МИНИСТЕРСТВО ОБРАЗОВАНИЯ КИРОВСКОЙ ОБЛАСТИ

Кировское областное государственное профессиональное образовательное

бюджетное учреждение

«Слободской колледж педагогики и социальных отношений»

**ОТЧЕТ**

**по производственной практике**

**ПМ.02 «Осуществление интеграции программных модулей»**

Подготовил студент

Ермаков Павел Андреевич

Группа 21П-1

Специальность 09.02.07 Информационные системы и программирование

Директор

КОГПОБУ «Слободской колледж педагогики и социальных отношений  
 Наименование организации

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/О.М. Шеренцова

Подпись расшифровка

М.П.

Руководитель практики от организации

Подпись расшифровка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Калинин Арсений Олегович

Подпись расшифровка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Калинин Арсений Олегович

Руководители практики от колледжа:

Калинин Арсений Олегович

2024 год

**СОДЕРЖАНИЕ**

1. ВВЕДЕНИЕ
2. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ
3. РУКОВОДСТВО ОПЕРАТОРА
4. РАБОТА В СИСТЕМЕ КОНТРОЛЯ ВЕРСИЙ
5. РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ
6. ОТЛАДКА ПРОГРАММНОГО МОДУЛЯ
7. ЗАКЛЮЧЕНИЕ
8. ПРИЛОЖЕНИЯ

**ВВЕДЕНИЕ**

Практику по данному модулю я проходил в КОГПОБУ «Слободской колледж педагогики и социальных отношений» с 6 по 17 мая. Целью данного модуля практики является осуществление интеграции программных модулей разрабатываемых нами систем.

Некоторые задачи практики:

* Создать приложение для медицинской лаборатории
* Создать приложение для телефонного справочника

**АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

Медицинская лаборатория предоставляет специализированные услуги по проведению исследований биоматериалов для поликлиник города. Было сделано множество диаграмм и техническое задание.

ERD-Диаграмма

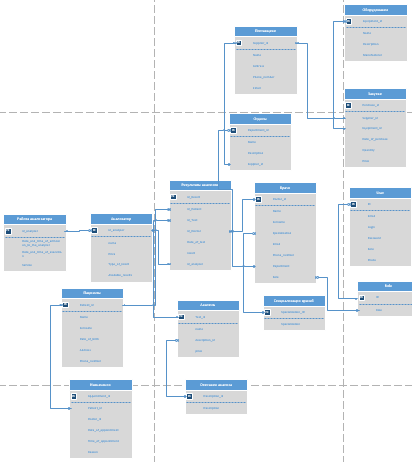


Диаграмма вариантов использования

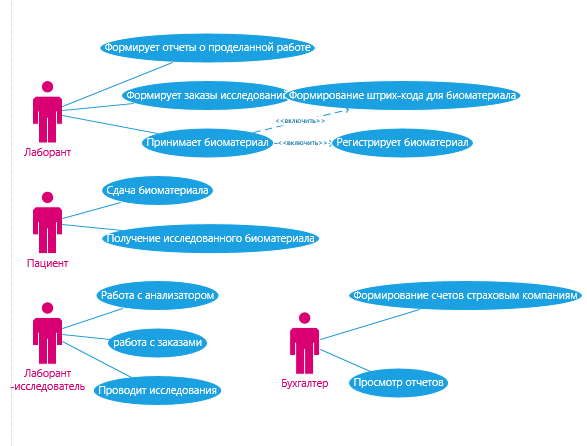


Диаграмма деятельности

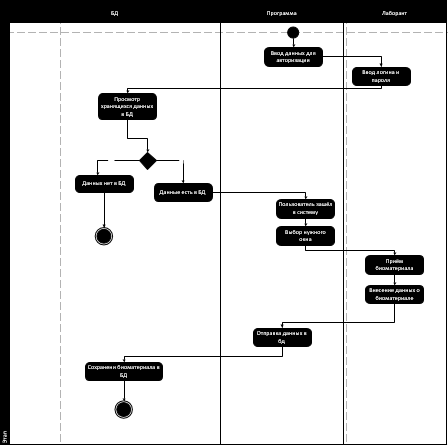
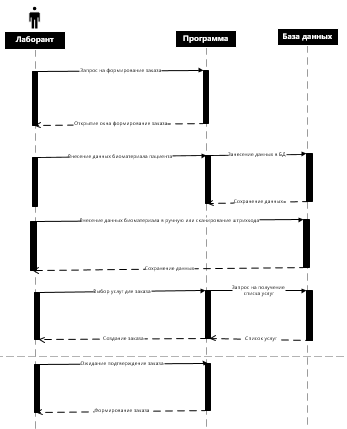
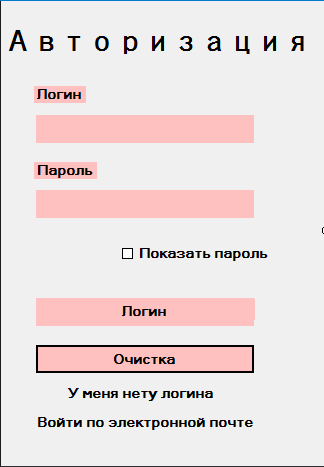


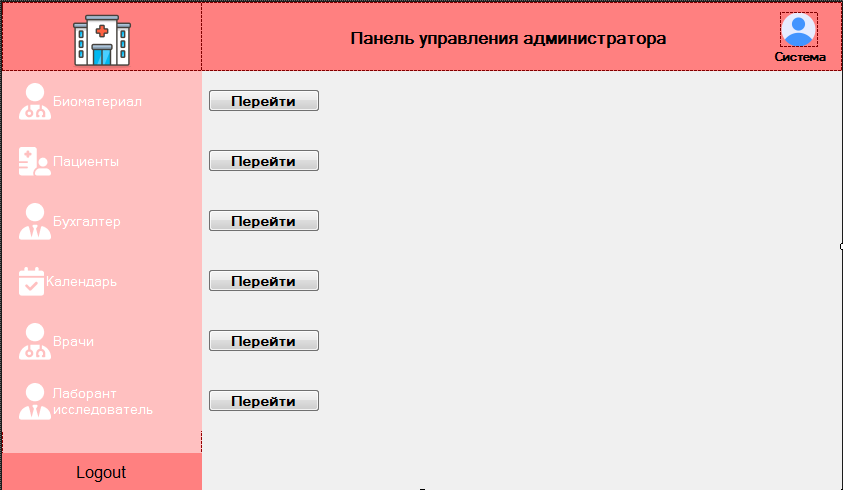
Диаграмма последовательности



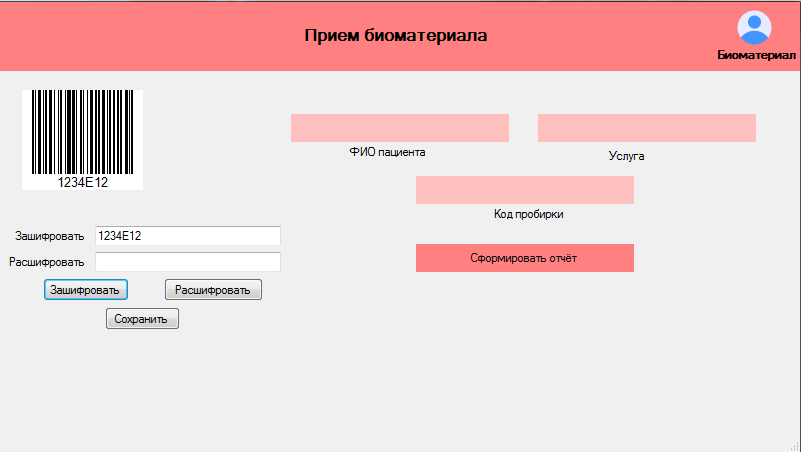
**РУКОВОДСТВО ОПЕРАТОРА**



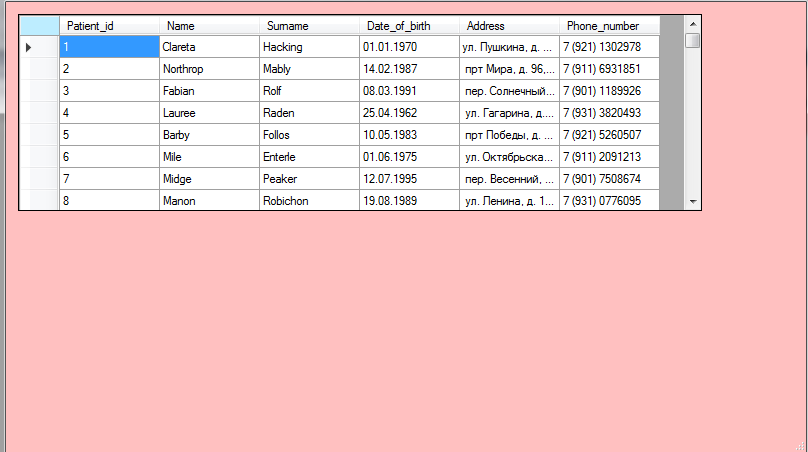
Окно авторизации представлено двумя полями для ввода логина и пароля



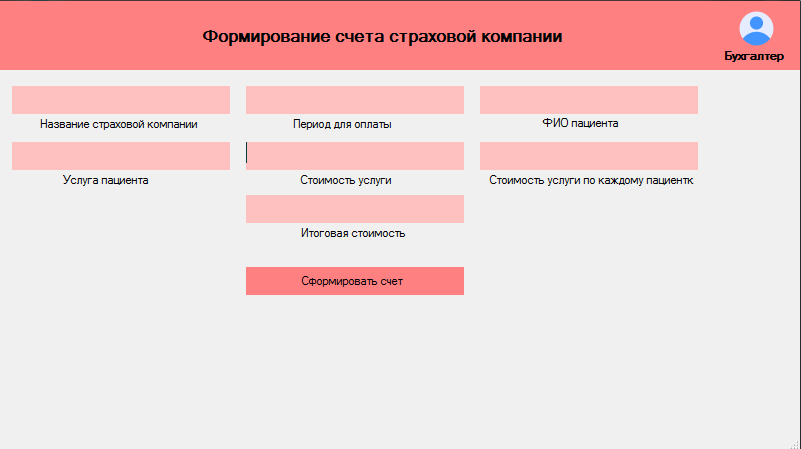
Главное окно



Окно приема биоматериала



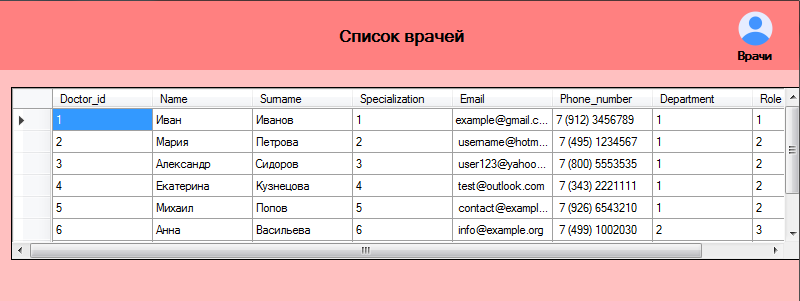
Окно списка пациентов



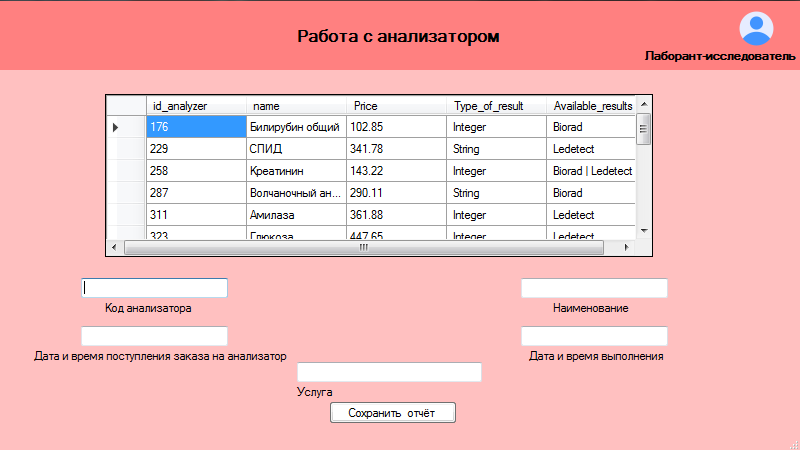
Окно для формирования счета страховым компаниям



Календарь



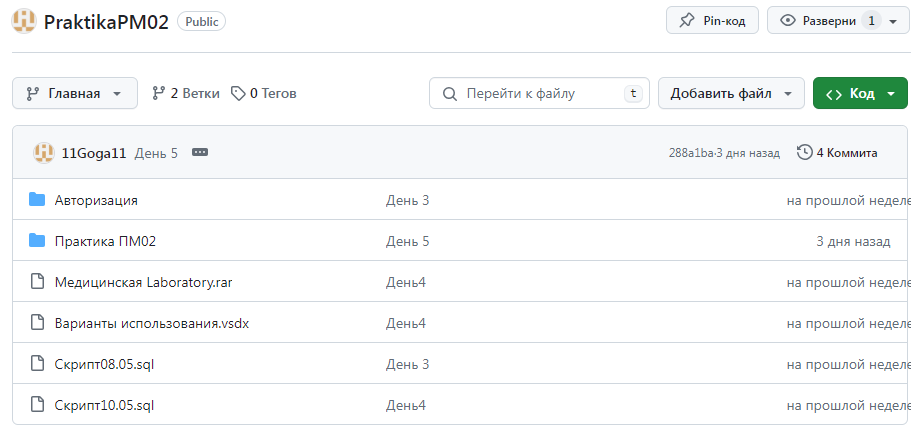
Окно списка врачей



Окно работы с анализатором

**РАБОТА В СИСТЕМЕ КОНТРОЛЯ ВЕРСИЙ**

За все время практики я работал с системой контроля версий GitHub. Я создал несколько репозиториев, в которые загружал все выполняемые мной задания:

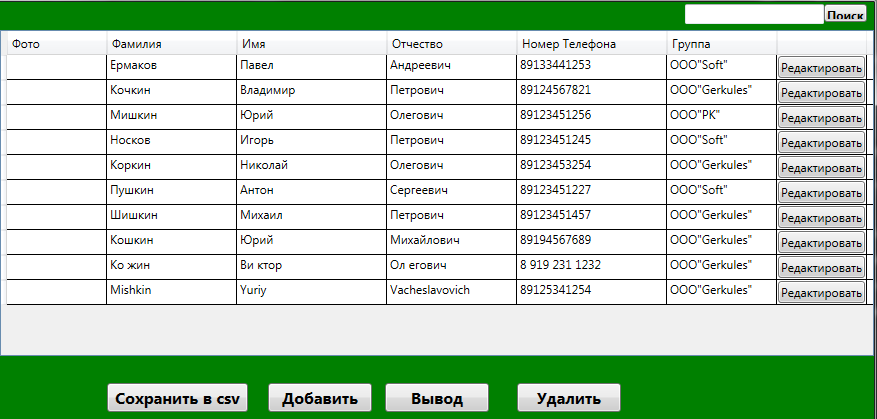


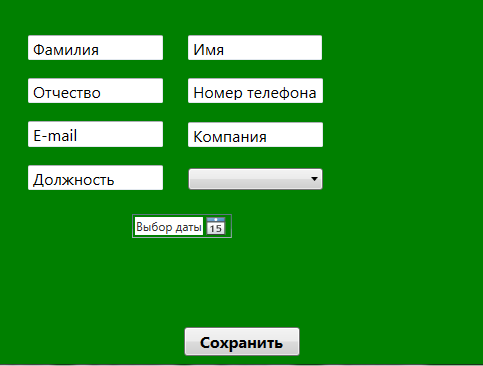
**РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ**

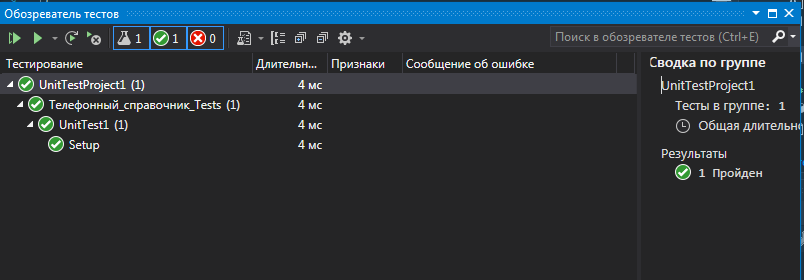
**Тестирование программы Телефонный справочник**

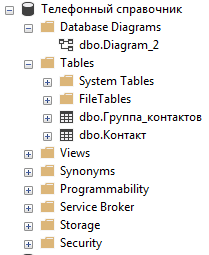
**Функциональное тестирование:**

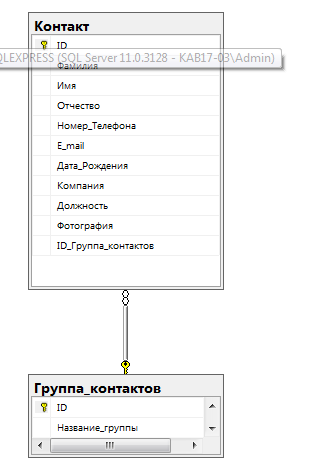
**Цель:** Проверка функциональности приложения «Телефонный справочник» для обеспечения соответствия требованиям.









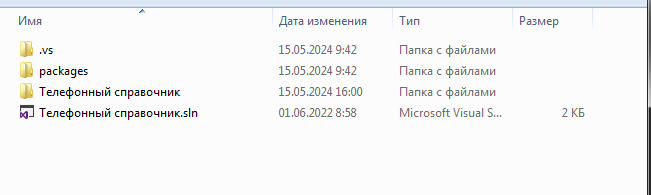


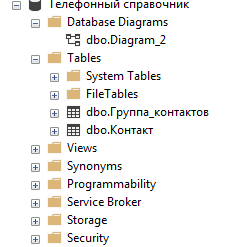
Код Unit.Test

|  |
| --- |
| using Microsoft.VisualStudio.TestTools.UnitTesting;  using System;  using System.Collections.Generic;  using Телефонный\_справочник;  namespace Телефонный\_справочник\_Tests  {  [TestClass]  public class UnitTest1  {  private Mock<IContactRepository> \_mockContactRepository;  private PhoneBookService \_phoneBookService;  [TestMethod]  public void Setup()  {  \_mockContactRepository = new Mock<IContactRepository>();  \_phoneBookService = new PhoneBookService(\_mockContactRepository.Object);  }  public void AddContact\_WhenCalled\_ShouldAddContactToRepository()  {  // Arrange  var contact = new Contact { Name = "John Doe", PhoneNumber = "1234567890" };  // Act  \_phoneBookService.AddContact(contact);  }  public void GetAllContacts\_WhenCalled\_ShouldReturnAllContactsFromRepository()  {  // Arrange  var contacts = new List<Contact>  {  new Contact { Name = "John Doe", PhoneNumber = "1234567890" },  new Contact { Name = "Jane Doe", PhoneNumber = "0987654321" }  };  \_mockContactRepository.Setup((contacts));  // Act  var result = \_phoneBookService.GetAllContacts();  // Assert  Assert.AreEqual(contacts, result);  }  }  internal class Times  {  internal static object Once;  }  internal class Mock<T>  {  internal IContactRepository Object;  internal void Setup(Func<object, object> p)  {  throw new NotImplementedException();  }  internal void Setup(List<Contact> contacts)  {  throw new NotImplementedException();  }  internal void Verify(Func<object, object> p, object once)  {  throw new NotImplementedException();  }  }  internal class Contact  {  public string Name { get; set; }  public string PhoneNumber { get; set; }  }  internal class PhoneBookService  {  private IContactRepository @object;  public PhoneBookService(IContactRepository @object)  {  this.@object = @object;  }  internal void AddContact(Contact contact)  {  throw new NotImplementedException();  }  internal List<Contact> GetAllContacts()  {  throw new NotImplementedException();  }  }  internal interface IContactRepository  {  void Add(Contact contact);  void GetAllContacts(List<Contact> contacts);  }  } |

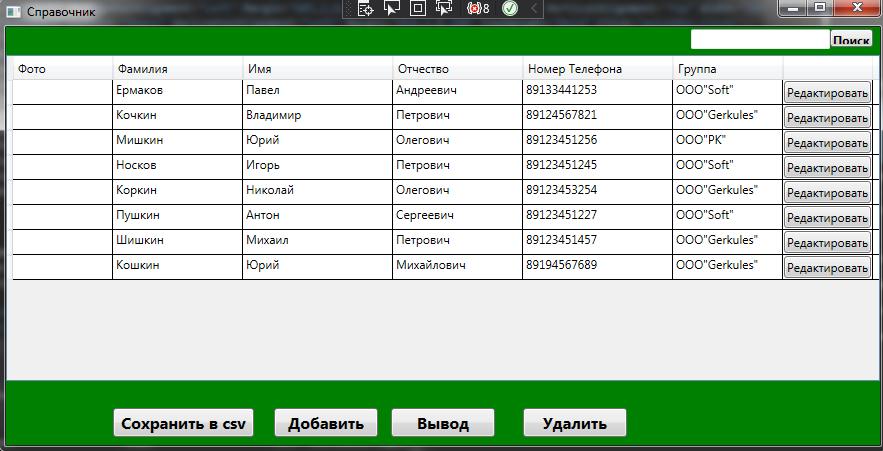
Тестирование приложения “Телефонный справочник”

1. Подготовка к тестированию

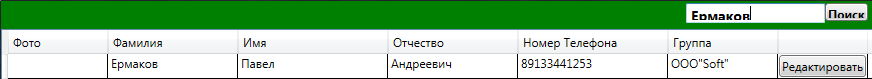




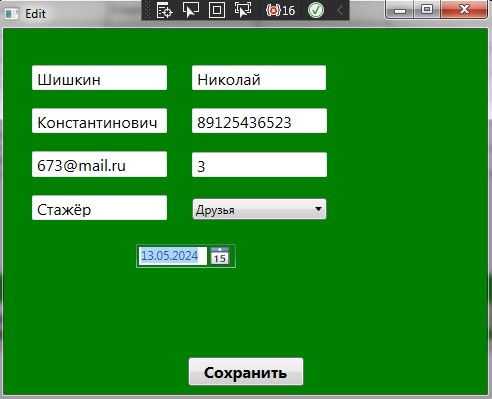
1. Проверка главной страницы

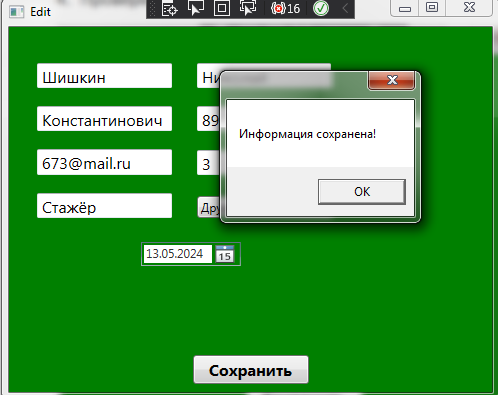


1. Проверка поиска контактов



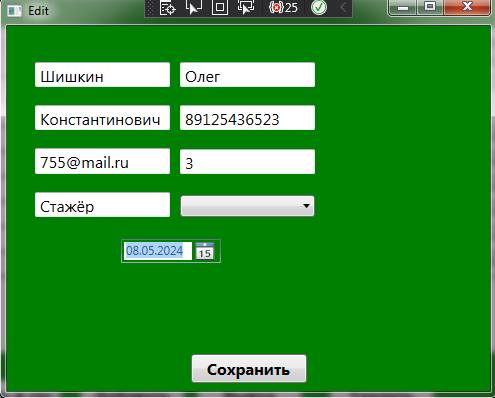
1. Проверка добавления контакта

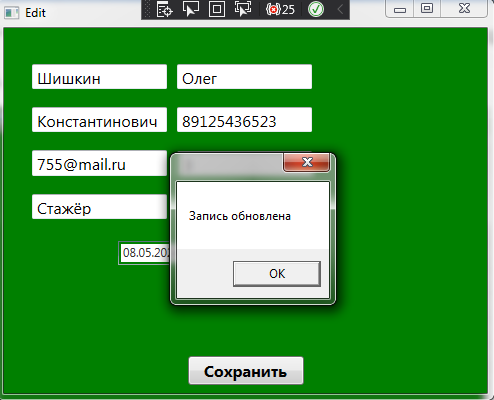




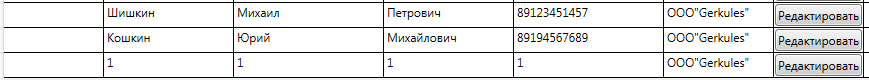


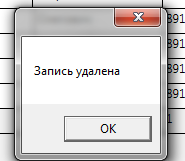
1. Проверка редактирования контакта

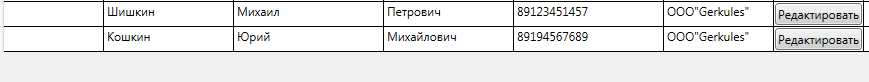




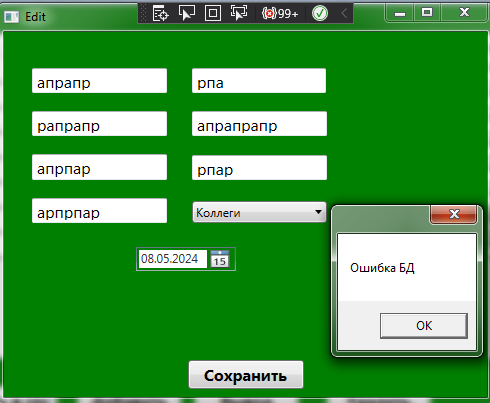
1. Проверка удаления контакта



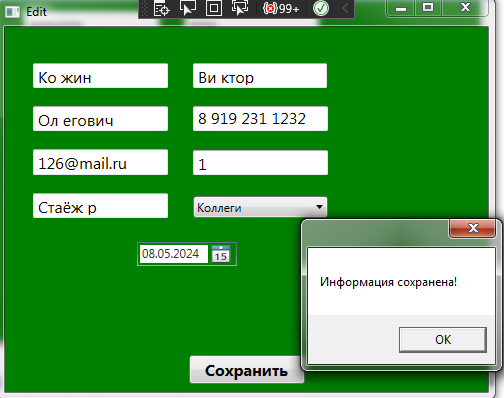




1. Проверка состояния ошибки

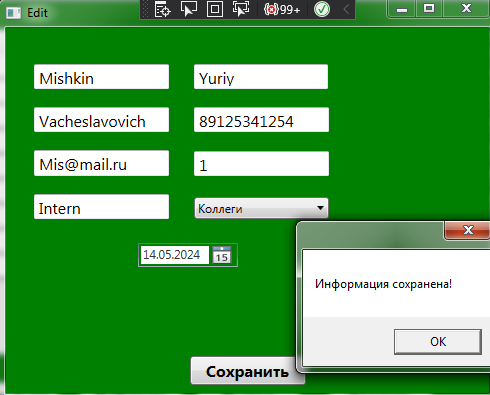


1. Проверка работы с пробелами и специальными символами





1. Проверка работы приложения в разных языковых настройках





**ЗАКЛЮЧЕНИЕ**

Учебная практика дала мне ценный опыт, который помог расширить мои профессиональные знания и навыки. Во время практики, я изучил осуществление интеграции программных модулей и вспомнил, то чему учился на протяжении учебного года.