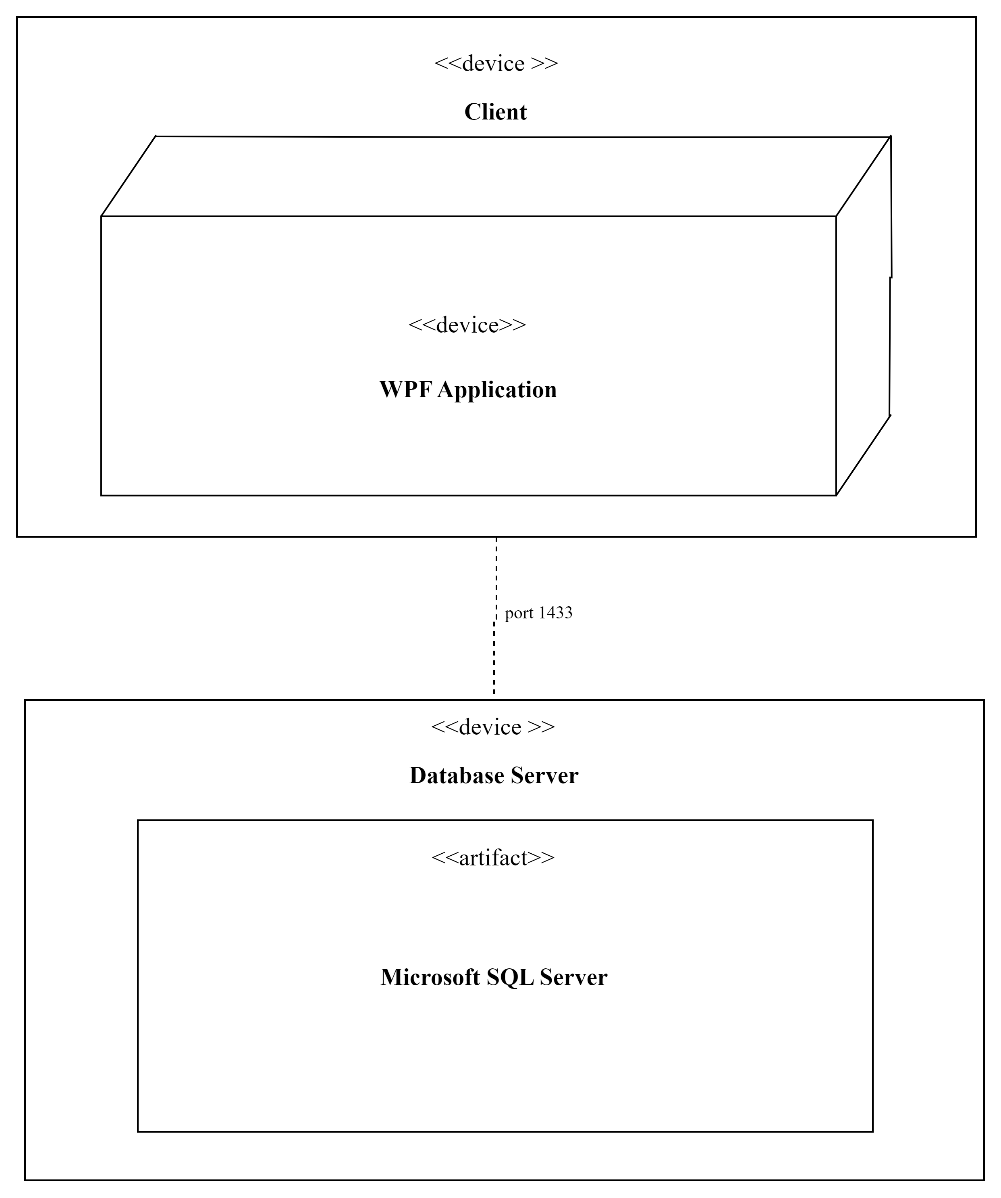


Рисунок 3 – Диаграмма развертывания

## 2 Разработка программного обеспечения

### 2.1 Разработка интерфейса пользователя

В процессе разработки ИС для ООО «Космодом» на платформе .NET применены передовые технологии и современные библиотеки: WPF и MaterialDesign, реализован страничный интерфейс пользователя [17].

Для авторизации и регистрации созданы отдельные окна. Фрагмент кода окна авторизации представлен листингом 1.

Листинг 1 – Фрагмент кода страницы авторизации

|  |
| --- |
| <Grid>  <!–– Определение столбцов ––>  <Grid.ColumnDefinitions>  <ColumnDefinition /> <!–– Первый столбец ––>  <ColumnDefinition /> <!–– Второй столбец ––>  <ColumnDefinition /> <!–– Третий столбец ––>  </Grid.ColumnDefinitions>  <!–– Определение строк ––>  <Grid.RowDefinitions>  <RowDefinition Height="50" /> <!–– Верхний отступ ––>  <RowDefinition /> <!–– Основное содержимое ––>  <RowDefinition Height="50" /> <!–– Нижний отступ ––>  </Grid.RowDefinitions>  <!–– форма авторизаци ––>  <Border Grid.Column="1" Grid.Row="1" Style="{StaticResource CustomBorderStyle}" Height="600">  <Grid>  <!–– Определение строк в рамке ––>  <Grid.RowDefinitions>  <RowDefinition Height="Auto" />  <!–– Верхний отступ ––>  <RowDefinition /> <!–– Основное содержимое ––>  <RowDefinition Height="Auto" />  <!–– Отступ между элементами ––>  <RowDefinition /> <!–– Основное содержимое ––>  <RowDefinition Height="90" />  <!–– Отступ между элементами ––>  <RowDefinition Height="20"/>  <!–– Пространство ––>  <RowDefinition /> <!–– Основное содержимое ––>  </Grid.RowDefinitions>  <!–– Заголовок "Авторизация" ––>  <TextBlock Grid.Column="0" Grid.ColumnSpan="3" Text="Авторизация" HorizontalAlignment="Center"  FontFamily="Montserrat" FontSize="36" />  <!–– Поле для ввода логина ––>  <Label Content="Введите логин" Grid.Row="1" Grid.Column="1" FontFamily="Montserrat" Margin="30,30,0,0" />  <TextBox x:Name="LoginTextBox" Text="manager" Grid.Row="2" Grid.Column="1"  Style="{StaticResource TextBoxStyle}"  Width="500" FontSize="30" Margin="0" Height="70" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" />  <!–– Поле для ввода пароля ––>  <Label Content="Введите пароль" Grid.Row="3" Grid.Column="1" Margin="30,30,0,0" />  <PasswordBox x:Name="PasswordTextBox" Password="manager\_pass" Grid.Row="4" Grid.Column="1"  Style="{StaticResource PasswordBoxStyle}" Width="500" FontSize="30" Margin="0" HorizontalContentAlignment="Center" VerticalContentAlignment="Center" />  <!–– Кнопка для выполнения авторизации ––>  <Button x:Name="AuthorizatiorButton" Content="Войти" Width="360" Grid.Row="6" Grid.Column="1"  HorizontalContentAlignment="Center" VerticalContentAlignment="Center"  Click="AuthorizatiorButton\_OnClick" />  </Grid>  </Border>  </Grid> |

В ходе разработки был создан интуитивно–понятный, эстетичный и высокофункциональный интерфейс пользователя, соответствующий современным стандартам и ожиданиям пользователей.

### 2.2 Разграничение прав доступа пользователей

В целях разграничения прав доступа пользователей в приложении разработан метод, представленный листингом 2, который сверяет введенные данные с уже хранящимися в БД.

Листинг 2 – Код метода «GetAuthorization»

|  |
| --- |
| void GetAuthorization()  {  // Получаем имя пользователя и пароль из текстовых полей на //форме  var username = LoginTextBox.Text;  var password = PasswordTextBox.Password;    // Проверяем, заполнены ли поля имени пользователя и пароля  if (string.IsNullOrEmpty(username) || string.IsNullOrEmpty(password))  {  MessageBox.Show("Пожалуйста, введите имя пользователя и пароль.", "Authentication Error", MessageBoxButton.OK, MessageBoxImage.Error);  return; // Если одно из полей пустое, показываем //сообщение об ошибке и выходим из метода  }  // Проверяем, содержат ли имя пользователя и пароль //начальные или конечные пробелы  if (username.Trim() != username || password.Trim() != password)  {  MessageBox.Show("Имя пользователя и пароль не должны содержать начальных или конечных пробелов.", "Authentication Error", MessageBoxButton.OK, MessageBoxImage.Error);  return; // Если содержат, показываем сообщение об //ошибке и выходим из метода  }  try  {  // Пытаемся найти пользователя в базе данных по логину //и паролю  var user = App.Context.Workers  .FirstOrDefault(u =>  u.Login == username &&  u.Password == password);  // Если пользователь не найден, выбрасываем исключение  if (user == null)  {  throw new Exception();  }    // Устанавливаем текущего пользователя приложения  App.User = user;  // Если пользователь успешно установлен, открываем //новое окно и закрываем текущее  if (App.User != null)  {  new NavigationFrame().Show(); // Открываем новое окно  Close(); // Закрываем текущее окно  }  }  catch (Exception e)  {  // В случае ошибки подключения показываем сообщение об //ошибке  MessageBox.Show("Ошибка подключения. " + e.Message, "Authentication Error", MessageBoxButton.OK, MessageBoxImage.Error);  }  } |

### 2.3 Разработка программных модулей

Для удобства использования ИС добавлены модуль рисования плана здания (представлен листингом 3) и модуль расчета по этому плану необходимого количества фасадных материалов и комплектующих (представлен листингом 4).

Листинг 3 – Код метода «DrawingCanvas»

|  |
| --- |
| private void DrawingCanvas()  {  // Получаем начальную точку, привязываемую к сетке  startPoint = SnapToGrid(e.GetPosition(DrawingCanvas));  // Создаем новую линию с начальной точкой и настройками  currentLine = new Line  {  Stroke = Brushes.Black, // Цвет линии  StrokeThickness = 2, // Толщина линии  X1 = startPoint.X, // Начальная координата X  Y1 = startPoint.Y, // Начальная координата Y  X2 = startPoint.X, // Конечная координата X  Y2 = startPoint.Y // Конечная координата Y  };  // Добавляем линию на холст  DrawingCanvas.Children.Add(currentLine);  // Устанавливаем флаг, что сейчас происходит рисование  isDrawing = true;  // Получаем конечную точку, привязываемую к сетке  Point endPoint = SnapToGrid(e.GetPosition(DrawingCanvas));  // Обновляем конечные координаты линии  currentLine.X2 = endPoint.X;  currentLine.Y2 = endPoint.Y;  // Добавляем линию в список линий  lines.Add(currentLine);  // Вычисляем длину линии в пикселях  double lengthInPixels = Math.Sqrt(Math.Pow(currentLine.X2 – currentLine.X1, 2) + Math.Pow(currentLine.Y2 – currentLine.Y1, 2));  // Преобразуем длину из пикселей в метры  double lengthInMeters = lengthInPixels / cellSize \* cellSideLength;  // Создаем новый объект Walldn с параметрами  Walldn newWall = new Walldn  {  Id = walldn.Count > 0 ? walldn.Max(w => w.Id) + 1 : 1, // Устанавливаем новый ID  Height = 3, // Высота стены  Length = lengthInMeters, // Длина стены в метрах  LineRepresentation = currentLine  // Графическое представление линии  };  // Добавляем новый объект Walldn в список  walldn.Add(newWall);  // Сбрасываем текущую линию  currentLine = null;  // Устанавливаем флаг, что рисование завершено  isDrawing = false;  } |

Листинг 4 – Код метода «CalculateAndDisplayResults»

|  |
| --- |
| private void CalculateAndDisplayResults()  {  double totalArea = 0; // Общая площадь всех стен  double totalLength = 0; // Общая длина всех линий  int internalAngles = 0; // Количество внутренних углов  int externalAngles = 0; // Количество внешних углов  // Подсчитываем общую площадь всех стен  foreach (var wall in walldn)  {  double wallArea = wall.Length \* wall.Height; // Площадь одной стены  // Добавляем площадь стены к общей площади  totalArea += wallArea;  }  // Подсчитываем общую длину всех линий  foreach (var line in lines)  {  double lengthInPixels = Math.Sqrt(Math.Pow(line.X2 – line.X1, 2) + Math.Pow(line.Y2 – line.Y1, 2));  // Длина линии в пикселях  double lengthInMeters = lengthInPixels / cellSize \* cellSideLength; // Перевод длины линии в метры  totalLength += lengthInMeters;  // Добавляем длину линии к общей длине  }  // Подсчитываем количество внутренних и внешних углов, если //линий больше одной  if (lines.Count > 1)  {  for (int i = 0; i < lines.Count; i++)  {  Line line1 = lines[i];  Line line2 = lines[(i + 1) % lines.Count];  // Используем оператор % для замыкания на последнюю линию  // Общая точка двух линий  Point commonPoint = new Point(line1.X2, line1.Y2);  // Конечная точка первой линии  Point line1End = new Point(line1.X1, line1.Y1);  // Конечная точка второй линии  Point line2End = new Point(line2.X2, line2.Y2);  // Вычисляем угол между двумя линиями  double angle = CalculateAngle(line1, line2);  string angleType = ClassifyAngle(commonPoint, line1End, line2End);  // Определяем тип угла (внутренний или внешний)  if (angleType == "Внутренний")  {  // Увеличиваем счетчик внутренних углов  internalAngles++;  }  else  {  // Увеличиваем счетчик внешних углов  externalAngles++;  }  }  }  // Отображаем результаты расчетов  ResultsTextBlock.Text = $"Общая длина: {totalLength:F2} метров\n" + $"Общая площадь: {totalArea:F2} метров квадратных\n" + $"Внутренние углы: {internalAngles}\n" + $"Внешние углы: {externalAngles}";  // Сохраняем состояние переменной walldn в приложении  App.Walldn = walldn;  } |

2.4 Импорт и экспорт данных

Для возможности добавления изображения к материалам реализован метод «AddImage» для импорта изображений в БД. Код метода «AddImage» представлен листингом 5. При вызове этого метода открывается диалоговое окно для выбора изображения. Пользователь может выбрать изображение в поддерживаемых форматах (например, JPG, PNG, GIF).

Указанный метод включает в себя логику для обработки выбранного изображения и предоставляет возможность пользователю успешно добавить изображение в приложение. В случае возникновения ошибок при импорте изображения метод обрабатывает их и сообщает об ошибке пользователю.

Листинг 5 – Код метода «AddImage»

|  |
| --- |
| private void AddImage()  {  try  {  // Создание диалогового окна для выбора изображения.  var dlg = new OpenFileDialog();  dlg.DefaultExt = ".jpg";  dlg.Filter = "Image Files|\*.jpg;\*.png;\*.gif;  // Проверка, выбрано ли изображение  if (dlg.ShowDialog() == true)  {  var imagePath = dlg.FileName;  try  {  // Создание BitmapImage из выбранного пути.  var bitmap = new BitmapImage  (new Uri(imagePath));  //Установка выбранного изображения в источник  SelectedImage.Source = bitmap;  // Чтение изображения в виде массива байтов.  \_imageData = File.ReadAllBytes(imagePath);  }  catch (Exception ex)  {  //Обработка ошибок чтения  MessageBox.Show("Ошибка: " + ex);  }  }  }  catch (Exception ex)  {  // Обработка других ошибок  MessageBox.Show("Ошибка: " + ex);  }  } |

В рамках реализации экспорта данных о результатах вычислений создан метод «ReplaceValuesInFile» (представлен листингом 6), генерирующий файл с расчетами на основе файла–шаблона.

Листинг 6 – Код метода «ReplaceValuesInFile»

|  |
| --- |
| private void ReplaceValuesInFile (Dictionary<string, string> items)  {  try  {  // Получаем базовый каталог текущего приложения  string baseDirectory =  AppDomain.CurrentDomain.BaseDirectory.TrimEnd('\\', '/');  // Определяем относительный путь к ресурсному файлу  string relativePath = "Resources/hablon.txt";  // Сочетаем базовый каталог  string absolutePath = $"{baseDirectory}\\{relativePath}";  // Создаем диалоговое окно для сохранения файла  SaveFileDialog saveFileDialog = new SaveFileDialog();  saveFileDialog.Filter =  "Text files (\*.txt)|\*.txt|All files (\*.\*)|\*.\*";  saveFileDialog.Title = "Сохранение файла";  // Показываем диалоговое окно для сохранения файла  saveFileDialog.ShowDialog();  // выбран ли пользователем допустимый путь к файлу  if (saveFileDialog.FileName != "")  {  // Получаем выбранный путь к файлу  string newFilePath = saveFileDialog.FileName;  // Читаем все строки из исходного файла  string[] fileLines = File.ReadAllLines(absolutePath);  // Проходим по каждой строке в файле  for (int i = 0; i < fileLines.Length; i++)  {  // Проходим по коллекции 'items'  foreach (var item in items)  {  //Проверяем, содержит ли текущая строка 'items’  if (fileLines[i].Contains(item.Key))  {  //Заменяем ключ соответствующим значением  fileLines[i] =  fileLines[i].Replace(item.Key, item.Value);  }  }  }  // Записываем измененные строки в новый путь к файлу  File.WriteAllLines(newFilePath, fileLines);  // Выводим сообщение об успешном выполнении операции  MessageBox.Show("Файл успешно создан.");  }  }  catch (Exception ex)  {  // Если происходит исключение, выводим сообщение об ошибке  MessageBox.Show($"Ошибка: {ex.Message}");  }  } |

## Тестирование и отладка программного обеспечения

### 3.1 Модульное тестирование

Тестирование ПО является неотъемлемой частью жизненного цикла разработки, нацеленной на обеспечение высокого уровня функциональности и надежности приложений. Этот процесс включает систематическое выполнение тестов для проверки соответствия реального поведения программы ее ожидаемым функциям [5].

Особое внимание уделяется методу «белого ящика», который анализирует внутреннюю структуру программы. В фрагменте кода модульного теста (представлен листингом 7) проверяется метод добавления данных о новом пользователе, обеспечивая корректность работы приложения.

Разработка и поддержка модульных тестов играют важную роль в обеспечении качества программного продукта. Они предоставляют разработчикам инструмент для раннего выявления и устранения потенциальных проблем.

Листинг 7 – Фрагмент кода модульного теста «TestCreateUser»

|  |
| --- |
| // Arrange: Подготовка тестовых данных и контекста базы данных  using (var context = new TestDbContext())  {  // создание экземпляра для добавления нового пользователя  var userAdder = new UserAdder();  // Act:Вызов метода для добавления нового пользователя в БД  userAdder.AddNewUser  ("test\_user", "password123", 1, context);  // Assert: Проверка, что пользователь был успешно добавлен  var user = context.Users.FirstOrDefault  (u => u.Username == "test\_user");  Assert.NotNull(user);  Assert.Equal("password123", user.Password);  Assert.Equal(1, user.RoleId);  } |

Результаты модульного тестирования представлены на рисунке 4.



Рисунок 4 – Rider. Вид окна с результатами прохождения модульного тестирования

### 3.2 Функциональное тестирование

Для проверки правильности работы приложения необходимо провести тестирование методом «черного ящика» [11]. В таблице 1 представлен набор тестов для страницы «Добавление нового пользователя», который охватывает различные сценарии использования приложения.

Таблица 1 – Набор тестов страницы «Добавление нового пользователя»

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Полученный результат |
| Нажатие кнопки «Добавить» с незаполненными полями для ввода логина, пароля, имени, фамилии, отчества, почты и телефона | Вывод сообщения «Пожалуйста, заполните все обязательные поля.» | Совпал с ожидаемым |
| Ввод в поле логина пользователя: «loginTest»,  ввод в поле пароля пользователя: «passwordTest»,  ввод в поле имени пользователя: «Иван»,  ввод в поле фамилии пользователя: «Иванов»,  ввод в поле отчества пользователя: «Иванович»,  ввод в поле почты пользователя: «ivanovvv@yandex.ru»,  ввод в поле телефона пользователя: «88005553535»,  выбор в выпадающем списке роль: менеджер,  нажатие кнопки «Добавить». | Вывод сообщения «Пользователь успешно добавлен» | Совпал с ожидаемым |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Полученный результат |
| Ввод в поле логин существующего логина «manager»,  ввод в поле пароля пользователя: «managerpassword»,  ввод в поле имени пользователя: «Иван»,  ввод в поле фамилии пользователя: «Иванов»,  ввод в поле отчества пользователя: «Иванович»,  ввод в поле почты пользователя: «ivanovvv@yandex.ru»,  ввод в поле телефона пользователя: «88005553535»,  выбор в выпадающем списке роль: менеджер,  нажатие кнопки «Добавить». | Вывод сообщения «Пользователь с таким логином уже существует.» | Совпал с ожидаемым |
| Ввод в поле логин существующего логина «manager»,  ввод в поле пароля пользователя: «managerpassword»,  ввод в поле имени пользователя: «Иван»,  ввод в поле фамилии пользователя: «Иванов»,  ввод в поле отчества пользователя: «Иванович»,  ввод в поле почты пользователя: «ivanovvv@yandex.ru»,  ввод в поле телефона пользователя: «88005553535»,  выбор в выпадающем списке роль: менеджер,  нажатие кнопки «Добавить». | Вывод сообщения  «Пользователь с таким логином уже существует.» | Совпал с ожидаемым |
| Ввод в поле номер телефона пользователя: «Номер телефона»,  ввод в поле логина пользователя: «loginTest» ввод в поле пароля пользователя: «passwordTest»,  ввод в поле имени пользователя: «Иван»,  ввод в поле фамилии пользователя: «Фамилия»,  ввод в поле отчества пользователя: «Иванов»,  ввод в поле почты пользователя: «ivanovvv@yandex.ru»,  выбор в выпадающем списке роль: менеджер,  нажатие кнопки «Добавить». | Вывод сообщения «Некорректный номер телефона.» | Совпал с ожидаемым |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Полученный результат |
| Ввести корректные данные в поля ввода пользователя, который уже зарегистрирован, но выбор у него другую роль в выпадающем списке,  ввод в поле логин существующего логина «manager»,  ввод в поле пароля пользователя: «managerpassword»,  ввод в поле имени пользователя: «Иван»,  ввод в поле фамилии пользователя: «Фамилия»,  ввод в поле отчества пользователя: «Иванов»,  ввод в поле почты пользователя: «ivanovvv@yandex.ru»,  ввод в поле телефона пользователя: «88005553535»,  выбор в выпадающем списке роль: администратор,  нажатие кнопки «Добавить». | Вывод сообщения «Пользователь с таким логином уже существует.» | Совпал с ожидаемым |

После тестирования можно сделать следующие выводы о качестве разработанного приложения с учетом критериев качества ПО:

* основные функции приложения работают,
* интерфейс приложения является удобным и интуитивно–понятным для пользователей,
* приложение соответствует функциональным и эксплуатационным требованиям.

## Инструкция по эксплуатации программного обеспечения

* 1. Установка приложения

Для работы системы на стороне сервера необходимо установить следующее ПО:

* ОС Windows 10 или более новые версии, Windows Server 2016 или более новой [7],
* сервер БД: Microsoft SQL Server версии не ниже 2019 года [10],
* программное обеспечение для конфигурации, управления и администрирования Microsoft SQL Server: SQL Server Management Studio [15].

Для функционирования системы необходимо создать БД и объекты БД. Для создания БД можно использовать файл скрипта «Скрипт создания БД.sql».

Для функционирования БД необходимо заполнить ее данными. Для заполнения БД можно использовать файл скрипта «Данные.sql».

Для работы системы на стороне клиента требуется ПК с ОС Windows 10 и выше и .NET 8. Имеется возможность установки разработанного ПО или его использования как portable–приложения.

Для входа в стандартную учетную запись администратора используются следующие данные: логин «admin», пароль «admin\_pass».

Для входа в стандартную учетную запись менеджера используются следующие данные: логин «manager», пароль «manager \_pass».

**4.2 Инструкция по работе**

После запуска ПО отображается экран авторизации, где пользователь может войти в систему, используя свои учетные данные. Для обеспечения безопасности и эффективного управления информацией доступ к страницам модификации и сохранения данных в БД предоставляется только авторизованным пользователям с ролью «менеджер» или «администратор». Это обеспечивает доступ к расширенным функциональным возможностям приложения.

После авторизации пользователь с ролью «Менеджер» попадает на страницу по умолчанию, «Калькулятор», вид которой показан на рисунке 5. На этой странице пользователь имеет возможность просматривать, получать подробную информацию о выбранном материале, ввести данные для расчета и при необходимости выполнить расчет необходимого количества материалов возможностью сохранения их в виде .txt файла или сохранить результаты расчета в БД.

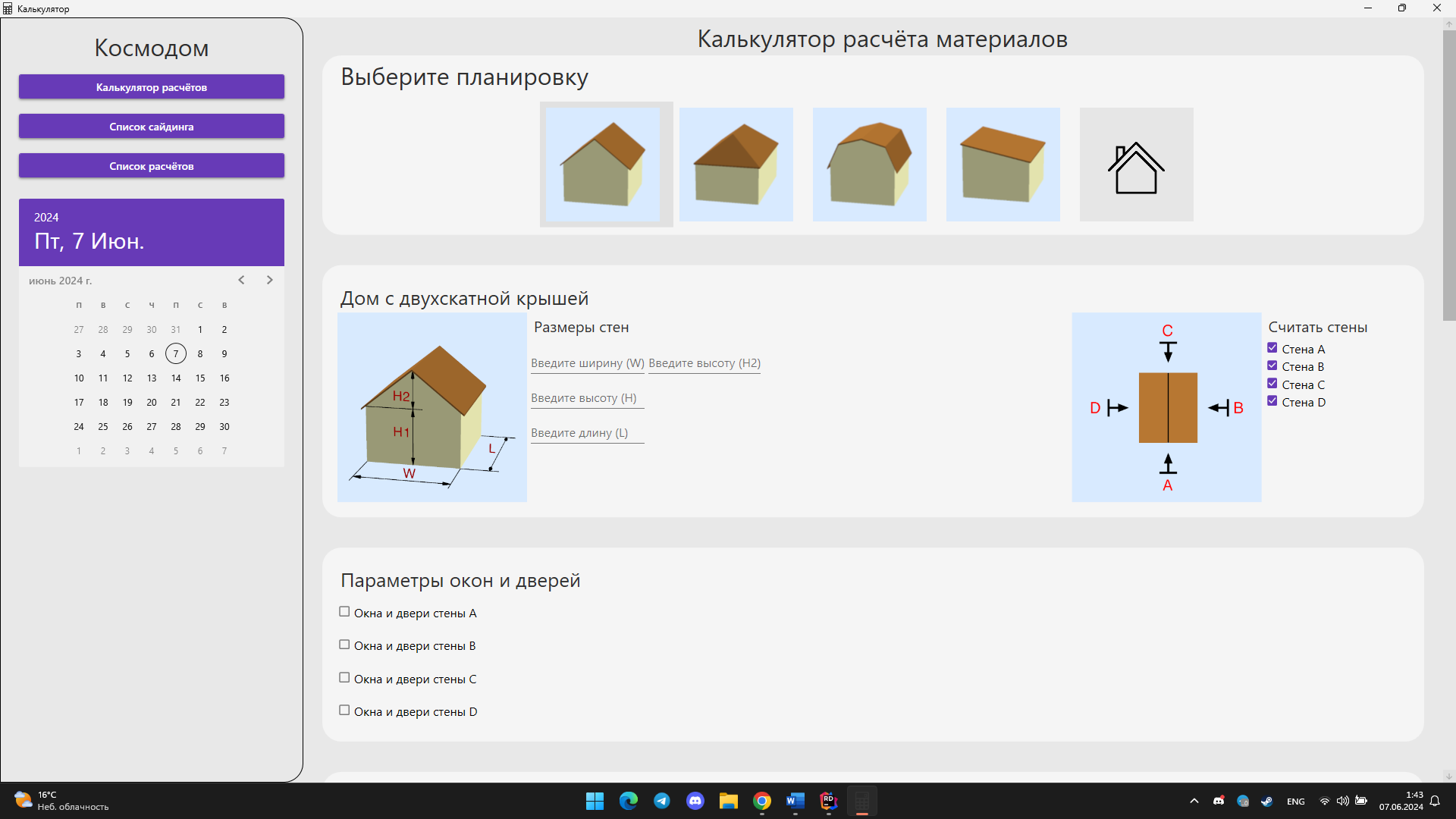


Рисунок 5 – Калькулятор. Вид окна «Создание дома»

Для удобства вычисления необходимого количества фасадных материалов сделаны четыре типовых дома и добавлена возможность самостоятельного рисования периметра дома. При нажатии на иконку с самостоятельным рисованием периметра дома открывается окно «Создание дома», представленное на рисунке 6.

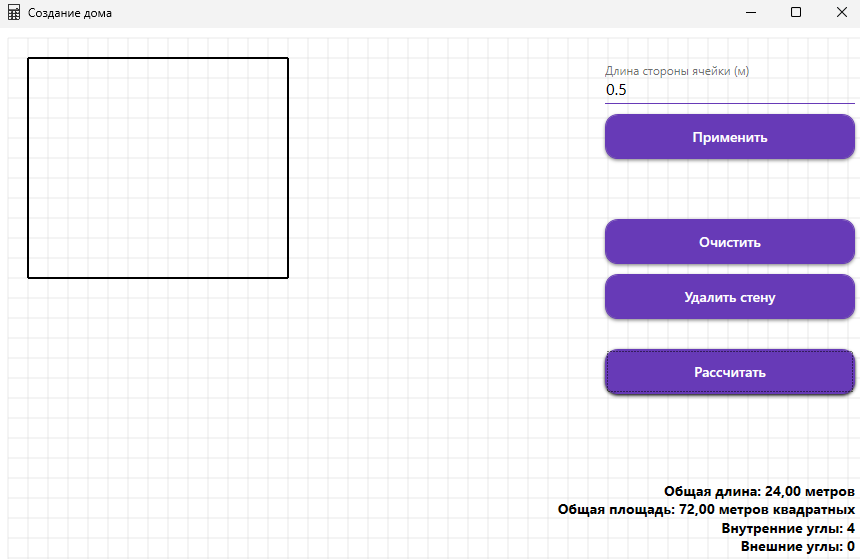


Рисунок 6 – Калькулятор. Вид страницы «Калькулятор» при загрузке

После выбора определенного материала в нижней части страницы расчетов будет выведена вся информация о нем, вид страницы с информацией показан на рисунке 7.

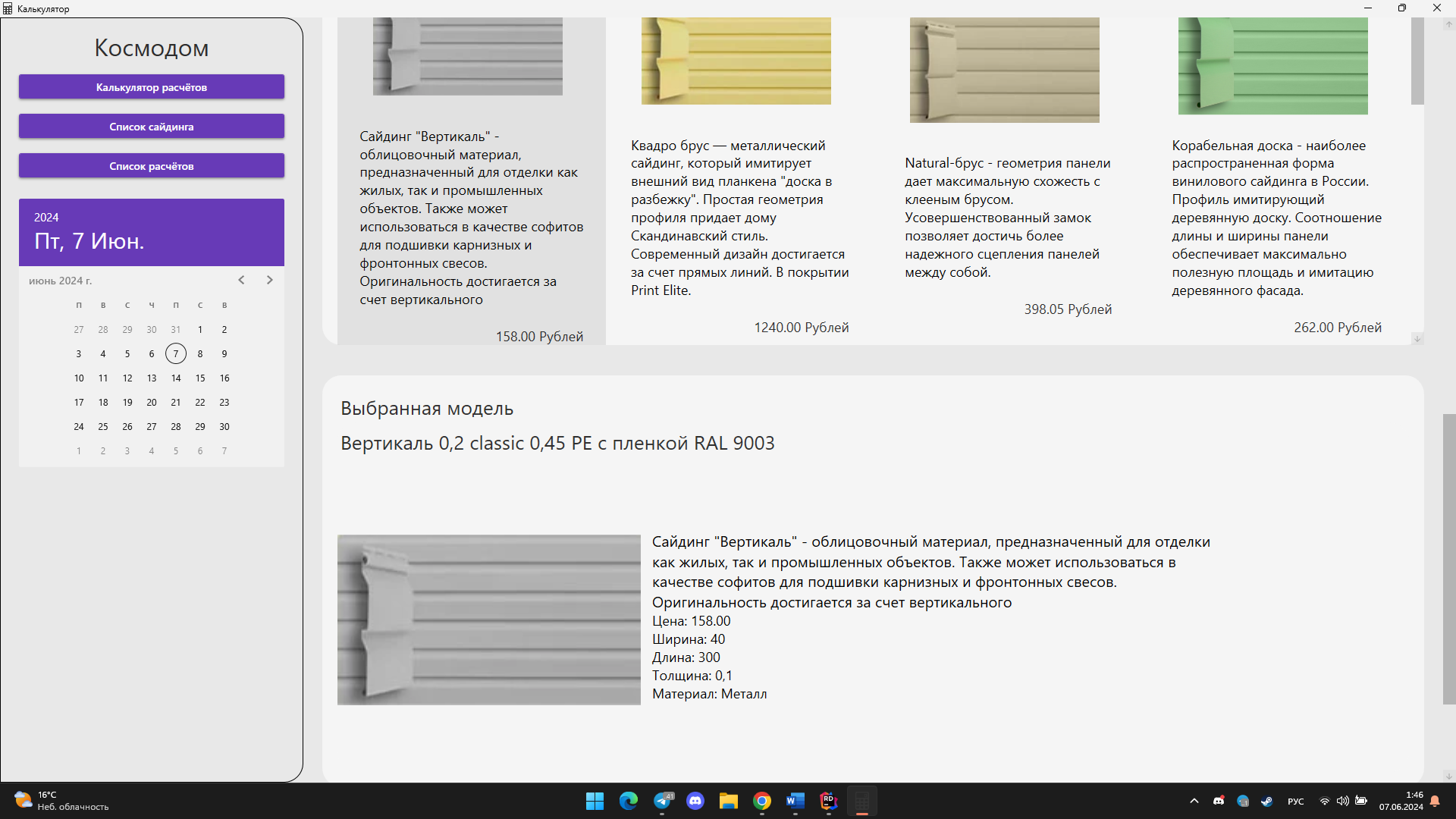


Рисунок 7 – Калькулятор. Вид страницы «Калькулятор» при выводе информации о выбранной модели

Внизу страницы располагается дополнительный пункт с параметрами, который может быть выбран пользователем. Он необходим для более точного подбора материалов, вид страницы представлен на рисунке 8. При нажатии на квадратик с выбором перед пользователем создаются текстовые поля, в которых он может ввести данные для расчета и выбрать необходимую толщину утеплителя. При нажатии на кнопку «Сделать расчет» происходит вычисление необходимого количества фасадных материалов (рисунок 9) с возможностью сохранения результатов расчета в файл .txt или в БД. Пример файла расчет представлен рисунком 10.

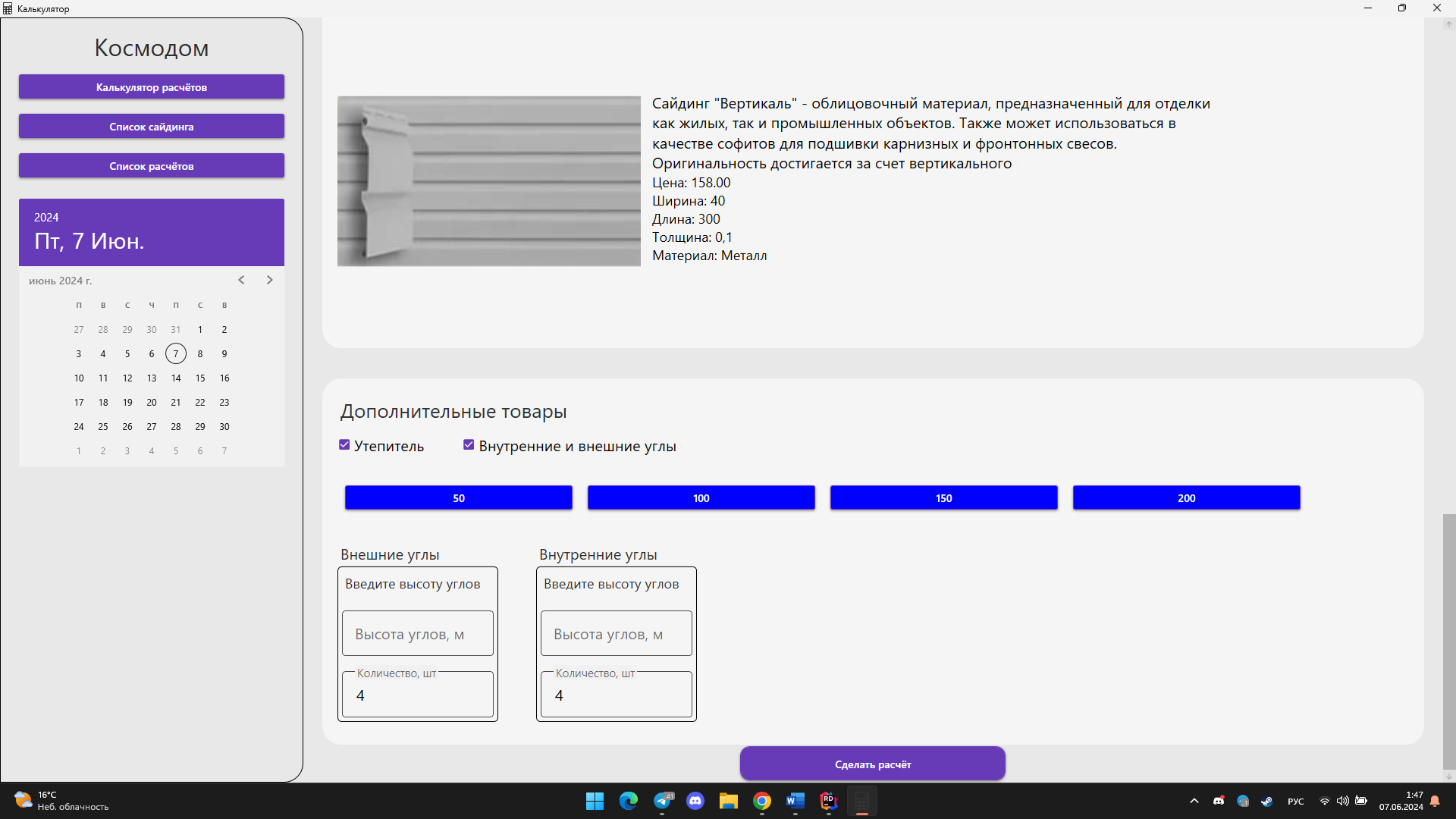


Рисунок 8 – Калькулятор. Вид страницы «Калькулятор» с дополнительными параметрами

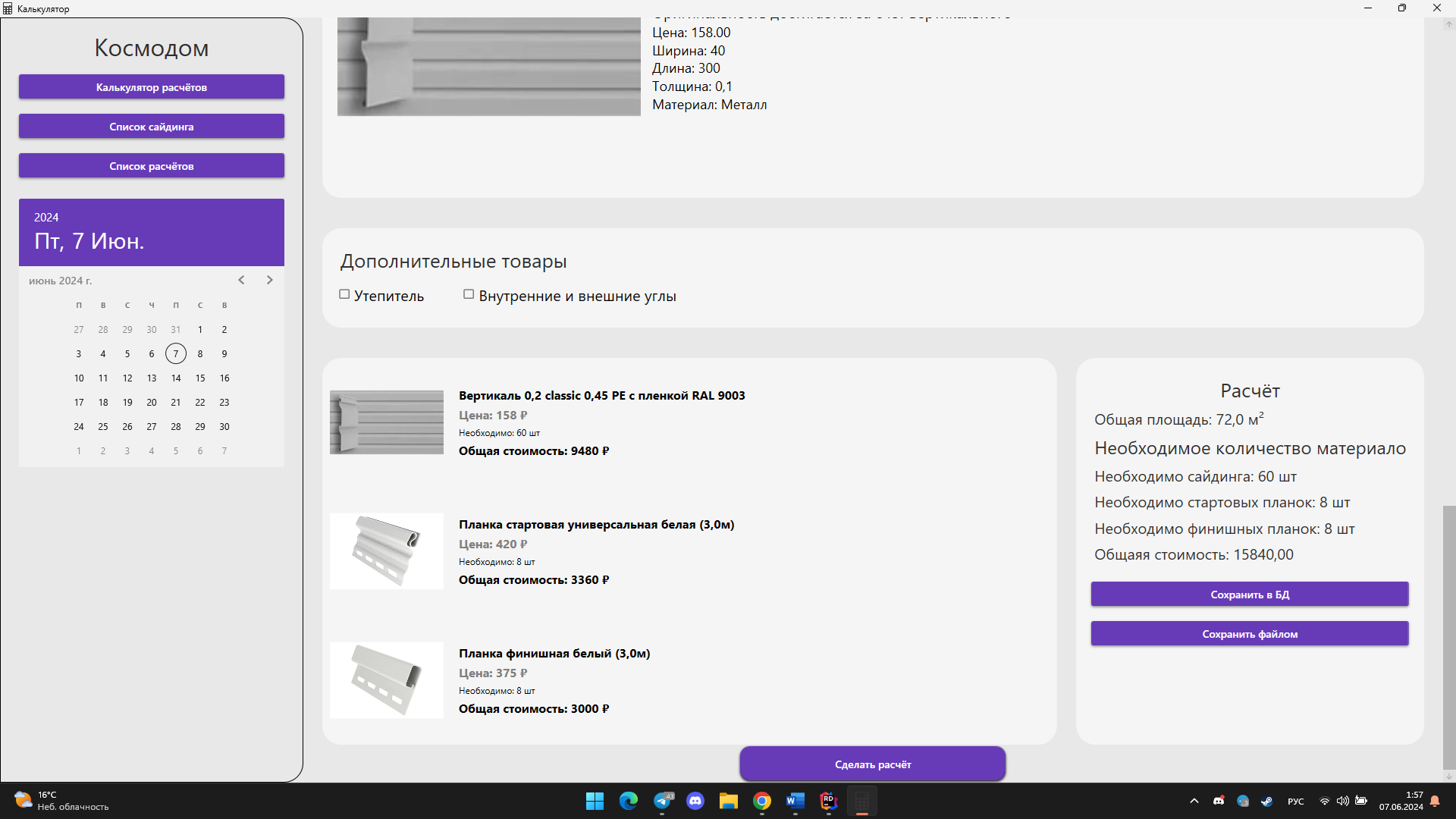


Рисунок 9 – Калькулятор. Вид страницы «Калькулятор» при выводе информации о необходимом количестве материалов

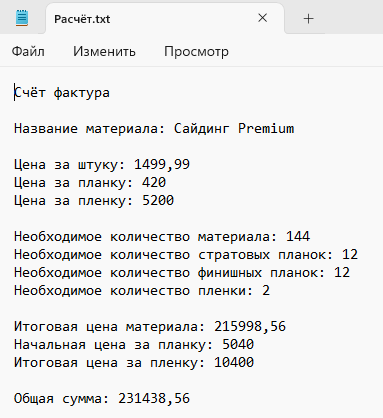


Рисунок 10 – Блокнот. Пример содержимого файла расчетов

После авторизации пользователь с ролью «Администратор» попадает на страницу «Калькулятор», представленную на рисунке 11, где пользователю доступна возможность выполнить такие же действия что и менеджеру, но есть свои дополнительные возможности.

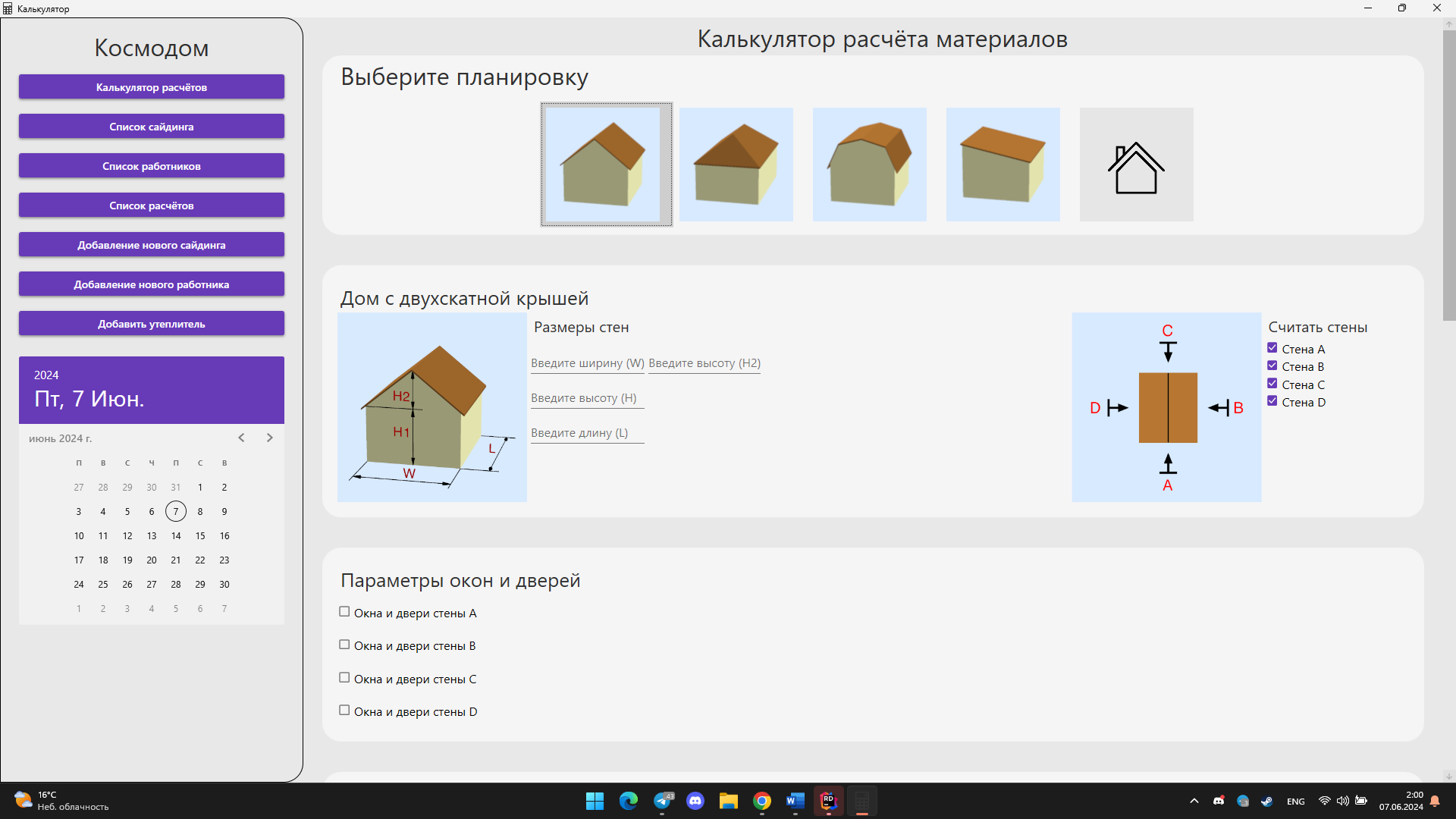


Рисунок 11 – Калькулятор. Вид страницы «Калькулятор» при загрузке

У администратора есть возможность перейти на страницу «Список работников» по нажатию соответствующей кнопки, страница представлена на рисунке 12. При нажатии кнопки «Редактировать» открывается окно, в котором администратор может обновить информацию о сотруднике (представлено на рисунке 13).

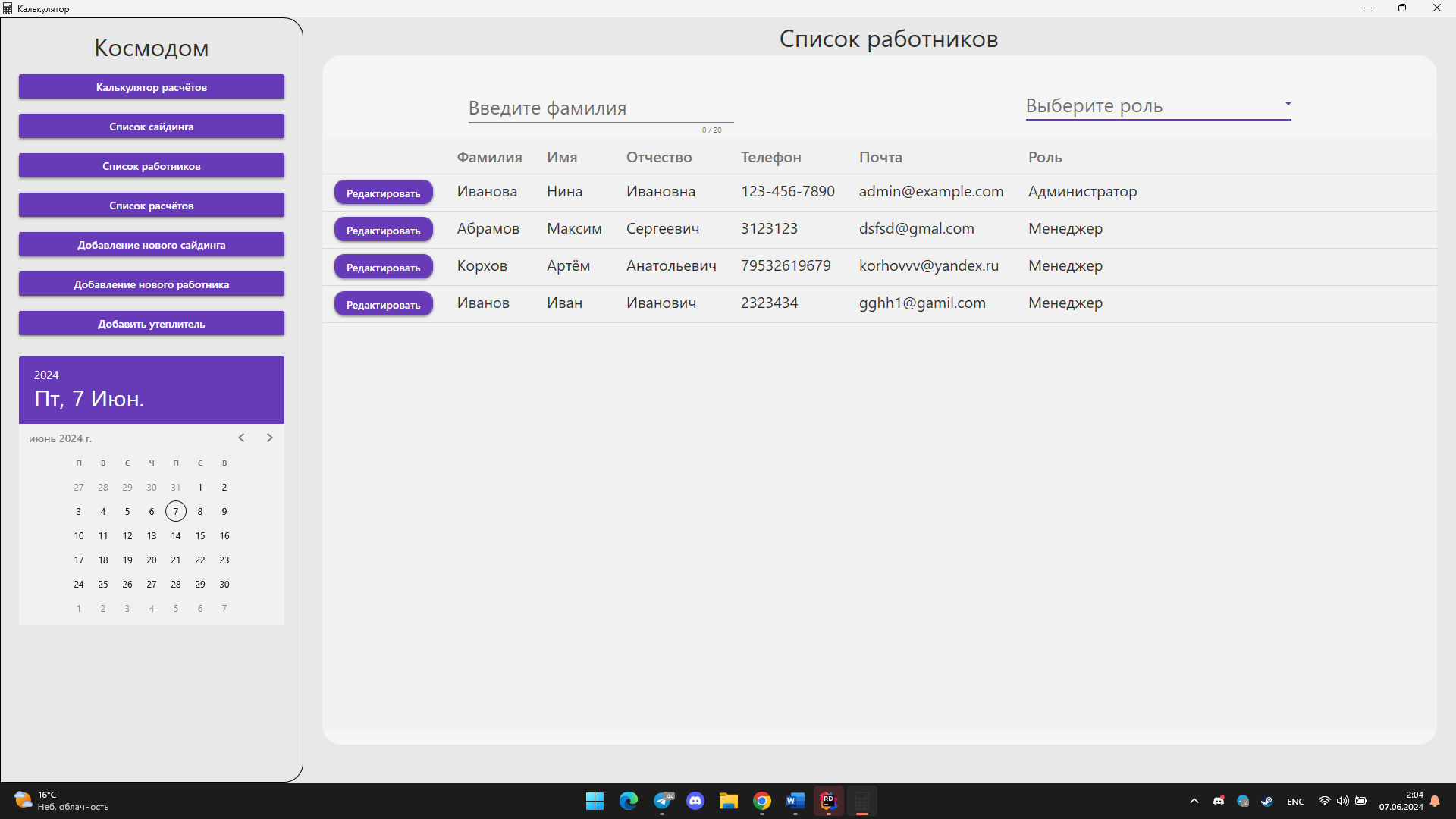


Рисунок 12 – Калькулятор. Вид страницы «Cписок работников» при загрузке

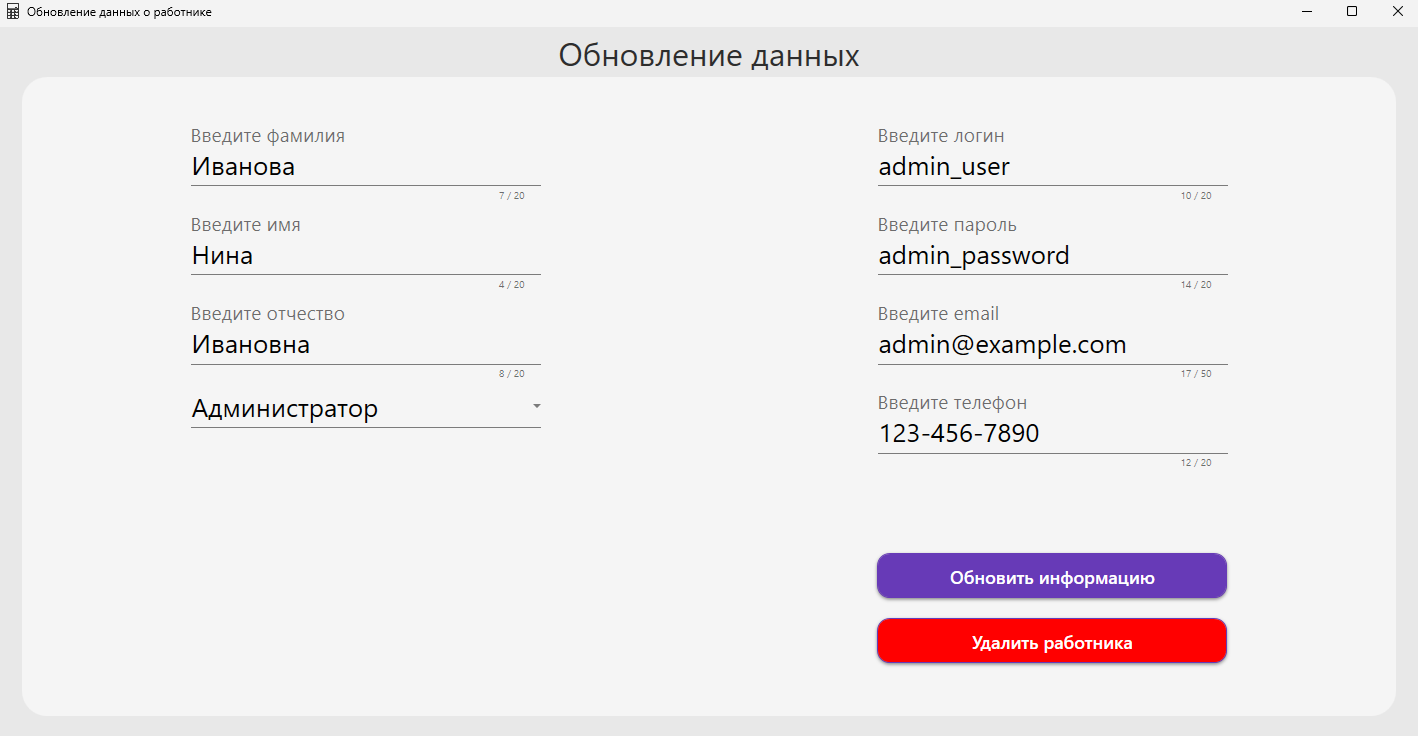


Рисунок 13 – Калькулятор. Вид окна «Обновление данных о работнике»

при загрузке

## 5 Определение затрат на разработку

Определение затрат на разработку является одним из ключевых элементов дипломного проектирования. Оно позволяет провести комплексную оценку целесообразности и эффективности разработки ПО.

Процесс разработки ИС необходимо планировать, контролировать и завершить в срок. Труд разработчика программного продукта нужно оплачивать по результатам его труда (его качеству, количеству и интенсивности).

Дипломное проектирование направлено на разработку ИС. Затраты на разработку ИС складываются из средств, затраченных на создание программного продукта, затрат на эксплуатацию ЭВМ и общих затрат, необходимых для написания программы:

Затраты на создание программного продукта Зспп, руб., определяются по формуле

, (1)

где Змв – затраты на оплату машинного времени, руб.;

Зобщ – общие затраты.

Трудоемкость разработки ИС, то есть время на него потраченное, рассчитывается по формуле

, (2)

где t𝑜 – затраты труда на подготовку описания задачи, чел.ч;

t𝑢 – затраты труда на исследование алгоритма решения, чел.ч;

t𝑎 – затраты труда на разработку блок–схемы алгоритма, чел.ч;

tп – затраты труда на программирование, чел.ч;

tотл – затраты труда на отладку программы, чел.ч;

tд – затраты труда на подготовку документации, чел.ч.

Все составляющие трудозатраты определяются через условное число операторов. Условное число операторов (Q) в программе может быть определено по формуле

, (3)

где q – число исходных команд;

с – коэффициент, учитывающий новизну и сложность программы;

р – коэффициент коррекции программы в ходе ее разработки;

зависит от точности и корректности поставленной задачи.

Коэффициент, учитывающий новизну и сложность программы (с) определяется исходя из таблицы 2 на пересечении групп сложности и степени новизны.

Программный продукт по степени новизны может быть отнесен к одной из четырех групп:

* группа А – разработка принципиально новых задач,
* группа Б – разработка оригинальных программ,
* группа В – разработка программ с использованием типовых,
* группа Г – разовая типовая задача.

По степени сложности программные продукты могут быть отнесены к

одной из трех групп:

* 1 – алгоритмы оптимизации и моделирования систем,
* 2 – задачи учета, отчетности и статистики,
* 3 – стандартные алгоритмы.

Коэффициент, учитывающий новизну и сложность программы, определяется исходя из таблицы 2. Созданный программный продукт по степени новизны относится к разработка оригинальных программ, а по степени сложности алгоритма к алгоритмам оптимизации и моделирования систем.

Таблица 2 – Значение коэффициента с и В

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Язык  программирования | Группа сложности | Степень новизны с | | | | Коэффициент В |
| А | Б | В | Г |
| Высокого уровня | 1 | 1,38 | 1,26 | 1,15 | 1,20 | 1,20 |
| 2 | 1,30 | 1,19 | 1,08 | 0,65 | 1,35 |
| 3 | 1,20 | 1,10 | 1,00 | 0,60 | 1,50 |
| Низкого уровня | 1 | 1,58 | 1,45 | 1,32 | 0,79 | 1,20 |
| 2 | 1,49 | 1,37 | 1,24 | 0,74 | 1,35 |
| 3 | 1,38 | 1,26 | 1,15 | 0,69 | 1,50 |

Тогда, по таблице 2 коэффициент c = 1,19 и коэффициент B = 1,35.

Зная исходное число команд (q), для определения условного числа операторов Q нужно воспользоваться формулой (3). Число p можно принять равным 0,04

.

Затраты труда на изучение описания задачи с учетом уточнения описания и квалификации программиста определяются как

, (4)

где В – коэффициент увеличения затрат труда вследствие недостаточного описания задачи, уточнений и некоторой недоработки;

К – коэффициент квалификации разработчика;

75…85 – здесь и далее – диапазон значения.

Коэффициент увеличения затрат труда вследствие недостаточного описания задачи (В) определяется по таблице 2 в зависимости от найденного значения с. Коэффициент квалификации разработчика (К) можно установить по данным таблицы 3 в зависимости от его стажа работы.

Таблица 3 – Коэффициент квалификации разработчика

|  |  |
| --- | --- |
| Опыт работы | Коэффициент квалификации |
| До двух лет | 0,80 |
| 2–3 года | 1,00 |
| 3–5 лет | 1,10 – 1,20 |
| 5–7 лет | 1,30 – 1,40 |
| Более 7 лет | 1,50 – 1,60 |

По таблице 3 для работающих до двух лет К = 0,8

Затраты труда на разработку алгоритма решения задачи ta, чел.ч, определяются по формуле

, (5)

Затраты труда на разработку схемы алгоритма решения задачи определяются следующим образом

, (6)

Затраты труда на составление программы по готовой схеме алгоритма tп, чел.ч, определяются по формуле

, (7)

Затраты труда на отладку программы на ЭВМ при комплексной отладке tот, чел.ч, определяются по формуле

, (8)

где tА – затраты труда на отладку программы на ЭВМ при автономной отладке одной задачи.

Затраты труда на отладку программы на ЭВМ при автономной отладке одной задачи tА, чел.ч, определяются по формуле

, (9)

Расчет затраты труда на отладку программы на ЭВМ при комплексной

отладке выполняется по формуле (8)

Затраты труда на подготовку документации по задаче tд, чел.ч, определяются по формуле

, (10)

где tдр – затраты труда на подготовку материалов рукописи, чел.ч;

tдо – затраты на редактирование, печать и оформление документации, чел.ч.

Затраты труда на подготовку материалов рукописи находятся по формуле

, (11)

Затраты на редактирование, печать и оформление документации tдо, чел.ч, определяются по формуле

, (12)

Расчет затрат труда на подготовку документации по задаче выполняется по формуле (10)

Трудоемкость разработки программного продукта, согласно формуле (2), составляет

t = 50 + 13,9 + 14,4 + 10,56 + + = 118,61 чел.ч

Трудоемкость разработки составила 118,61 чел.ч, что составляет 15 рабочих дней при восьмичасовом рабочем дне. Далее необходимо вычислить все финансовые затраты на разработку проекта. На рисунке 16 показана структура трудозатрат проекта.

Рисунке 16 – Структура трудозатрат проекта. Диаграмма круговая

Затраты на оплату машинного времени при отладке программы Змв , руб., определяются по формуле

, (13)

где Счас – цена машино–часа арендного времени, руб/ч;

tэвм – фактическое время отладки программы на ЭВМ, чел.ч.

Фактическое время отладки 𝑡эвм, чел.ч, определяется по формуле

, (14)

Цена машино–часа Счас, руб/ч, определяется по формуле

, (15)

где Зэвм – затраты на эксплуатацию ЭВМ, руб.;

Tэвм – действительный месячный фонд времени ЭВМ, ч.

Действительный месячный фонд времени ЭВМ Tэвм ч, определяется по формуле

, (16)

где 8 – восьмичасовая рабочая норма времени при 40–часовой рабочей

неделе;

Кд – количество календарных дней в месяце;

Кпв – число праздничных и выходных дней в месяце;

tпр – время простоя в профилактических работах – это необходимая еженедельная профилактика и цифровая гигиена по 4 часа на протяжении месяца, то есть 4 недели.

Общее количество календарных дней в месяце , число праздничных и выходных дней в месяце в соответствии со статьей 112 трудового кодекса РФ [4].

Выполняется расчет действительного месячного фонда времени ЭВМ по формуле (16)

Затраты на эксплуатацию ЭВМ Зэвм, руб., определяются по формуле

, (17)

где Зам – издержки на амортизацию, руб.;

Зэл – издержки на электроэнергию, потребляемую ЭВМ, руб.

Согласно ч. 2 ст.256 и ст. 257 п. 1 налогового кодекса (НК) РФ, амортизируемым имуществом признается имущество со сроком полезного использования более 12 месяцев и первоначальной стоимостью более   
100,0 тыс. руб. [1].

Так как стоимость ноутбука менее 100,0 тыс. руб., ежемесячные амортизационные отчисления не выполняются, а стоимость оборудования сразу и полностью списывается на расходы предприятия. Значит сумма годовых амортизационных отчислений принимается за 0.

Стоимость электроэнергии, потребляемой за месяц, определяется по формуле

(18)

где РЭВМ – суммарная мощность ЭВМ;

ТЭВМ – время работы компьютера, час.;

Сэл – стоимость 1кВт × ч электроэнергии.

Согласно техническому паспорту ЭВМ, потребление электроэнергии составляет 0,135 кВт. Исходя из тарифа 4,30 который используется [ссылку на тариф по электричеству], рассчитаем стоимость электроэнергии, потребляемой за месяц по формуле (18)

Затраты на эксплуатацию ЭВМ, согласно формуле (17), составляют

Цена машино–часа, согласно формуле (15), составляет

Затраты на оплату машинного времени при отладке программы, согласно формуле (13), составляют

Общие затраты Зобщ, руб., определяются по формуле

, (19)

где Ззп – издержки на заработную плату, руб.;

Зсв – издержки на страховые взносы с оплаты труда, руб.;

Зам – издержки на амортизацию, руб.;

Зэл – издержки на электроэнергию, потребляемую ЭВМ, руб.;

Зпр – издержки на прочие и накладные расходы, руб.

Для расчета заработной платы оклад программиста принимается равным величине МРОТ, который согласно ст.1 Федерального закона от 19.06.2000 №82–ФЗ «О минимальном размере оплаты труда» в редакции Федерального закона от 27.11.2023 № 548–ФЗ [3], составляет 19242,00 руб. в месяц без учета районных коэффициентов и надбавок.

Так как разработка проекта ведется на территории с особыми климатическими условиями – г. Архангельск, то на основании статьи 317 ТК РФ [2] и ст. 11 закона РФ от 19.02.1993 г. № 4520–1 [2] к окладу применяются районные коэффициенты и надбавки

, (20)

где РК – районный коэффициент для работающих в местах с определенными климатическими условиями, в рассматриваемом случае – районы, приравненные к Крайнему Северу;

СН – коэффициент, учитывающий северную надбавку за стаж работы на территориях, приравненных к Крайнему Северу.

Районный коэффициент в г. Архангельске составляет 20% от основной заработной платы, а выплаты за выслугу лет, проработанных на территории, приравненной к территории Крайнего Севера – 50% от основной заработной платы. За отработанный месяц заработная плата должна быть не меньше МРОТ, который учитывая районные и северные коэффициенты в г. Архангельске составляет 32711,40 руб. Заработная плата рассчитана по формуле (20) за полный отработанный месяц

Издержки на страховые взносы с оплаты труда определяются гл. 34 ст. 425 НК РФ как сумма взносов в ПФР – 22%, ФОМС – 5.1% и ФСС – 2,9%.C 01.01.2023 года страховые взносы перечисляются в налоговую единым платежным поручением общей суммой, поэтому размер страховых взносов с заработной платы можно принять равным 30% без учета взносов на травматизм, так как эта сумма зависит от опасности производства

, (21)

где Фот – фонд оплаты труда, руб;

Нвнеб – страховые взносы с оплаты труда, %.

Прочие затраты Зпр, руб., принимаются в размере 10% в общей сумме затрат и определяются по формуле

, (22)

Рассчитываем общие затраты, руб., по формуле (19)

Затраты на создание программного продукта, согласно формуле (1), составляют

В результате выполненных расчетов затраты на создание ИС составляют ,59 руб. На рисунке 14 предоставлена диаграмма затрат.

Рисунок 14 – Диаграмма затрат

Создание ИС для ООО «Космодом» потребовало 118,61 человеко–часов, что привело к затратам в размере ,59 рублей. В Архангельске средняя стоимость разработки ИС колеблется от 50000 до 160000 рублей в зависимости от сложности системы, объема работ и других факторов, влияющих на разработку. Для ООО «Космодом» такие затраты являются приемлемыми. Учитывая рыночные цены и допустимые уровни общих затрат, создание ИС представляется целесообразным. Тем не менее необходимо тщательно оценить риски и затраты, а также учесть конкуренцию и потенциальную прибыль.

С учетом выявленных рисков и при соблюдении всех параметров планирования в допустимых пределах можно заключить, что проект имеет высокую вероятность успешного завершения, достижения поставленных целей и обеспечения экономической эффективности.

При расчете экономической части учтены затраты на разработку и внедрение системы, а также оценены экономические эффекты от ее использования. В результате анализа установлено, что внедрение ИС позволит значительно сократить время на ручное высчитывание и подготовку документов для заказчика, улучшить качество расчетов и повысить эффективность работы персонала. Кроме того, использование разработанной системы позволит снизить затраты на ручной счет, что приведет к экономической выгоде для компании. В целом, определение затрат на разработку проекта подтвердило его целесообразность и эффективность внедрения. Проект является коммерческим, предназначен для упрощения работы с заказами и удовлетворяет интересам заказчика.

## 6 Охрана труда и техника безопасности при работе на ПК

### 6.1 Общие требования охраны труда

К самостоятельной работе на ПК допускаются лица, прошедшие предварительные и периодические медицинские осмотры, вводный инструктаж по охране труда, обучение безопасным приемам работы на ПК и первичный инструктаж на конкретном рабочем месте.

При работе на ПК работнику необходимо учитывать возможное воздействие следующих вредных и опасных производственных факторов: повышенная температура поверхностей ПК, повышенное напряжение в электрической цепи, уровень статического электричества, уровень электромагнитных излучений, напряженность электрического поля, недостаточная освещенность рабочей зоны, повышенная яркость света, повышенная контрастность, зрительное напряжение, длительные статические нагрузки и монотонность трудового процесса.

При работе на ПК работник обязан соблюдать производственную и технологическую дисциплину, регулировать режим труда и отдыха, выполнять задачи по должностной инструкции, поддерживать порядок на рабочем месте, сообщать о неисправностях ПК и электропитания руководителю и соблюдать требования пожарной и электробезопасности.

За нарушение требований охраны труда работник привлекается к ответственности в порядке, установленном законодательством РФ.

### 6.2 Требования охраны труда перед началом работы

Перед началом работы на ПК пользователь должен проветрить помещение, отрегулировать освещенность на рабочем месте, проверить правильность расположения компьютера для исключения неудобных поз и напряжений тела, убрать посторонние предметы с рабочего места и освободить подходы к нему.

Необходимо проверить исправность кабельных соединений, проверить отсутствие пыли на экране монитора и клавиатуре, протереть их при необходимости специальной салфеткой.

### 6.3 Требования охраны труда во время работы

Подключение ПК к сети электропитания выполнять только имеющимися штатными сетевыми кабелями при закрытых кожухах и наличии заземления.

При включении ПК необходимо соблюдать следующую последовательность действий: сначала включить сетевой фильтр, затем источник бесперебойного питания, системный блок и монитор. После включения ПК и запуска программы, убедиться в отсутствии дрожания и мерцания изображения на экране монитора, настроить яркость, контрастность, цвет и размер символов, и фон экрана для наиболее комфортного и четкого восприятия изображения.

Во время работы на ПК работник должен избегать натягивания, скручивания, перегиба и пережима шнуров электропитания, предотвращать попадание предметов на шнуры, избегать влаги на поверхности ПК, не прикасаться к задней панели системного блока при включенном питании, не оставлять включенный ПК без наблюдения, не загромождать рабочее место, не проводить самостоятельный ремонт ПК, не закрывать вентиляционные отверстия, не часто отключать и включать ПК, соблюдать расстояние от глаз до экрана (50–70 см), не превышать двух часов непрерывной работы без перерыва.

### 6.4 Требования охраны труда по окончании работы

Отключить ПК от электросети сухими руками, держась за вилку штепсельного соединителя, после чего, привести рабочее место в порядок.

6.5 Требования охраны труда в аварийных ситуациях

При возникновении аварийной ситуации на рабочем месте необходимо немедленно прекратить работу и сообщить о ней непосредственному руководителю или его замещающему. Под руководством непосредственного руководителя работник должен принять участие в ликвидации аварийной ситуации, если это не представляет угрозы для здоровья и жизни. При необходимости работник должен покинуть опасную зону.

При возникновении нарушений в работе ПК или электросети необходимо немедленно прекратить работу, отключить ПК от электросети и сообщить о случившемся непосредственному руководителю или его замещающему. Недопустимо приступать к работе до полного устранения повреждений или неисправностей.

При возникновении возгорания необходимо немедленно прекратить работу, отключить ПК и сообщить о случившемся руководителю. Затем требуется провести попытку тушения с использованием углекислотного или порошкового огнетушителя. Если тушение невозможно, следует покинуть опасную зону через эвакуационные выходы в соответствии со схемой эвакуации.

При резком ухудшении состояния здоровья или травмировании необходимо прекратить работу, сообщить о случившемся непосредственному руководителю или лицу, его замещающему. При необходимости следует воспользоваться аптечкой.

## ЗАКЛЮЧЕНИЕ

В результате дипломного проектирования разработана ИС для ООО «Космодом». Разработанная ИС представляет собой значимый шаг в направлении автоматизации расчетов, обеспечивая возможность оперативного внесения изменений в расчет и его обновления для всех пользователей калькулятора.

Анализ требований осуществлен с учетом разнообразия фасадных материалов и предполагаемых запросов пользователей. В результате достигнута основная цель дипломного проектирования, т.е. разработана ИС для ООО «Космодом».

В ходе разработки ИС решены следующие задачи:

* проанализированы функциональные и эксплуатационные требования к ПО,
* спроектирована архитектура ПО,
* разработана БД,
* созданы и заполнены таблицы БД,
* реализовано добавление и обновление данных о фасадных материалах,
* реализован модуль для расчета количества требуемых фасадных материалов и необходимых комплектующих,
* реализован модуль для графического построения контура дома с последующим вычислением его периметра и площади требуемых фасадных материалов,
* реализован экспорт и импорт данных ПО,
* реализовано разграничение прав доступа пользователей,
* проведена интеграция модулей ПО,
* выполнены тестирование и отладка ПО,
* разработана программная и эксплуатационная документация.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Российская Федерация. Закон. Налоговый Кодекс Российской Федерации : Федеральный закон № 117 – ФЗ : в редакции с 29.05.2024 с изменениями и дополнениями вступившими в силу с 01.06.2024 г. : [принят Государственной Думой 21 мая 2024 года : одобрен Советом Федерации 22 мая 2024 года]. – Москва : Эксмо, 2024. – 1682 с.
2. Российская Федерация. Закон. О государственных гарантиях и компенсациях для лиц, работающих и проживающих в районах Крайнего Севера и приравненных к ним местностях : Федеральный закон № 4520–1 – ФЗ : в редакции от 25.12.2023 г. : [принят и одобрен Президентом Российской Федерации Б. Ельцином от 19 февраля 1993 года]. – Москва : Проспект, 2023. – 36 с.
3. Российская Федерация. Закон. О минимальном размере оплаты труда : Федеральный закон № 82 – ФЗ : в редакции Федеральным законом № 548 – ФЗ : [принят Государственной Думой 16 ноября 2023 года : одобрен Советом Федерации 22 ноября 2023 года]. – Москва :Проспект, 2023. – 2 с.
4. Российская Федерация. Закон. Трудовой кодекс Российской Федерации : Федеральный закон № 197 – ФЗ : в редакции от 06.04.2024 г. : [принят Государственной Думой 21 декабря 2001 года : одобрен Советом Федерации 26 декабря 2001 года]. – Москва : Проспект, 2024. – 323 с.
5. Блэнд, Д. Тестирование бизнес–идей / Дэвид Блэнд, Алекс Остервальдер ; пер. с англ. – Москва : Альпина Паблишер, 2020. – 354 с. – ISBN 978–5–9614–3658–7. – URL: https://znanium.com/catalog/product/1222476 (дата обращения: 07.05.2024). – Режим доступа: по подписке. – Текст : электронный.
6. Бурков, А. В. Проектирование информационных систем в Microsoft SQL Server 2008 и Visual Studio 2008 : краткий курс / А. В. Бурков. – Москва : ИНТУИТ, 2016. – 226 с – URL: https://znanium.ru/catalog/product/2156693 (дата обращения: 07.06.2024). – Режим доступа: по подписке. – Текст : электронный.
7. Власов, Ю. В. Администрирование сетей на платформе MS Windows Server : краткий учебный курс / Ю. В. Власов, Т. И. Рицкова. – Москва : ИНТУИТ, 2016. – 431 с. – ISBN 978–5–94774–858–1. – URL: https://znanium.ru/catalog/product/2137408 (дата обращения: 07.05.2024). – Режим доступа: по подписке. – Текст : электронный.
8. Грекул, В. И. Проектирование информационных систем : краткий курс / В. И. Грекул. – Москва : ИНТУИТ, 2016. – 400 с. – URL: https://znanium.ru/catalog/product/2156692 (дата обращения: 07.05.2024). – Режим доступа: по подписке. – Текст : электронный.
9. Долженко, А. И. Разработка приложений на базе WPF и Silverlight : курс лекций / А. И. Долженко. – Москва : ИНТУИТ, 2016. – 352 с. – URL: https://znanium.ru/catalog/product/2157898 (дата обращения: 07.06.2024). – Режим доступа: по подписке. – Текст : электронный.
10. Митин, А. И. Работа с базами данных Microsoft SQL Server: сценарии практических занятий / А. И. Митин. – Москва : Директ–Медиа, 2020. – 142 с. – ISBN 978–5–4499–0420–1. – URL: https://znanium.com/catalog/product/1985743 (дата обращения: 07.04.2024). – Режим доступа: по подписке. – Текст : электронный.
11. Морозова, Ю. В. Тестирование программного обеспечения : учебное пособие / Ю. В. Морозова. – Томск : Эль–Контент, 2019. – 120 с. – ISBN 978–5–4332–0279–5. – – URL: https://znanium.com/catalog/product/1845910 (дата обращения: 07.05.2024). – Режим доступа: по подписке. – Текст : электронный.
12. Официальная документация .NET: электронное учебное пособие // Microsoft – 2019 – URL: <https://dotnet.microsoft.com/> (дата обращения: 05.03.2024) – Режим доступа: для всех пользователей. – Текст : электронный.
13. Официальная документация draw.io: электронное учебное пособие / JGraph Ltd – URL: <https://www.drawio.com/doc/> (дата обращения: 28.03.2024) – Режим доступа: для всех пользователей. – Текст : электронный.
14. Официальная документация Microsoft.EntityFrameworkCore: электронное учебное пособие // Microsoft – 2021 – URL: [https://docs.microsoft.com/en–us/ef/core/](https://docs.microsoft.com/en-us/ef/core/) (дата обращения: 21.03.2024) – Режим доступа: для всех пользователей. – Текст : электронный.
15. Разработка и защита баз данных в Microsoft SQL Server 2005 : краткий курс / . – Москва : ИНТУИТ, 2016. – 107 с. – Текст : электронный. – URL: https://znanium.ru/catalog/product/2157480 (дата обращения: 07.06.2024). – Режим доступа: по подписке.
16. Смит, Д. Entity Framework Core в действии : практическое руководство / Д. Смит ; пер. с англ. Д. А. Беликова. – Москва : ДМК Пресс, 2023. – 690 с. – ISBN 978–5–93700–114–6. – URL: https://znanium.com/catalog/product/2109485 (дата обращения: 07.06.2024). – Режим доступа: по подписке. – Текст : электронный.
17. Торстейнсон, П. Криптография и безопасность в технологии .NET / П. Торстейнсон, Г. А. Ганеш. – 4–е изд. – Москва : Лаборатория знаний, 2020. – 482 с. – (Программисту). – ISBN 978–5–00101–700–4. – URL: https://znanium.com/catalog/product/1201346 (дата обращения: 07.05.2024). – Режим доступа: по подписке. – Текст : электронный.
18. Фиайли, К. SQL. Руководство для использования с любыми SQL СУБД : учебное пособие / К. Фиайли ; пер. с англ. А. В. Хаванова. – 2–е изд. – Москва : ДМК Пресс, 2023. – 454 с. – ISBN 978–5–89818–323–3. – URL: https://znanium.com/catalog/product/2102610 (дата обращения: 07.06.2024). – Режим доступа: по подписке. – Текст : электронный.
19. Хорев, П. Б. Объектно–ориентированное программирование с примерами на С# : учебное пособие / П.Б. Хорев. – Москва : ФОРУМ : ИНФРА–М, 2023. — 200 с. — (Высшее образование: Бакалавриат). – ISBN 978–5–00091–680–3. – URL: https://znanium.ru/catalog/product/1926392 (дата обращения: 24.03.2024). – Режим доступа: по подписке. – Текст : электронный.
20. Шатрова, А. П. Практикум по курсу «Базы данных». Введение в MS SQL Server и T–SQL : практикум / А. П. Шатрова, А. В. Заворотный, Е. А. Крюкова. – Москва : Издательский Дом НИТУ «МИСиС», 2023. – 88 с. – URL: https://znanium.ru/catalog/product/2148222 (дата обращения: 10.03.2024). – Режим доступа: по подписке. – Текст : электронный.