1. **Назначение и условия применения программы**

Программа предназначена для упрощения и автоматизации работы со сбытом, распределением и просмотром хранящихся материалов на складах, а также для просмотра состава материалов.

Для занесения данных о проекте работают сотрудники различных должностей. При поступлении сотрудника на работу все данные регистрируются.

|  |  |
| --- | --- |
| Процессор | Intel® Core™ i5-6400 CPU @ 2.70GHz |
| Оперативная Память | 2 x 4 ГБ, DDR4, DIMM, 2133 МГц |
| Разрешение экрана | 1920x1080 |
| Размер экрана | 20 дюйма |
| Устройства ввода | Мышь, клавиатура |
| Дисковое пространство | 1Гб |
| Подключаемые устройства | Принтер |
| Операционная система | Windows 8.1 10 11 |

Требования к программным средствам:

Для корректной работы программы необходимо:

Для корректной работы программы необходимо:

ОС Windows 8.1/10/11 – операционная система, сделанная корпорацией Microsoft;

Visual Studio 2022 – лучшая интегрированная среда разработки для создания многофункциональных, привлекательных кроссплатформенных приложений для Windows;

Microsoft SQL Server Management Studio 18 – интегрированная среда для управления любой инфраструктурой SQL, от SQL Server до баз данных;

1. **Характеристика программы**

Программа должна обеспечивать следующие функции:

Добавление, удаление и редактирование материалов, заказов, складов, движение, хранения, состав материала, сбыт.

Поиск материалов, склада, сбыта, хранящихся материалов, пользователей по всем атрибутом

Фильтрация материалов, склада, сбыта, хранящихся материалов, по всем атрибутом

1. **Обращение к программе**

С# — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270

Код страницы «Заказ»:

using SalesOfMaterials.Classes;

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Animation;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace SalesOfMaterials.Pages

{

/// <summary>

/// Логика взаимодействия для PageOrder.xaml

/// </summary>

public partial class PageOrder : Page

{

public PageOrder()

{

InitializeComponent();

dgExpenseIvoices.ItemsSource = ClassFrame.db.ExpenseIvoices.ToList();

switch (Users.GetUsers.IdTypesEmployee)

{

case 1:

Button btn = new Button() { Name = "btnAddOrder", Width = 130, Height = 30, Content = "Добавить новый заказ", Margin = new Thickness(10) };

btn.Click += btnAddOrder\_Click;

stplButtons.Children.Add(btn);

Button btn2 = new Button() { Name = "btnDistribution", Width = 130, Height = 30, Content = "Распределить", Margin = new Thickness(10) };

btn2.Click += btnDistribution\_Click;

stplButtons.Children.Add(btn2);

break;

case 2:

Button btn3 = new Button() { Name = "btnAddOrder", Width = 130, Height = 30, Content = "Добавить новый заказ", Margin = new Thickness(10) };

btn3.Click += btnAddOrder\_Click;

stplButtons.Children.Add(btn3);

break;

case 3:

Button btn4 = new Button() { Name = "btnDistribution", Width = 130, Height = 30, Content = "Распределить", Margin = new Thickness(10) };

btn4.Click += btnDistribution\_Click;

stplButtons.Children.Add(btn4);

break;

}

}

private void btnDelet\_Click(object sender, RoutedEventArgs e)

{

List<ExpenseIvoices> OrderForRemoving = dgExpenseIvoices.SelectedItems.Cast<ExpenseIvoices>().ToList();

if (MessageBox.Show("Удалить заказы", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

foreach (ExpenseIvoices o in OrderForRemoving)

{

List<ExpenseComposition> list = ClassFrame.db.ExpenseComposition.Where(x => x.IdExpenseIvoices == o.IdExpenseIvoices).ToList();

foreach (ExpenseComposition c in list)

ClassFrame.db.Database.ExecuteSqlCommand("delete Nomenclature.dbo.Movement where idComposition = @com and ArrivalOrExpenditure = 1", new SqlParameter("@com", c.IdExpenseComposition));

ClassFrame.db.ExpenseComposition.RemoveRange(ClassFrame.db.ExpenseComposition.Where(x => x.IdExpenseIvoices == o.IdExpenseIvoices));

}

ClassFrame.db.ExpenseIvoices.RemoveRange(OrderForRemoving);

ClassFrame.db.SaveChanges();

}

catch(Exception ex)

{

MessageBox.Show(ex.Message.ToString());

return;

}

MessageBox.Show("Данные удаленны");

dgExpenseIvoices.ItemsSource = ClassFrame.db.ExpenseIvoices.ToList();

}

}

private void dgExpenseIvoices\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{

if(dgExpenseIvoices.SelectedItem != null)

ClassFrame.frmObj.Navigate(new PageExpenseComposition((ExpenseIvoices)dgExpenseIvoices.SelectedItem));

}

private void btnAddOrder\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageAddOrder(null));

}

private void btnBack\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

private void TxtOrder\_TextChanged(object sender, TextChangedEventArgs e)

{

if (TxtOrder.Text.Count() != 0)

dgExpenseIvoices.ItemsSource = ClassFrame.db.ExpenseIvoices.Where(x => x.Employee.FIO.ToLower().Contains(TxtOrder.Text.ToLower()) || x.Сounterparties.Nazv\_Сounterparties.ToLower().Contains(TxtOrder.Text.ToLower())).ToList();

else dgExpenseIvoices.ItemsSource = ClassFrame.db.ExpenseIvoices.ToList();

}

private void MenuItem\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageAddOrder((ExpenseIvoices)dgExpenseIvoices.SelectedItem));

}

private void btnDistribution\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new DistributionWarehouse((ExpenseIvoices)dgExpenseIvoices.SelectedItem));

}

}

}

# Входные и выходные данные

Входные данные:

|  |  |
| --- | --- |
| **Поля** | **Тип данных** |
|  |  |
| Название детали | Текстовый |
| Название шаблона | Текстовый |
|  |  |
| Название раздела | Текстовый |
| Название пункта | Текстовый |
| Логин | Текстовый |
| Пароль | Текстовый |
| Должность сотрудника | Текстовый |
| Ответ | Текстовый |
| Серийный номер | Числовой |
| Заводской номер | Текстовый |
| Дата приемки | Дата |

**Выходная информация:**

Выходной информацией является экспорт списка/таблицы в виде нового Excel-файла.

# Сообщения

Участок кода с подсчётом количества записей в таблице, работающий некорректно:

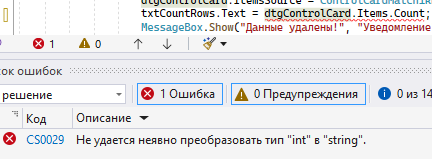


Рис.1 «Неявное преобразование типов»

Исправленный участок кода с подсчётом количества записей в таблице, работающий корректно:

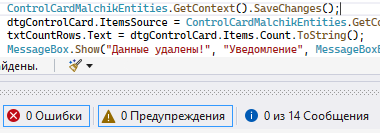


Рис.2 «Преобразование типа «int» в строку»

Участок кода с переходом на страницу добавления сотрудника, работающий некорректно:

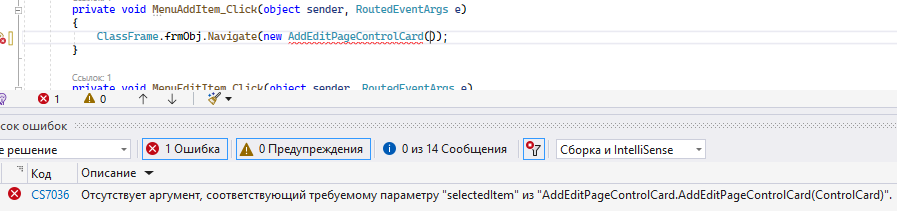


Рис.3 «Отсутствие аргумента, требуемой страницы»

Участок кода с переходом на страницу добавления сотрудника, работающий корректно:

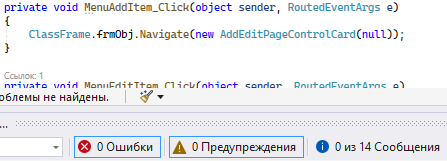


Рис.4 «Добавление пустого аргумента, требуемой странице»