**ТЕСТИРОВАНИЕ ПРОГРАММЫ ТЕЛЕФОННЫЙ СПРАВОЧНИК**

**Функциональное тестирование:**

**Цель:** Проверка функциональности приложения «Телефонный справочник» для обеспечения соответствия требованиям.

**Область действия:** Функциональное тестирование будет охватывать следующие функциональные возможности:

* Добавление новых контактов
* Редактирование существующих контактов
* Удаление контактов
* Поиск контактов по части ФИО
* Сохранение изменений
* Экспорт данных в CSV
* Сортировка и фильтрация

**Критерии успеха:**

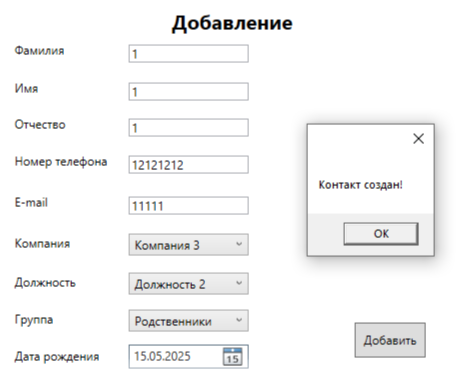
• Все тестовые сценарии проходят успешно.

• Приложение работает без сбоев или исключений.

• Результаты тестирования соответствуют ожидаемым результатам.

**Тестовые сценарии:**

Добавление нового контакта:



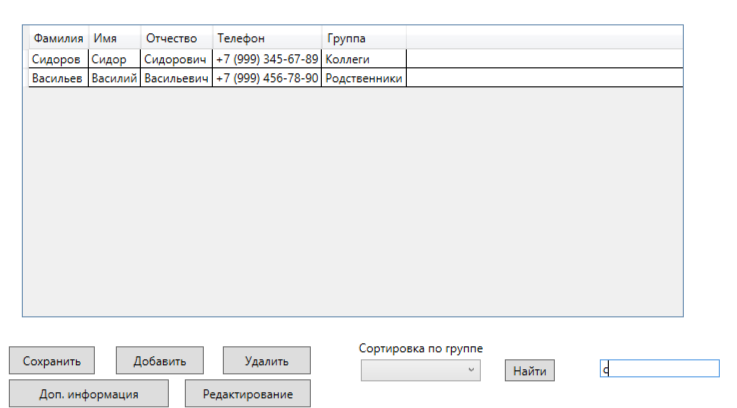
Редактирование существующего контакта:

Функция редактирования реализована, но не работает.

Удаление контакта:

Функция удаления контакта реализована, но не работает.

Поиск контактов по части ФИО:

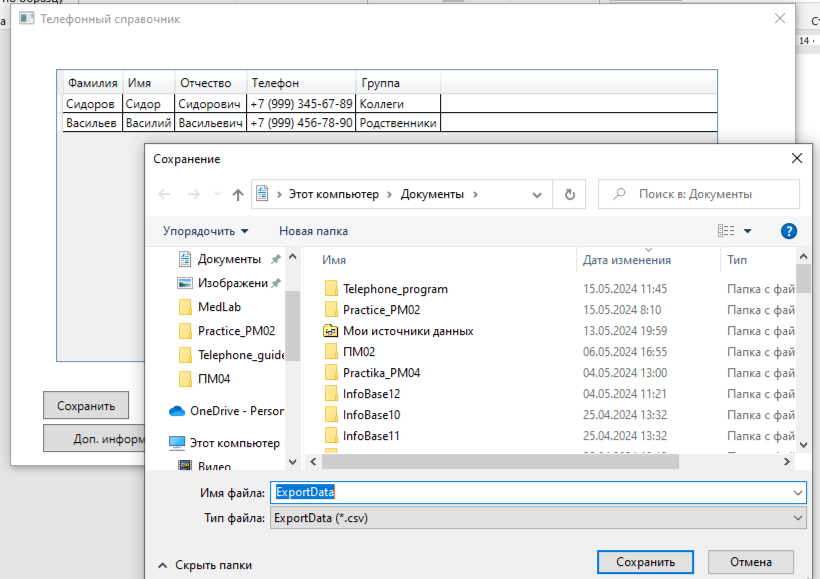


Поиск работает корректно.

Сохранение изменений:

Сохранение изменений реализовано автоматически при редактировании записей.

Экспорт данных в CSV:



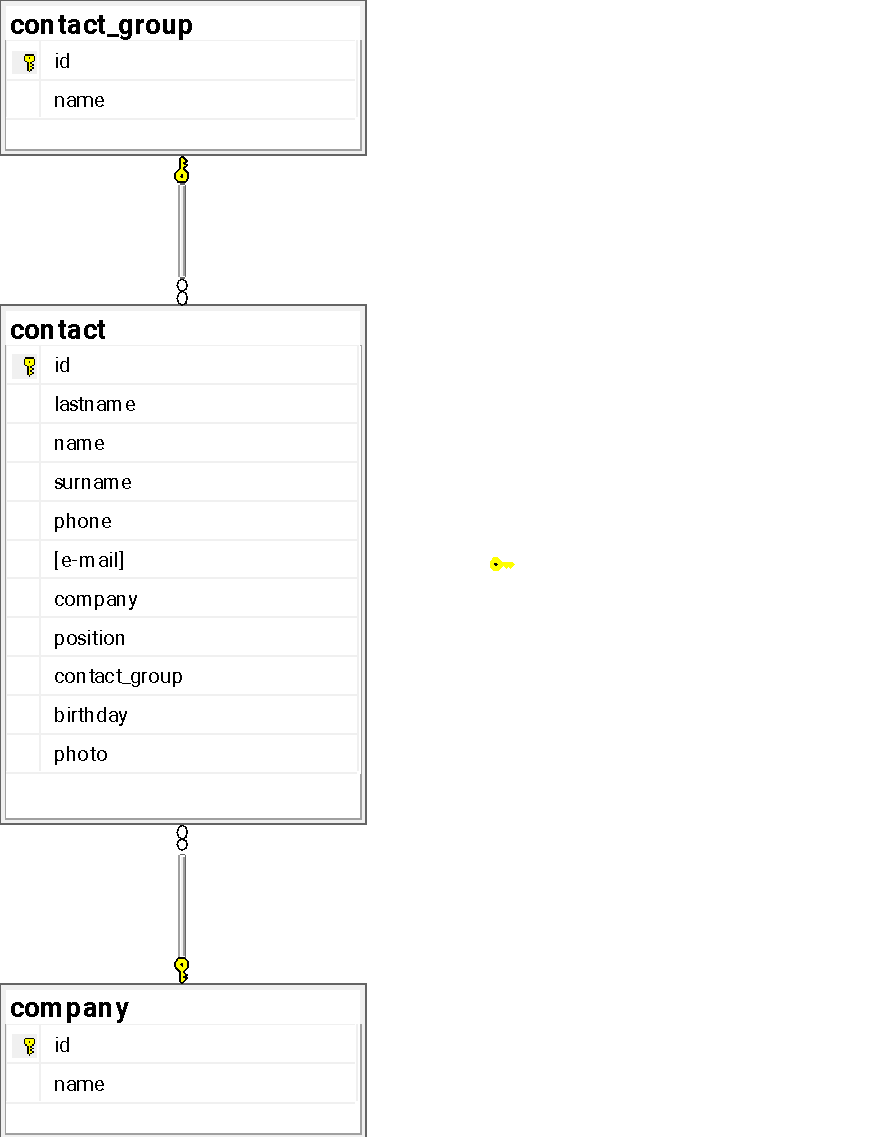
Экспорт работает корректно.

Сортировка и фильтрация работают корректно.

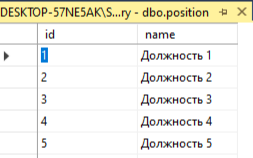
**Данные тестирования:**

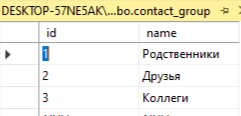
Для тестирования будут использоваться тестовые данные, включающие различные имена, номера телефонов и другую информацию о контактах.

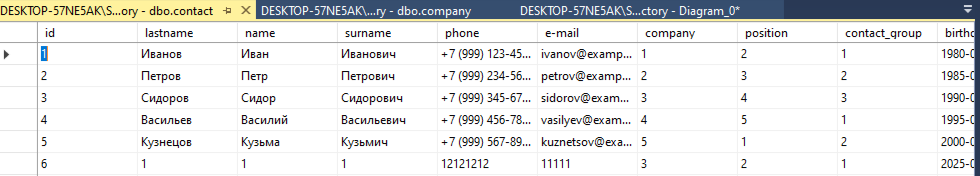
Диаграмма бд:

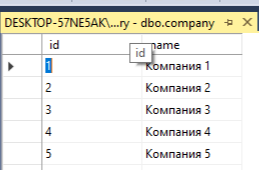


Данные в таблицах:









**Ожидаемые результаты:**

Ожидалось, что приложение будет работать корректно, но некоторые функции не работают.

**Тестирование интерфейса:**

**Цель:** проверить удобство использования, навигацию и общий дизайн пользовательского интерфейса приложения «Телефонный справочник».

**Область действия:** Тестирование пользовательского интерфейса будет охватывать следующие аспекты:

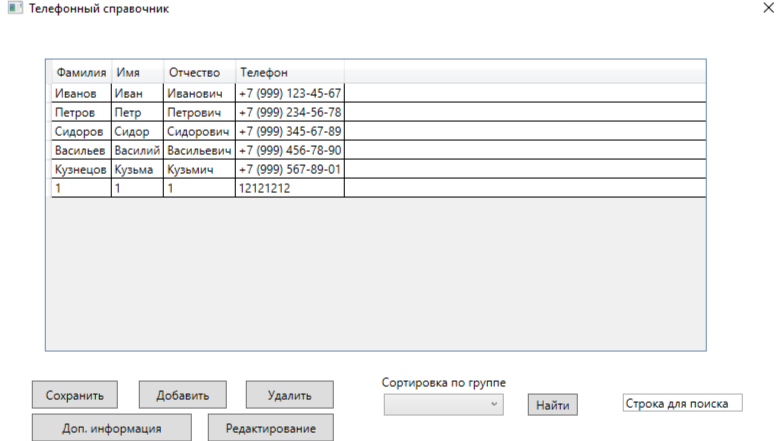
* Удобство использования: Легкость использования приложения для пользователей с различным уровнем знаний.
* Навигация: Четкость и простота навигации по различным функциям приложения.
* Дизайн: Привлекательность и соответствие общепринятым стандартам дизайна пользовательского интерфейса.

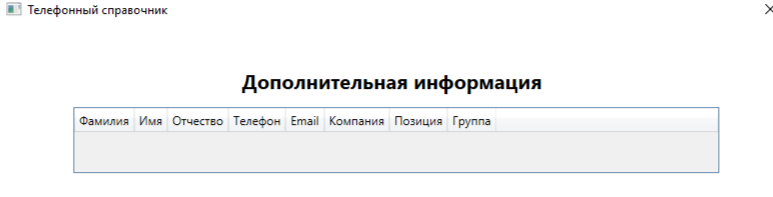
**Критерии успеха:**

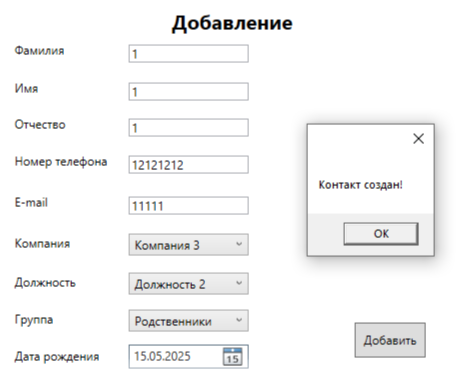
* Пользователи могут легко найти и использовать все функции приложения.
* Навигация по приложению интуитивно понятна и не вызывает затруднений.
* Пользовательский интерфейс эстетичен и соответствует современным тенденциям дизайна.

**Тестовые сценарии:**

**Удобство использования:**

****

****



Интерфейс очень простой и понятный в использовании. Отсутствует возможность изменения размеров окон.

**Навигация:**

Навигация реализована логически и интуитивно понятно. Для каждой из функций есть отдельные элементы. Перемещение между окнами плавное и не вызывает проблем.

**Дизайн:**

• Приложение имеет простой и интуитивно понятный пользовательский интерфейс с использованием современных элементов управления и дизайна.

• Основной экран содержит список контактов, с возможностью добавления, редактирования и удаления контактов и тд.

• Панель поиска позволяет быстро находить контакты по части ФИО.

• Дополнительных функций, таких как экспорт данных в CSV, можно получить доступ через меню приложения.

**Данные тестирования:**

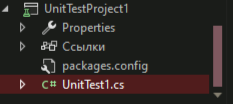
Тестирование проводилось с использованием реальных пользователей с различным уровнем компьютерной грамотности.

**Ожидаемые результаты:**

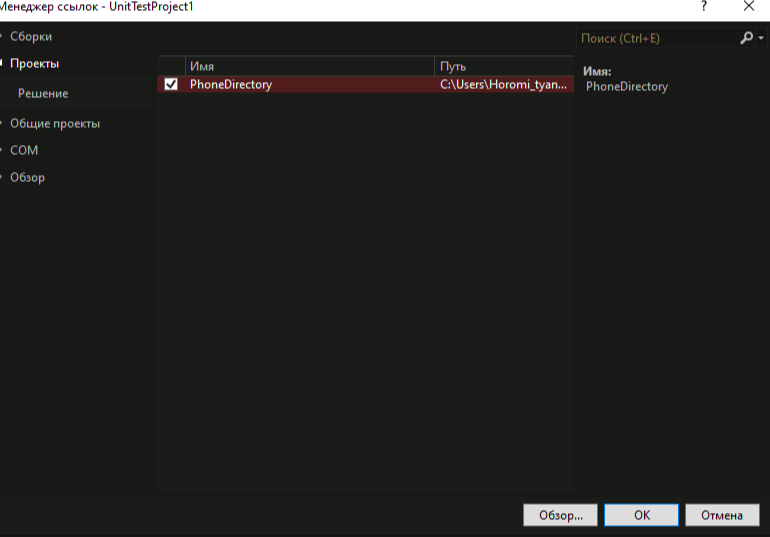
Пользовательский интерфейс простой, удобный, интуитивно понятный и эстетически привлекательный.

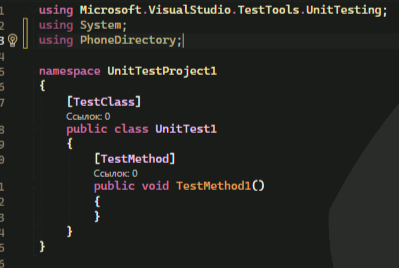
**Юнит тестирование:**

Создание юнит теста:



Добавление ссылки на проект в тест:





Код юнит теста:

using Microsoft.VisualStudio.TestTools.UnitTesting;

using PhoneDirectory;

namespace UnitTestProject1

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void TestMethod1()

{

// Arrange

var contact\_group = new Contact\_group { Id = 1, Name = "Группа 1" };

var position = new Position { Id = 2, Name = "Должность 2" };

var company = new Company { Id = 3, Name = "Компания 3" };

var contact = new Contact

{

Contact\_group = contact\_group,

Position = position,

Company = company,

Lastname = "Иванов",

Name = "Иван",

Surname = "Иванович",

Phone = "555-1234"

};

using (var db = new PhoneDirectoryEntities1())

{

db.contact\_group.Add(contact\_group);

db.position.Add(position);

db.company.Add(company);

db.contact.Add(contact);

db.SaveChanges();

}

var viewModel = new ViewModel();

// Act

viewModel.Text();

// Assert

Assert.IsNotNull(viewModel.DgCont.ItemsSource);

Assert.IsTrue(viewModel.DgCont.ItemsSource.Count > 0);

Assert.AreEqual("Иванов", ((dynamic)viewModel.DgCont.ItemsSource[0]).Фамилия);

Assert.AreEqual("Иван", ((dynamic)viewModel.DgCont.ItemsSource[0]).Имя);

Assert.AreEqual("Иванович", ((dynamic)viewModel.DgCont.ItemsSource[0]).Отчество);

Assert.AreEqual("555-1234", ((dynamic)viewModel.DgCont.ItemsSource[0]).Телефон);

}

}

internal class Contact : contact

{

public Contact\_group Contact\_group { get; set; }

public Position Position { get; set; }

public Company Company { get; set; }

public string Lastname { get; set; }

public string Name { get; set; }

public string Surname { get; set; }

public string Phone { get; set; }

}

internal class Company : company

{

public int Id { get; set; }

public string Name { get; set; }

}

internal class Position : position

{

public int Id { get; set; }

public string Name { get; set; }

}

internal class Contact\_group : contact\_group

{

public int Id { get; set; }

public string Name { get; set; }

}

}