МИНИСТЕРСТВО ОБРАЗОВАНИЯ КИРОВСКОЙ ОБЛАСТИ

Кировское областное государственное профессиональное образовательное

бюджетное учреждение

«Слободской колледж педагогики и социальных отношений»

**ОТЧЕТ**

**по учебной практике**

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

Студента

Платунов Павел Андреевич

Группа 21П-1

Специальность

09.02.07 Информационные системы и программирование

Руководитель практики от колледжа

Калинин Арсений Олегович

\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись расшифровка

2024 год

Оглавление

[Введение 3](#_Toc166837926)

[Анализ предметной области 4](#_Toc166837927)

[Руководство оператора 13](#_Toc166837928)

[Работа в системе контроля версий 22](#_Toc166837929)

[Разработка тестовых наборов и тестовых сценариев 23](#_Toc166837930)

[MedLaboratory 23](#_Toc166837931)

[Телефонный справочник 38](#_Toc166837932)

[Отладка программного модуля 48](#_Toc166837933)

[Заключение 50](#_Toc166837934)

[Приложения 51](#_Toc166837935)

# Введение

Практика по учебной практике по модулю "Осуществление интеграции программных модулей" проходила в Слободском колледже педагогики и социальных отношений с 6 мая 2024 по 17 мая 2024.

Целью прохождения учебной практики по модулю "Осуществление интеграции программных модулей " является приобретение практических навыков по интеграции программных модулей в разработанные системы их отладка и тестирование на наличии ошибок и некорректной работы. А также устранение выявленных в ходе тестирования ошибок и багов.

# Анализ предметной области

Основной задачей является разработка информационной системы для медицинской лаборатории № 20 города Санкт-Петербурга. Медицинская лаборатория № 20 работает на прием биоматериала из поликлиник города, проводит специализированные исследования с помощью новейшего оборудования, ведет базу данных исследований, предоставляет отчеты по проведенным исследованиям.

В системе будут состоять следующие члены персонала:

• лаборант

• лаборант-исследователь

• бухгалтер

• администратор

Каждый из членов персонала может выполнять определенные действия в системе:

• лаборант может принять биоматериал, получая штрих-код кода пробирки, формировать отчеты

• лаборант-исследователь может работать с анализатором

• бухгалтер может просмотреть отчеты, сформировать счет страховой компании

• администратор может сформировать отчеты, проконтролировать всех

Также для системы должны быть сформированы следующие ограничения:

При входе в учетную запись лаборанта и лаборанта-исследователя должен быть виден таймер (часы:минуты), который фиксирует время сеанса пользователя. Сеанс пользователя не должен превышать 2 ч 30 минут, так как через каждые 2 ч 30 минут необходимо выполнить кварцевание помещений. За 15 минут до окончания времени сеанса должно появиться сообщение об окончании времени сеанса. По окончании времени сеанса реализуйте выход из учетной записи и блокировку входа на 30 минут. Для удобства проверки экспертной группой - укажите время сеанса – 10 минут, появление сообщения – за 5 минут до окончания времени сеанса, блокировка входа – 1 минута.

После первой попытки неуспешной авторизации система выдает сообщение о неуспешной авторизации, а затем помимо ввода логина и пароля просит ввести captcha, состоящую из 4 символов (цифры и буквы латинского алфавита) и графического шума. CAPTCHA - должна содержать минимум 4 символа (буква или цифра), которые выведены не в одной линии. Символы должны быть либо перечеркнуты либо наложены друг на друга. Реализуйте возможность повторной генерации captcha, если пользователю непонятны символы из-за шума. После попытки неудачной авторизации с вводом captcha, система блокирует возможность входа на 10 секунд.

Функции системы:

В окне истории посещений необходимо реализовать просмотр всей истории входа, а также фильтрацию по логину пользователя. Кроме этого, необходимо добавить сортировку по дате попытки входа. Каждая запись истории должна содержать следующие данные: время, логин пользователя, успешная или ошибочная попытка входа.

При организации хранения данных вам необходимо учесть запрет на полное удаление данных, реализовав возможность отправки данных в архив. Кроме того, необходимо учесть, что данные о заказе не могут быть отправлены в архив, если не выполнена хотя-бы одна услуга в заказе.

Лаборант может выполнять поиск клиента по ФИО с помощью метода Левенштейна.

Лаборант-исследователь в окне должен видеть анализаторы лаборатории с возможными исследованиями на них. Выбрав один из представленных анализаторов, пользователь видит список невыполненных услуг, доступных на данном анализаторе. Некоторые услуги могут быть выполнены на нескольких анализаторах. Лаборант-исследователь выбирает услугу и отправляет биоматериал на анализатор с помощью нажатия специальной кнопки “отправить на исследование” около услуги.

Бухгалтер должен сформировать счет на каждую страховую компанию за определенный период времени и сохранить его в формате pdf.

Система должна иметь возможность формировать отчеты и график контроля качества.

В базе данных должны храниться следующие данные:

• услуги лаборатории (наименование, стоимость, код услуги, срок выполнения, среднее отклонение)

• данные пациентов (логин, пароль, ФИО, дата рождения, серия и номер паспорта, телефон, e-mail, номер страхового полиса, тип страхового полиса, страховая компания)

• данные о страховых компаниях (название страховой компании, адрес, ИНН, р/с, БИК)

• заказ (дата создания, которые входят в заказ, услуги, статус заказа, статус услуги в заказе, время выполнения заказа (в днях))

• оказанная услуга (услуга, когда и кем была и на каком анализаторе)

• данные о работе анализатора (дата и время поступления заказа на анализатор, дата и время выполнения (в секундах) услуг на анализаторе)

• данные лаборантов (логин, пароль, ФИО, последняя дата и время входа, набор услуг, которые он может оказывать)

• бухгалтер (логин, пароль, ФИО, последняя дата и время входа, набор услуг, выставленные счета страховым компаниям)

• администратор (логин и пароль)

• история входа в систему

При выполнение данный практики стоит цель создать базу данных MedLaboratory и приложение для работы с ней. Также необходимо создать телефонный справочник и две программы для решения ЗЛУ графическим (Рис. 1) и симплексным методом (Рис. 2). После создания приложений необходимо их отладить, протестировать и исправить ошибки.

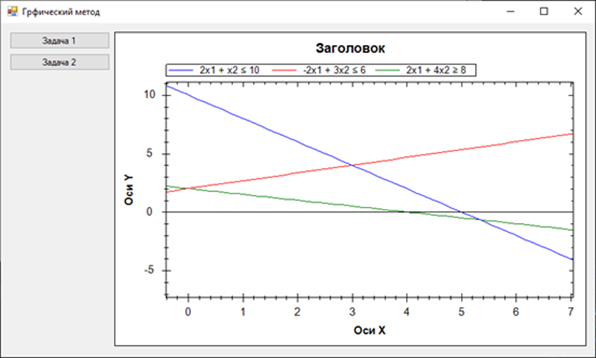


Рисунок 1 - Приложение для решения ЗЛУ графическим методом

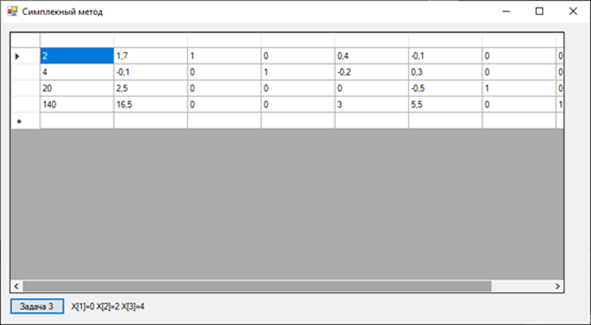


Рисунок 2 - Приложение для решения ЗЛУ симплексным методом

Для большего понимания создаваемой системы MedLaboratory я построил UML диаграмму вариантов использования (Диаграмма 1).



Диаграмма 1 - UML диаграмма вариантов использования

Для разработки системы мною были созданы следующие диаграммы: диаграмма архитектура систем (Диаграмма 2) для понятия движения данных в системе, диаграмма деятельности (Диаграмма 3) для понятия работы приложения, диаграмма классов (Диаграмма 4) для построения базы данных и функций программы, диаграмма последовательности (Диаграмма 5) добавления биоматериала.



Диаграмма 2 - Диаграмма архитектура систем



Диаграмма 3 - Диаграмма деятельности



Диаграмма 4 - Диаграмма классов



Диаграмма 5 - Диаграмма последовательности

После анализа данной предметной области написал техническое задание (Приложение 1) и создал базу данных в Management Studio (Рис. 3).

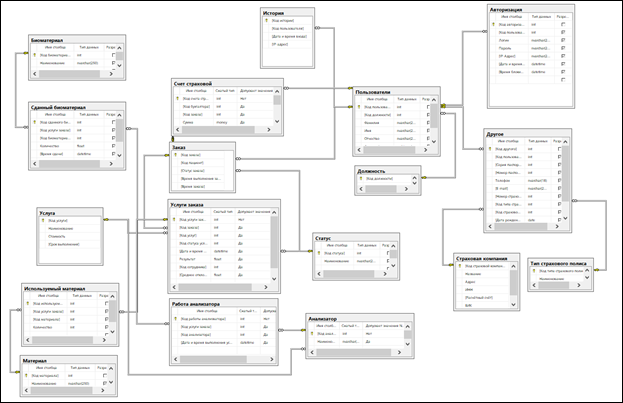


Рисунок 3 - Схема базы данных MedLaboratory

Таже мною была разработана ER-диаграмма аптек (Диаграмма 6).



Диаграмма 6 - ER-диаграмма аптек

# Руководство оператора

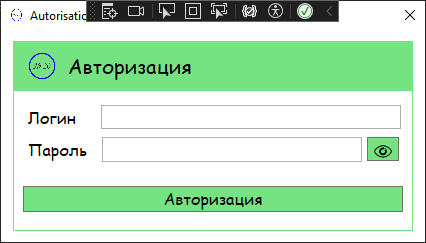
****

Рисунок 4 - Окно авторизации

Окно авторизации (Рис. 4) представлено двумя полями для логина и пароля. Окно поддерживает режим отображения введенного пароля. При повторном вводе неправильного логина или пароля будет выведено окно капчи.



Рисунок 5 - Окно прохождения капчи

Окно прохождения капчи (Рис. 5) предусматривает генерацию капчи, текст на которой читаем для человека, также есть возможность пересоздания капчи при неправильном вводе или при нажатии на соответствующую кнопку.

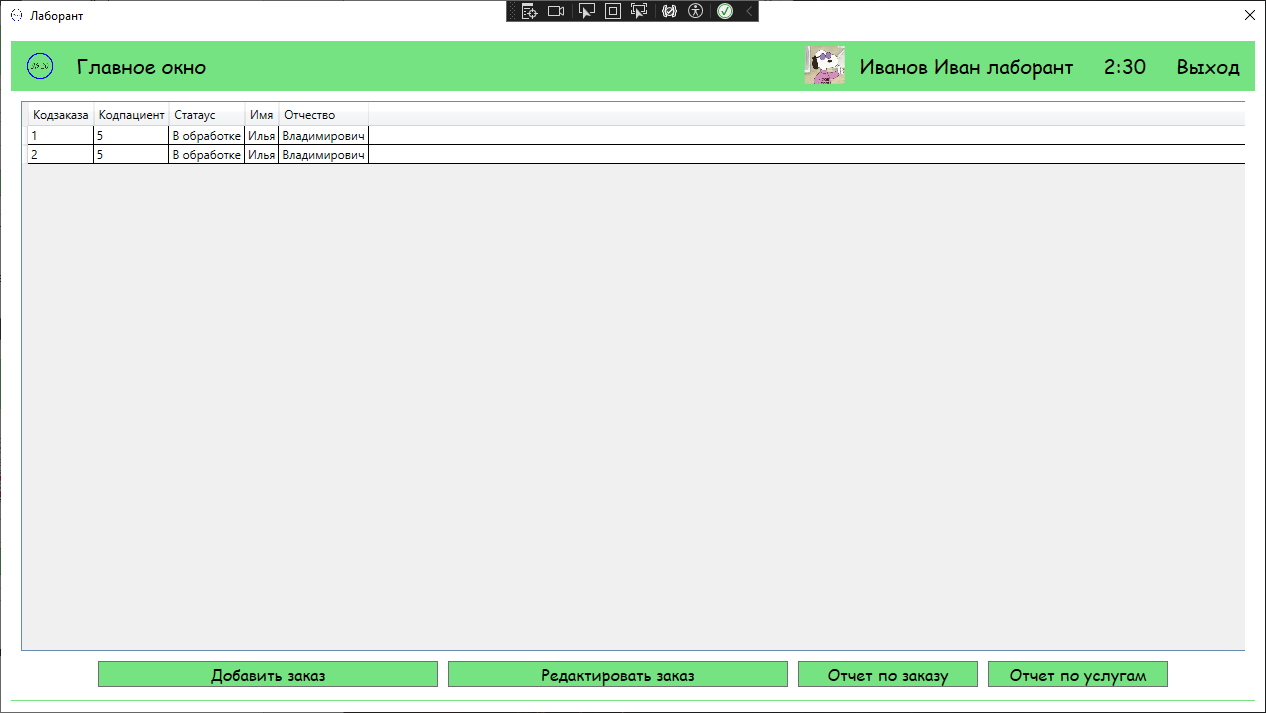


Рисунок 6 - Окно лаборанта

Окно лаборанта служит (Рис. 6) предназначено для работы с биоматериалами, в главном окне пользователь может добавить и отредактировать заказ, а также сформировать отчет.

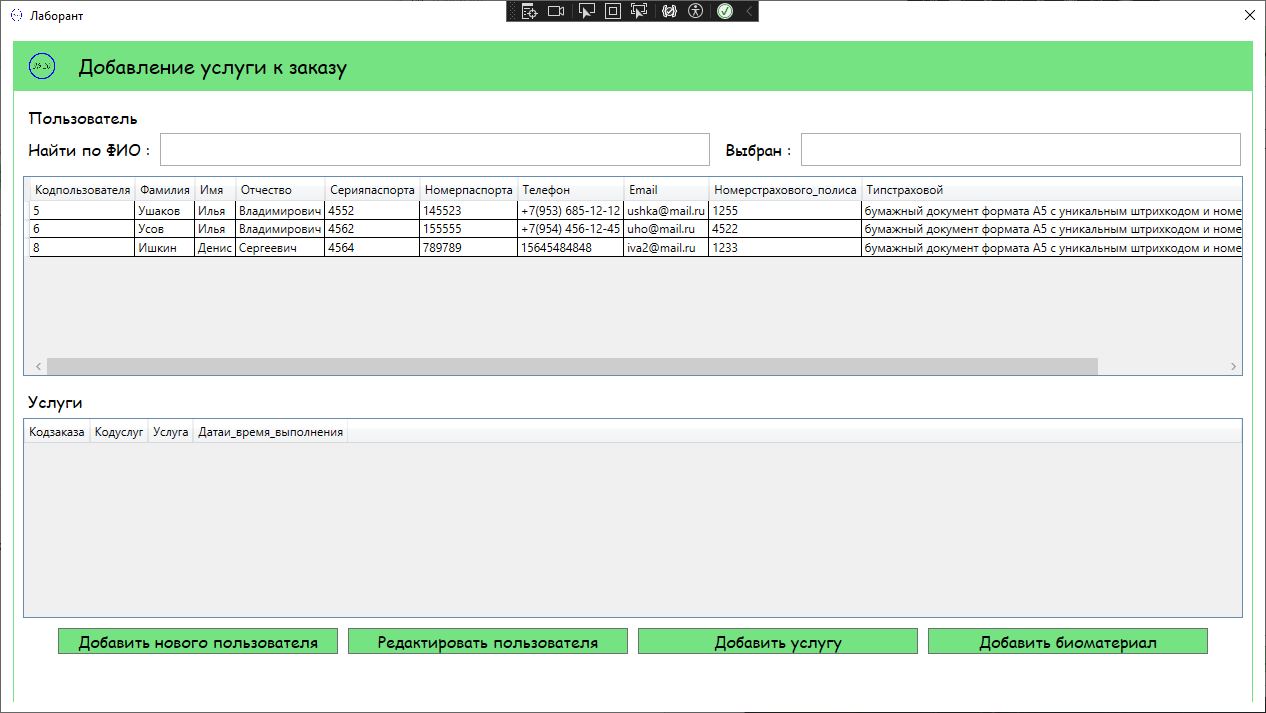


Рисунок 7 - Окно работы с заказа

Окно работы с заказом (Рис. 7) предусматривает поиск пациента по ФИО, его выбор, редактирование по необходимости, а также добавление нового пациента если такового нет в системе, после чего можно добавить услугу и добавить биоматериал.

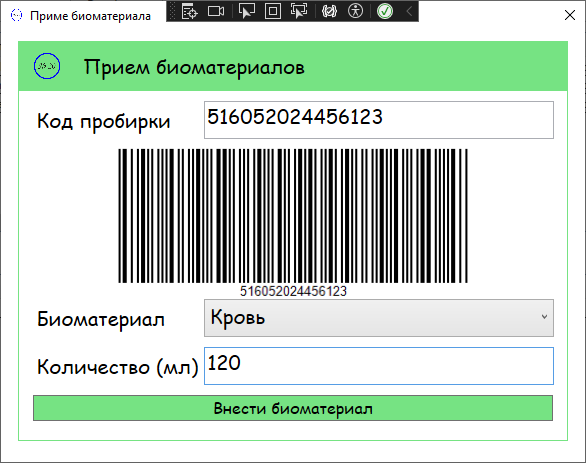


Рисунок 8 - Окно работы с биоматериалом

В окне работы с материалом (Рис. 8) можно указать код пробирки, который сформирует штрих код, ввести тип и количество биоматериала. После сохранения данных автоматически формируется pdf файл со штрих кодом.

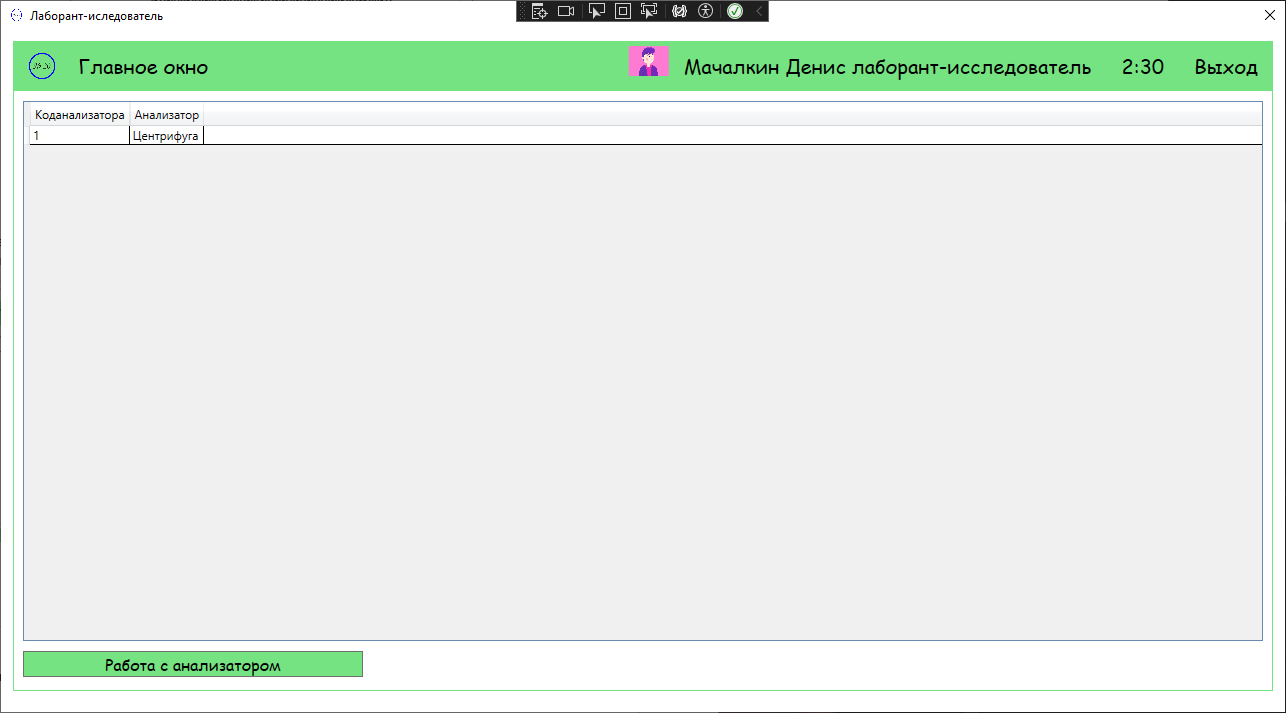


Рисунок 9 - Окно лаборанта-исследователя

Окно лаборанта-исследователя (Рис. 9) предназначено для работы с анализаторами, в данном окне лаборант-исследователь выбирает анализатор после чего открывается окно работы с ним.

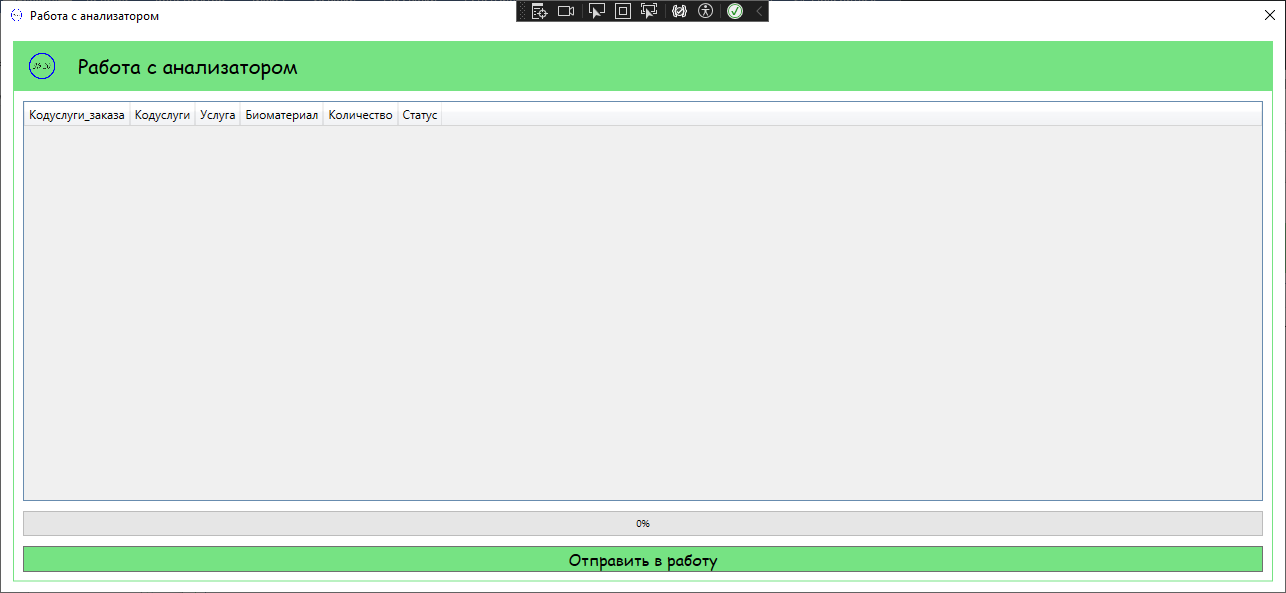


Рисунок 10 - Окно работы с анализатором

В окне работы с анализатором (Рис. 10) можно отправить биоматериал на анализ и наглядно просмотреть на процент завершенности текущего анализа.

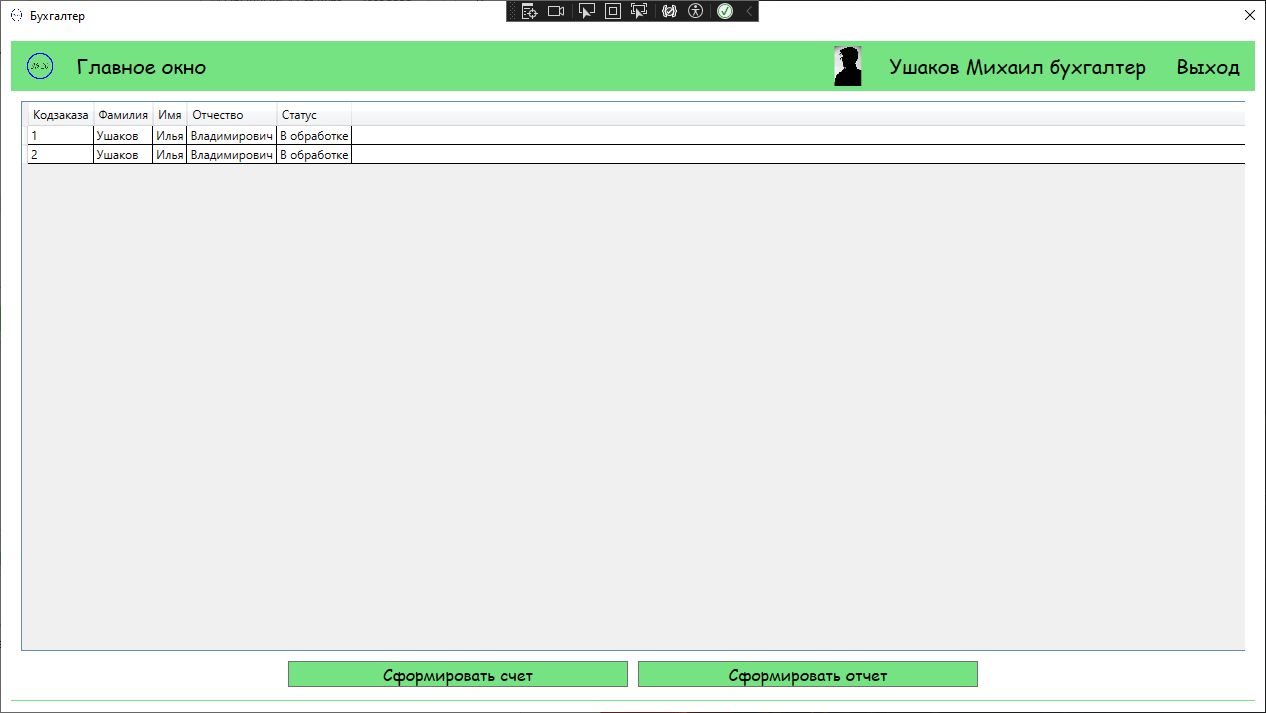


Рисунок 11 - Окно бухгалтера

В своем окне (Рис. 11) бухгалтер может наблюдать заказы, на которые, он может выставлять счета стразовым компаниям за выполненные услуги. А также сформировать отчет.

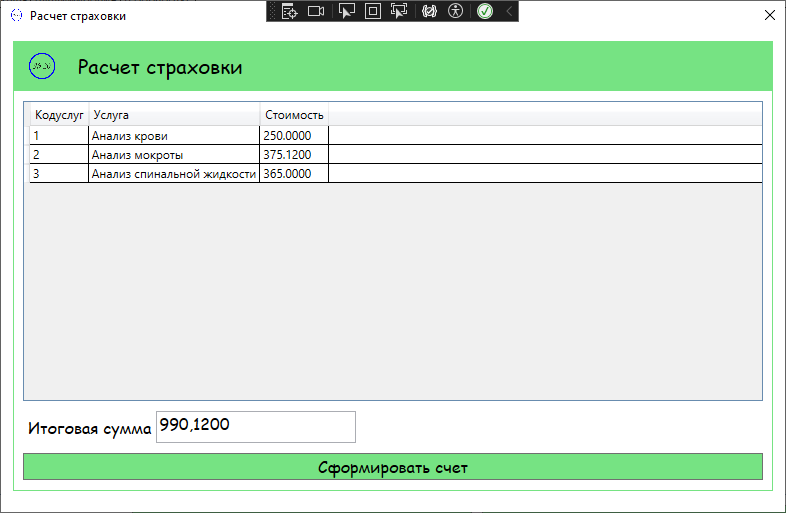


Рисунок 12 - Окно формирования счета компании

Окно расчета страховки для выставления счета страховой компании (Рис. 12) служит для показа оказанных услуг их стоимости и выставления счета страховой компании.

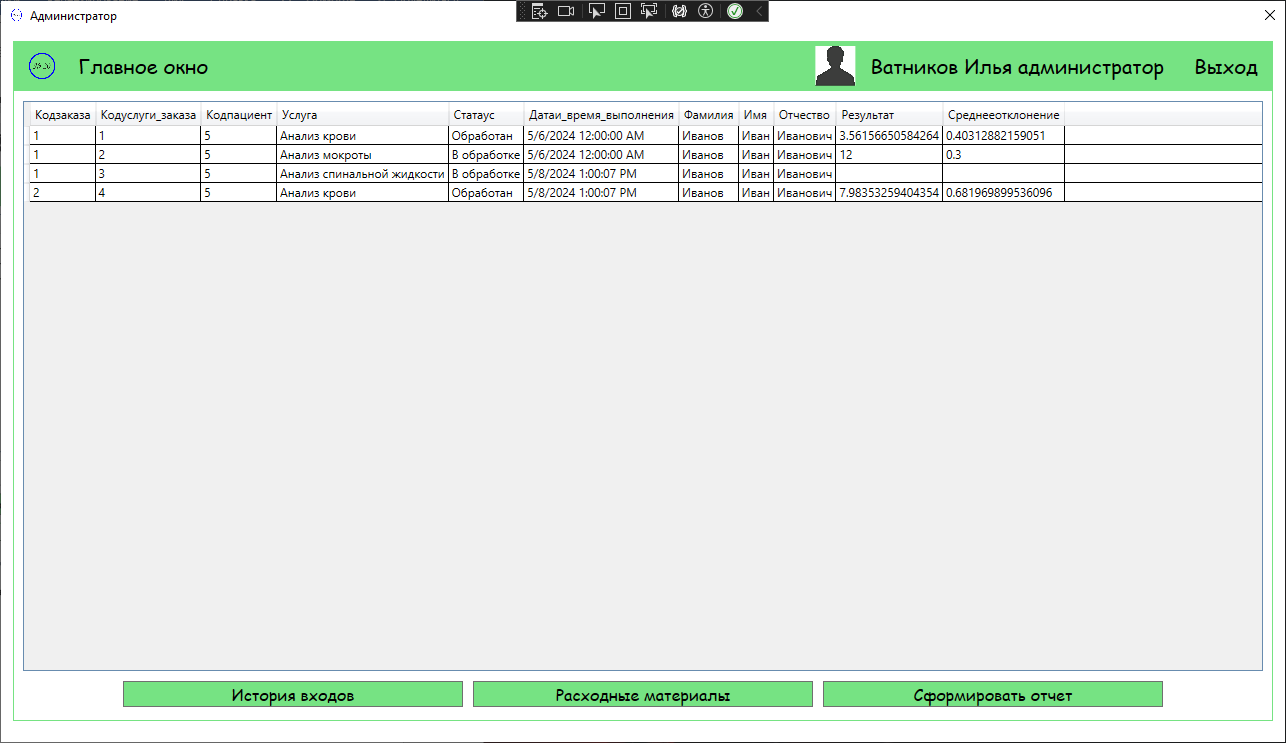


Рисунок 13 - Окно администратора

Окно администратора(Рис. 13) служит для просмотра заказов и оформлять расходные материалы на те или иные услуги в заказе, а также просмотр истории входов.

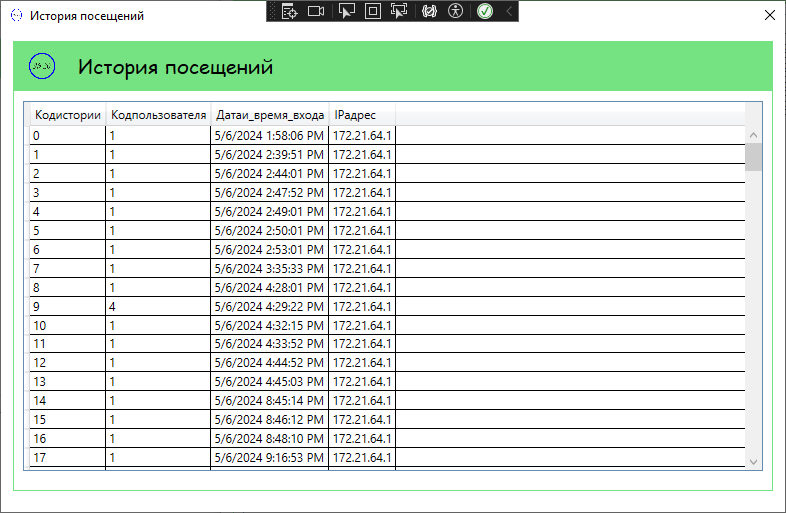


Рисунок 14 - Окно истории посещений

В окне просмотра истории посещений(Рис. 14) администратор может увидеть кто и когда заходил в систему.

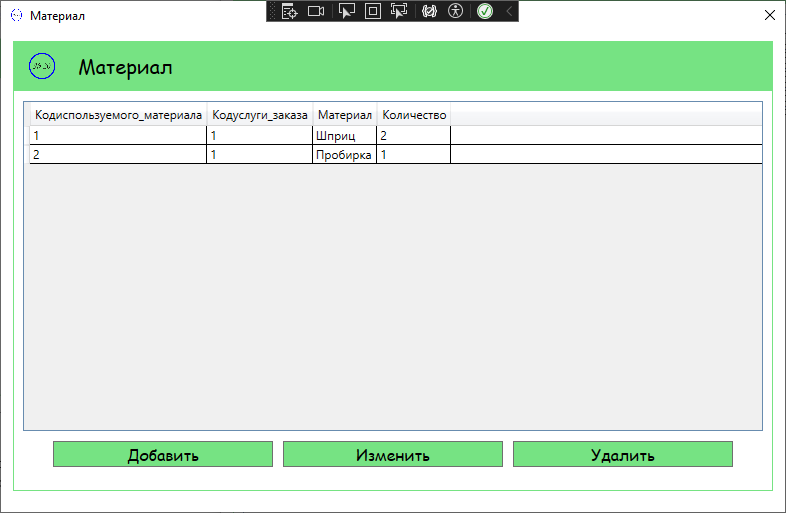


Рисунок 15 - Окно добавления расходных материалов

Окно добавления расходных материалов (Рис. 15) в данном окне можно добавить, изменить или удалить материал необходимый для услуги.

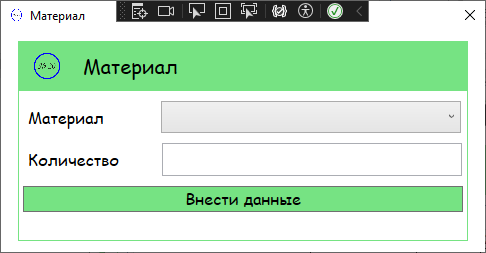


Рисунок 16 - Окно добавления материала

Окно добавления материала (Рис. 16) служит для выбора материала и его количества нужное для выполнения услуги.

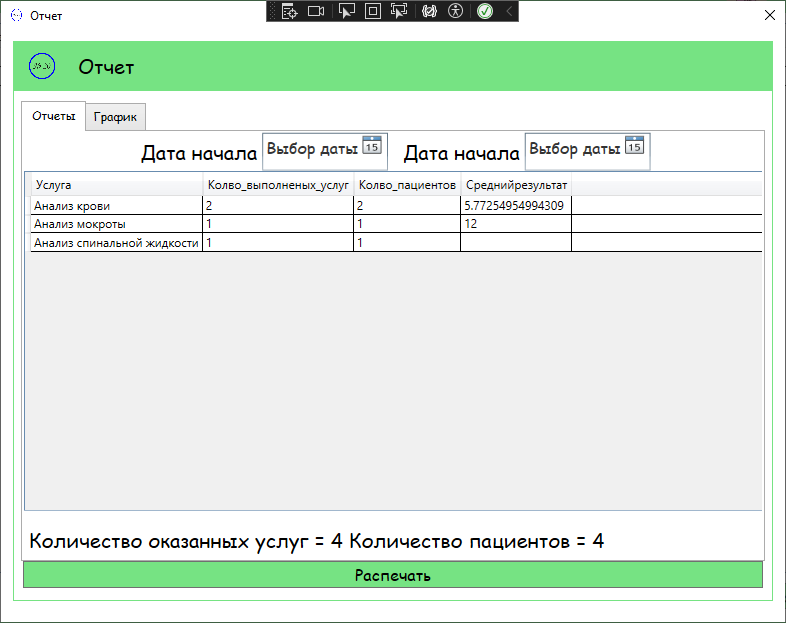


Рисунок 17 - Окно отчетов

Окно отчетов (Рис. 17) предусматривает построение графика (Рис. 18) и печати выводимых данных.

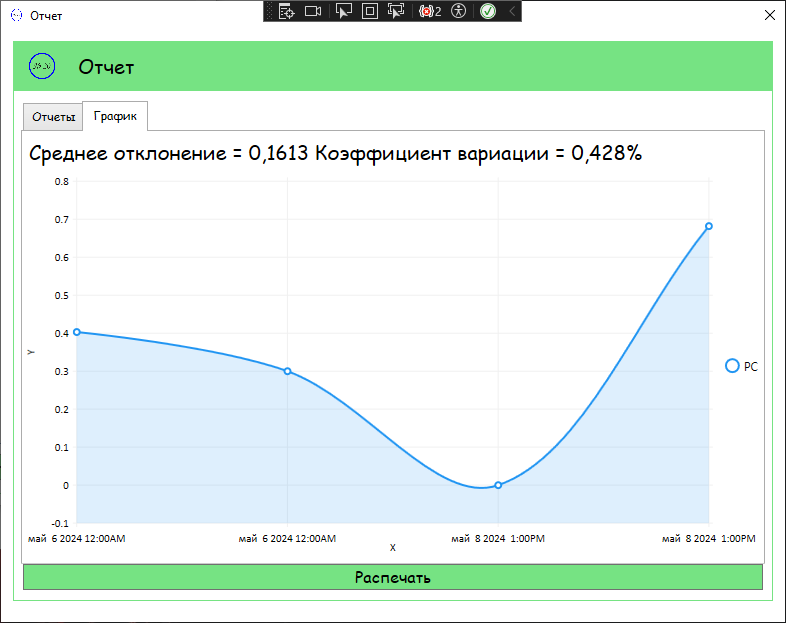


Рисунок 18 - Окно отчетов, просмотр графика

# Работа в системе контроля версий

При прохождение практики я работам с системой контроля версии git. В неё создал репозиторий, в который сохранял промежуточный результат выполнения практики (Рис. 19).

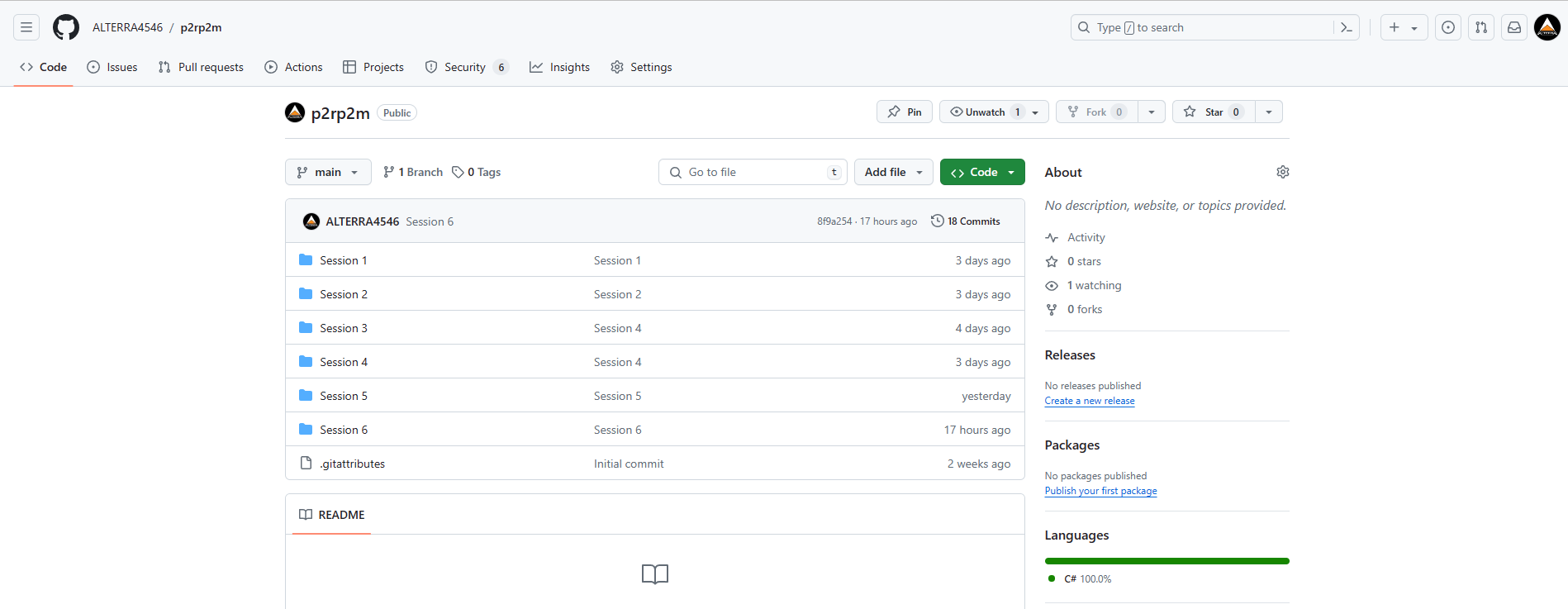


Рисунок 19 - Репозиторий GitHub

# Разработка тестовых наборов и тестовых сценариев

При разработке системы были созданы тестовые наборы для приложения MedLaboratory и телефонного справочника.

При проведение тестирования были проведены следующие тесты: функциональное тестирование, тестирование интерфейса и unit-тесты.

## MedLaboratory

**Функциональное тестирование (Таблица 1)**

Таблица 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Тестируемая функция | Ожидаемый результат | Реальный результат | Вывод |
| 1 | Авторизация | Пользователь входит в свою учетную запись | Пользователь входит в свою учетную запись | + |
| 2 | Показ пароля | Показывает и скрывает пароль | Показывает и скрывает пароль | + |
| 3 | Капча после неудачной попытки авторизации | После неудачной попытки авторизоваться в системе показ капчи | После неудачной попытки авторизоваться в системе показ капчи | + |
| 4 | Пересоздание капчи | При нажатие на кнопку генерации новой капчи, генерируется новая капча | При нажатие на кнопку генерации новой капчи, генерируется новая капча | + |
| 5 | Таймер времени посещения | Поле прошествии 2ч 30мин система блокируется на 30 минут | Поле прошествии 2ч 30мин система блокируется на 30 минут | + |
| 6 | Добавление заказа | При нажатии на кнопку добавить запись открывается окно добавления после заполнения которого данные заносятся в БД | При нажатии на кнопку добавить запись открывается окно добавления после заполнения которого данные заносятся в БД | + |
| 7 | Поиск по ФИО | Вводимые ФИО находиться и выводятся в БД | Вводимые ФИО находиться и выводятся в БД | + |
| 8 | Добавление нового пользователя | Добавление нового пользователя в БД | Добавление нового пользователя в БД | + |
| 9 | Редактирование пользователя | Пользователь был отредактирован и сохранен в БД | Пользователь был отредактирован и сохранен в БД | + |
| 10 | Добавление биоматериала | Введенный биоматериал был добавлен в БД | Введенный биоматериал был добавлен в БД | + |
| 11 | Генерация штрихкода для пробирки | Сгенерирован рабочий штрих код | Сгенерирован рабочий штрих код | + |
| 12 | Формирование отчета по заказам | Формируется отчет по заказам | Формируется отчет по заказам | + |
| 13 | Формирование отчета по услугам | Формируется отчет по услугам | Формируется отчет по услугам | + |
| 14 | Отправка биоматериала на анализ | Биоматериал отправляется на анализ после чего записываться результат анализа в БД | Биоматериал отправляется на анализ после чего записываться результат анализа в БД | + |
| 15 | Формирование счета страховой компании | Сформирован и записан счет страховой компании | Сформирован и записан счет страховой компании | + |
| 16 | Просмотр истории входов | Просмотр данных о выходе пользователей в систему | Просмотр данных о выходе пользователей в систему | + |
| 17 | Добавление расходных материалов | Добавление расходных материалов для оказания услуги в заказе | Добавление расходных материалов для оказания услуги в заказе | + |
| 18 | Вывод данных в PDF файл | Данные сохраняться в PDF файле с правильной кодировкой | Данные сохраняться в PDF файле с правильной кодировкой | + |
| 19 | Отображение графика контроля качества | Данные из БД отображаться на графике | Данные из БД отображаться на графике | + |

**Тестирование интерфейса**

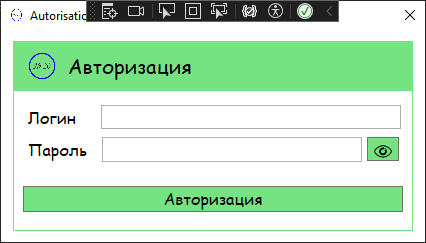
****

Рисунок 20 - Окно авторизации

Окно авторизации (Рис. 20) представлено двумя полями для логина и пароля. Окно поддерживает режим отображения введенного пароля.



Рисунок 21 - Окно прохождения капчи

Окно прохождения капчи (Рис. 21) предусматривает генерацию капчи, текст на которой читаем для человека но, если капча сложная её можно пере генерировать.

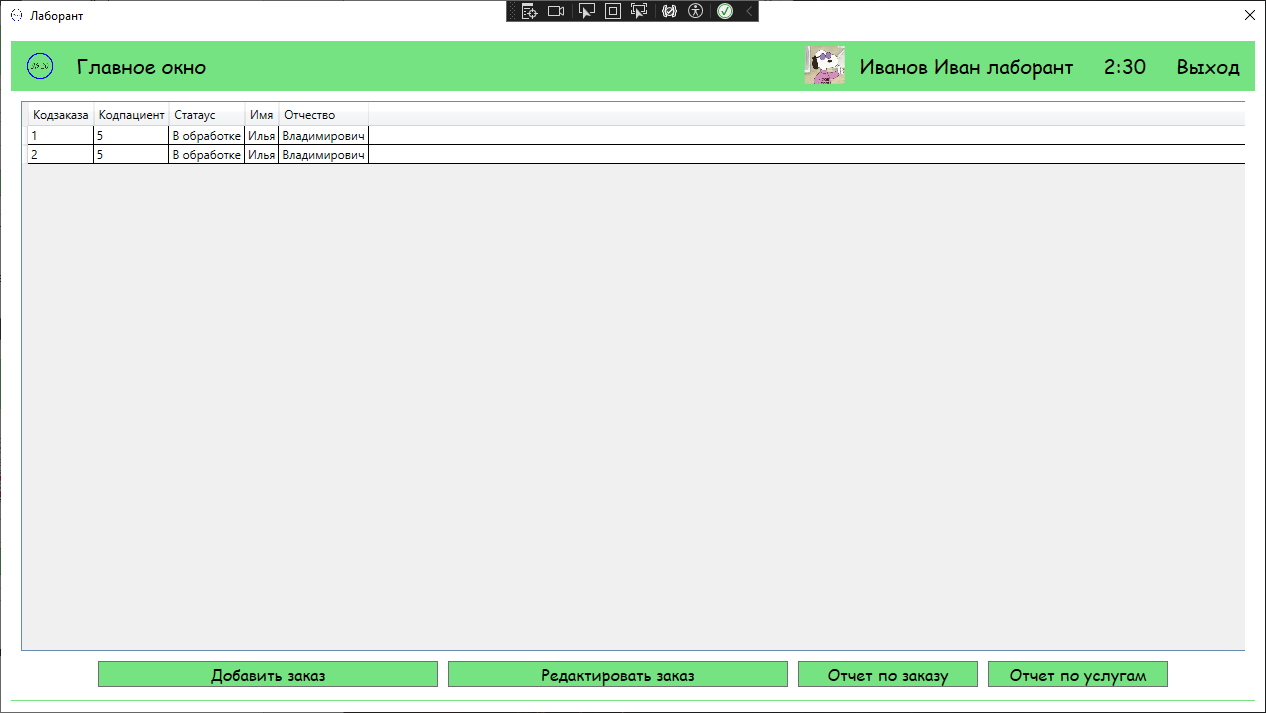


Рисунок 22 - Окно лаборанта

Окно лаборанта служит (Рис. 22) для работы с биоматериалами, окно выполнено интуитивно понятно без большого количества кнопок.

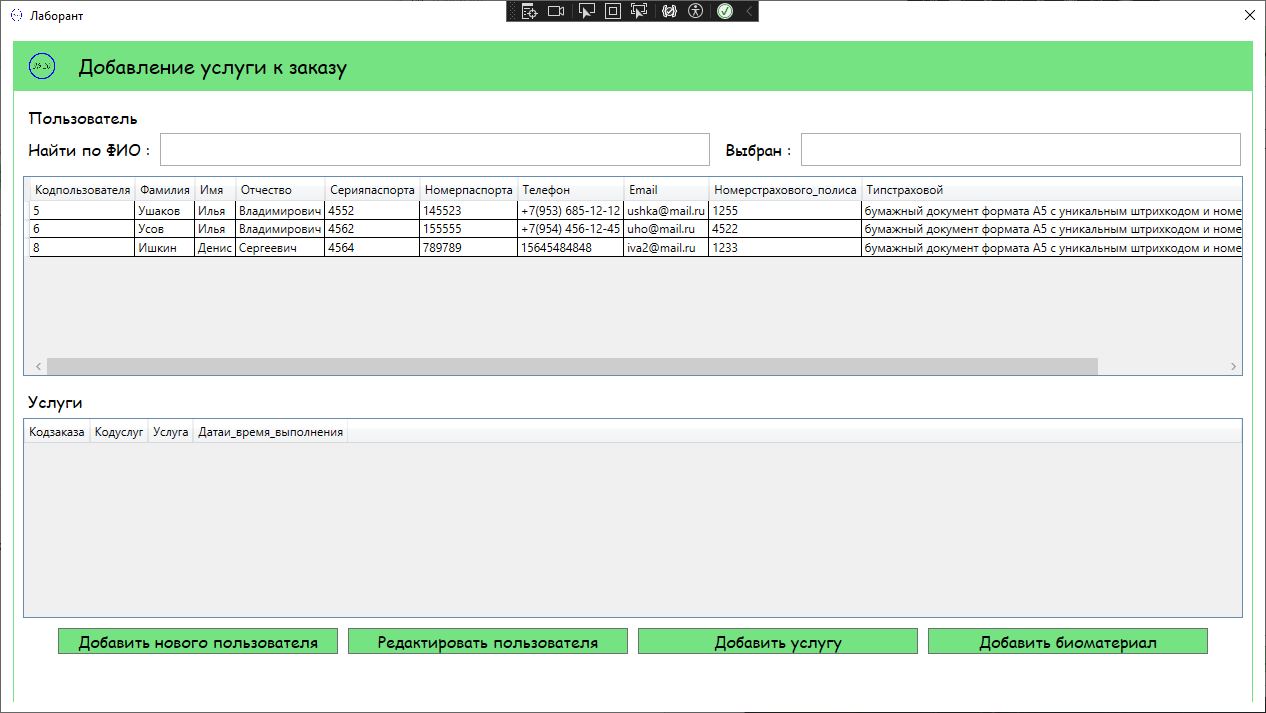


Рисунок 23 - Окно работы с заказа

Окно работы с заказом (Рис. 23) выполненное минималистично для удобного использования, для упрощения работы с заказами предусмотрен поиск по ФИО клиента.

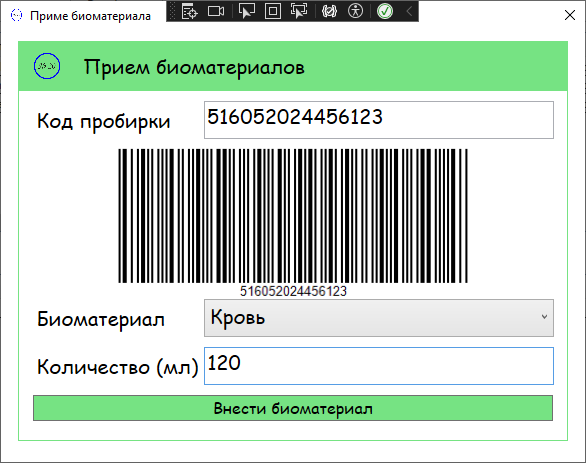


Рисунок 24 - Окно работы с биоматериалом

В окне работы с материалом (Рис. 24) можно указать код пробирки, который сформирует штрих код, ввести тип и количество биоматериала. После сохранения данных автоматически формируется pdf файл со штрих кодом.

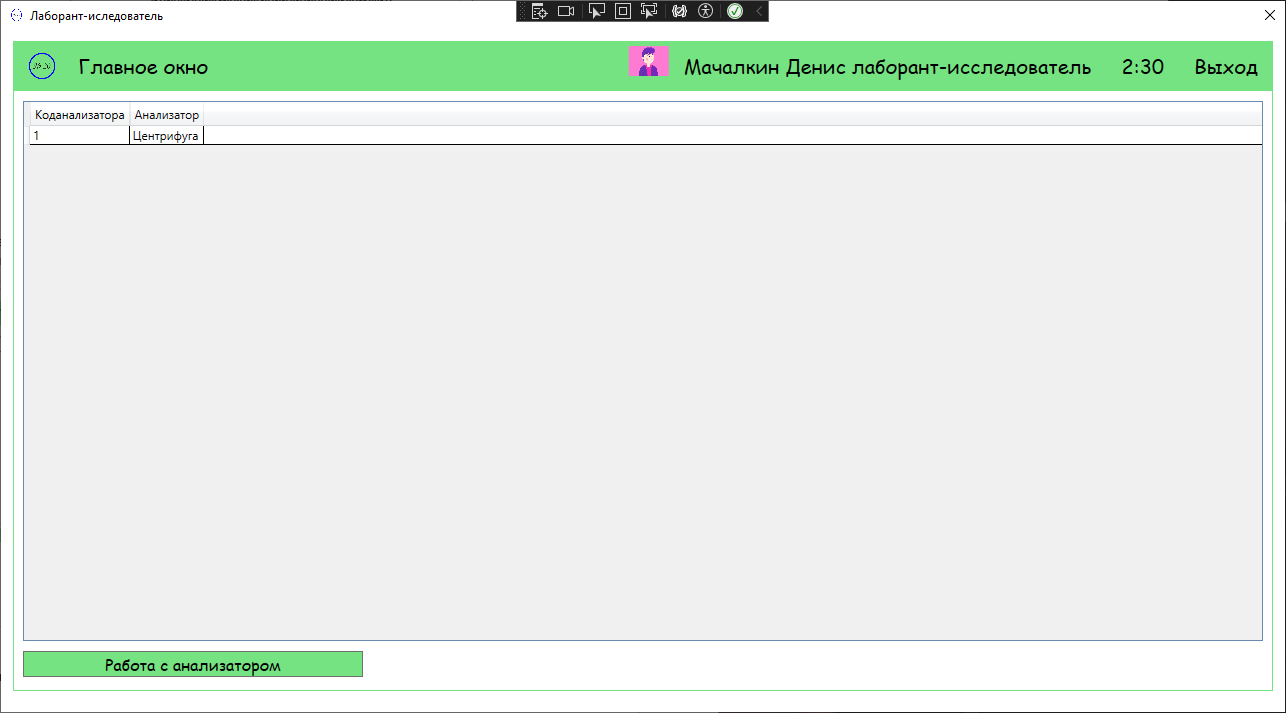


Рисунок 25 - Окно лаборанта-исследователя

Окно лаборанта-исследователя (Рис. 25) похоже по своей стилистике на окно лаборанта.

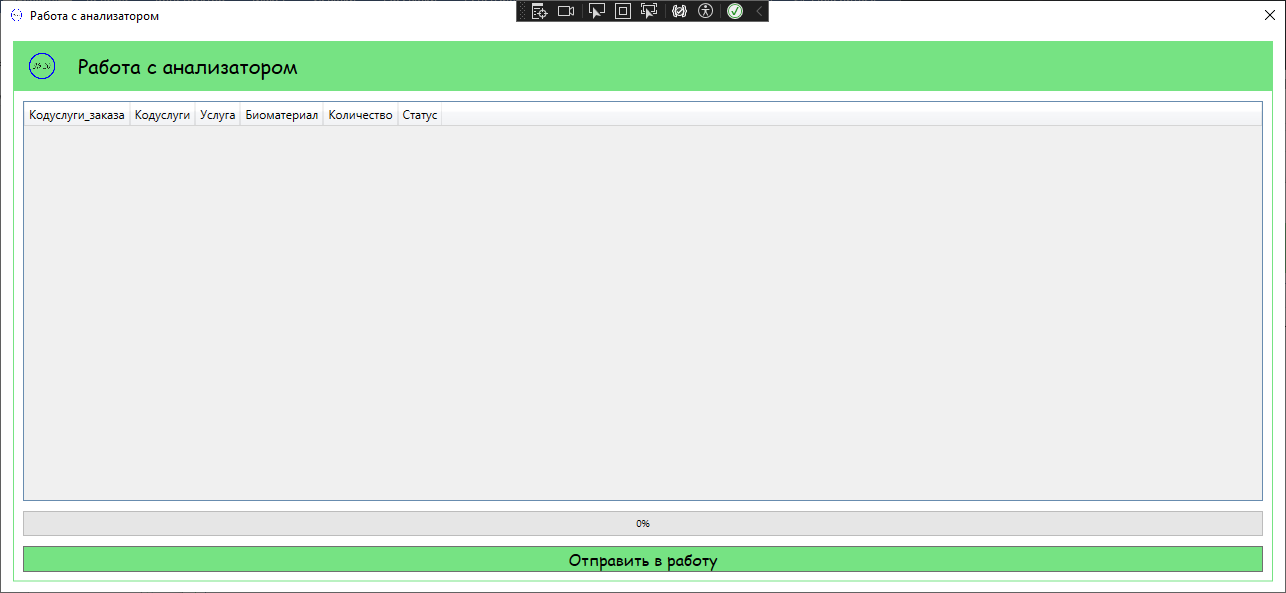


Рисунок 26 - Окно работы с анализатором

В окне работы с анализатором (Рис. 26) можно отправить биоматериал на анализ и наглядно просмотреть на процент завершенности анализа.

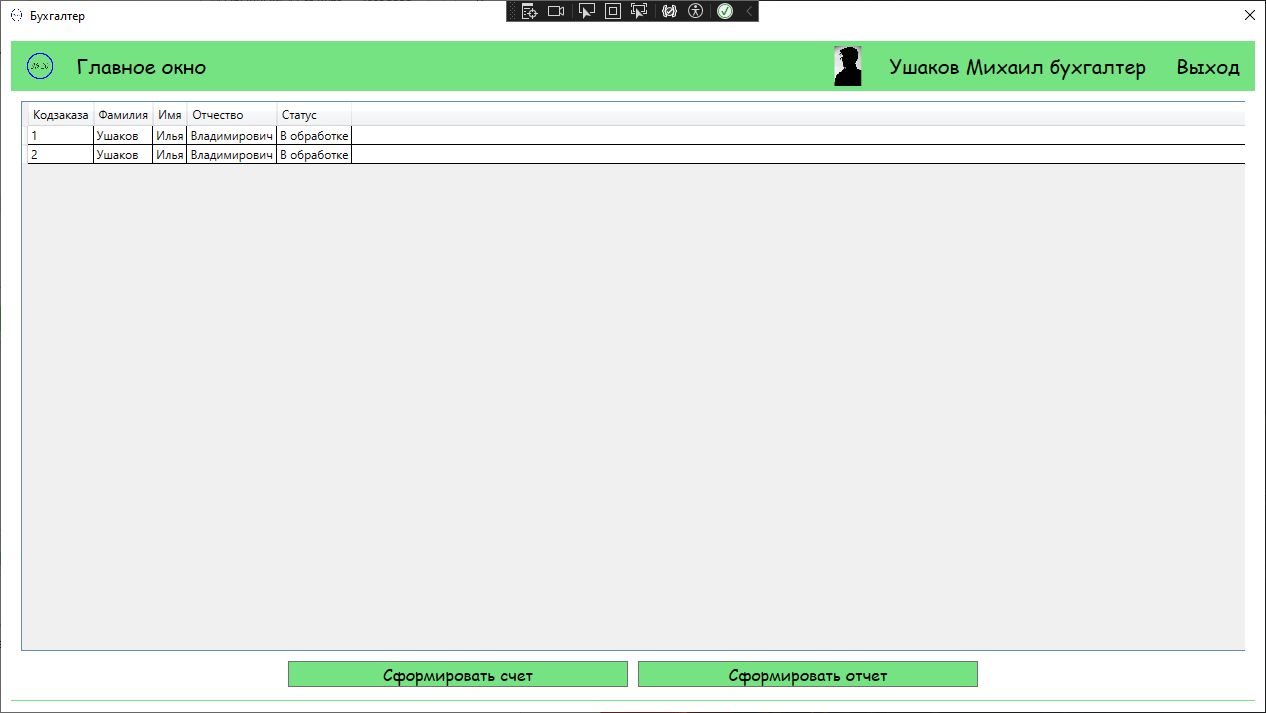


Рисунок 27 - Окно бухгалтера

В своем окне (Рис. 27) бухгалтер может наблюдать заказы, на которые но может выставлять счета стразовым компаниям за выпаленные услуги. Форма выполнена в едином стиле.

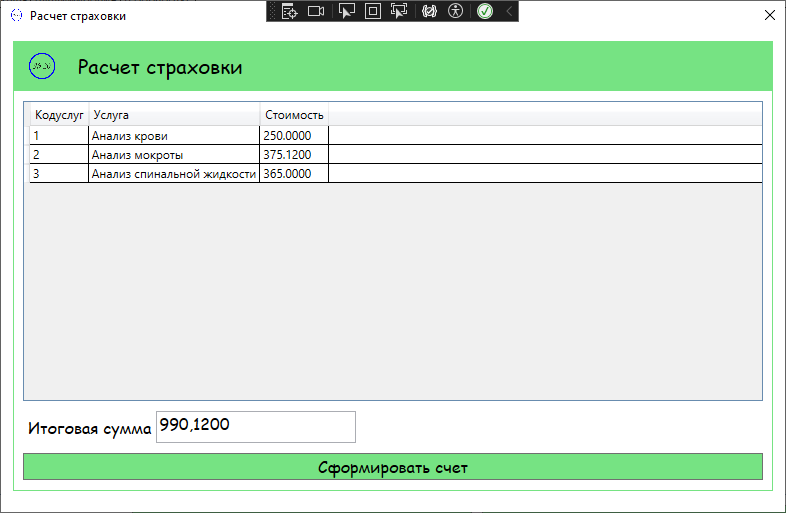


Рисунок 28 - Окно формирования счета компании

Окно расчета страховки для выставления счета страховой компании (Рис. 28) выполнено в едином стиле.

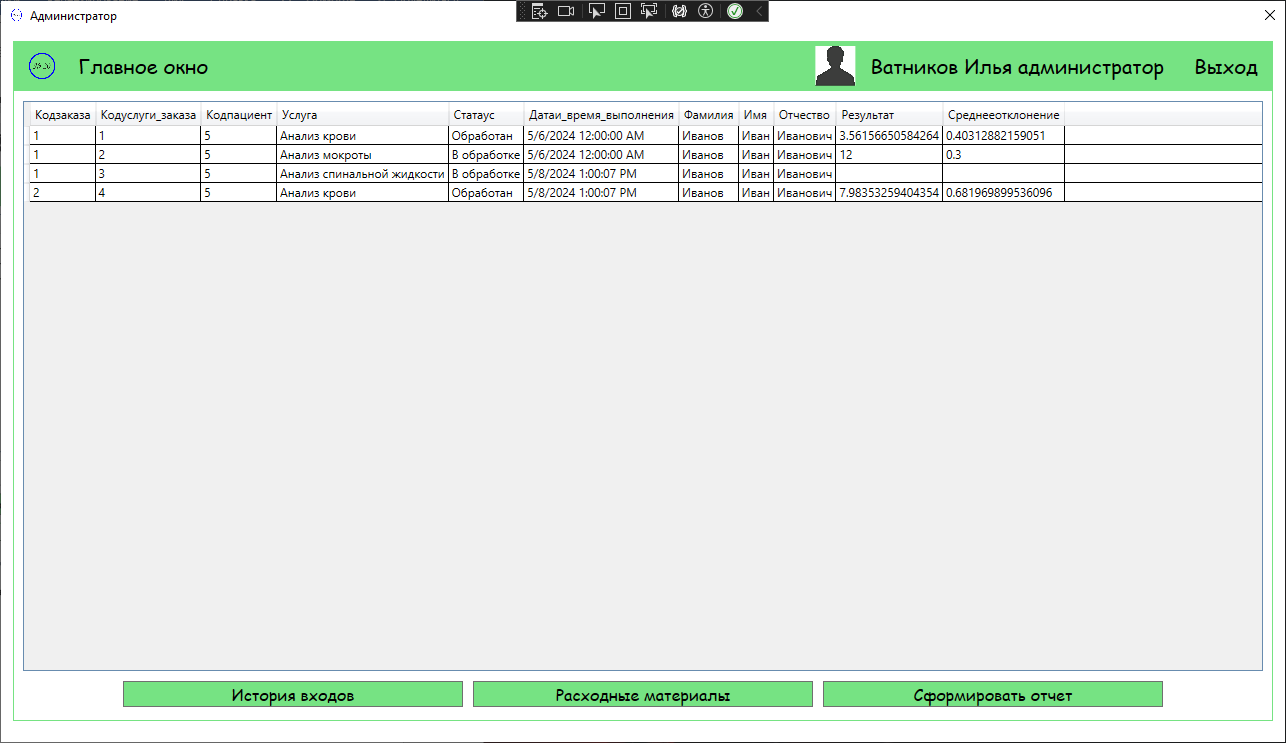


Рисунок 29 - Окно администратора

Окно администратора (Рис. 29) выполнено в единой стилистике, на нем представлены нужный функционал для выполнения работы.

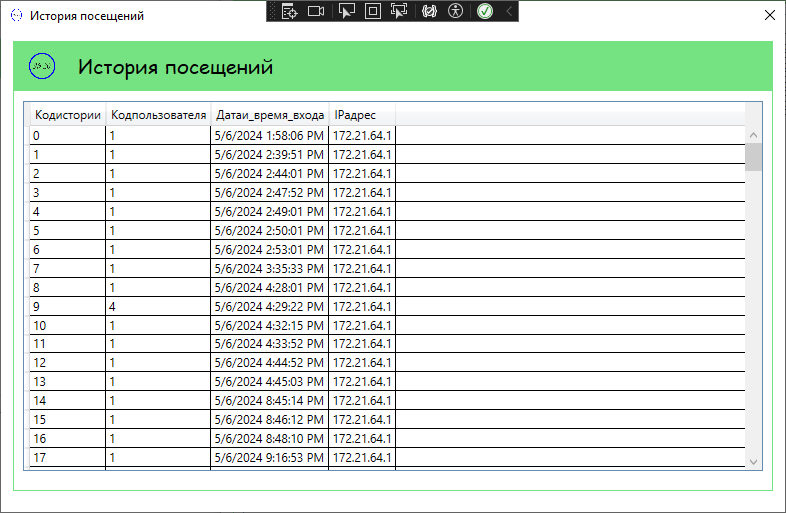


Рисунок 30 - Окно истории посещений

В окне просмотра истории посещений (Рис. 30) администратор может увидеть кто и когда заходил в систему. Форма выполнена также в едином стиле.

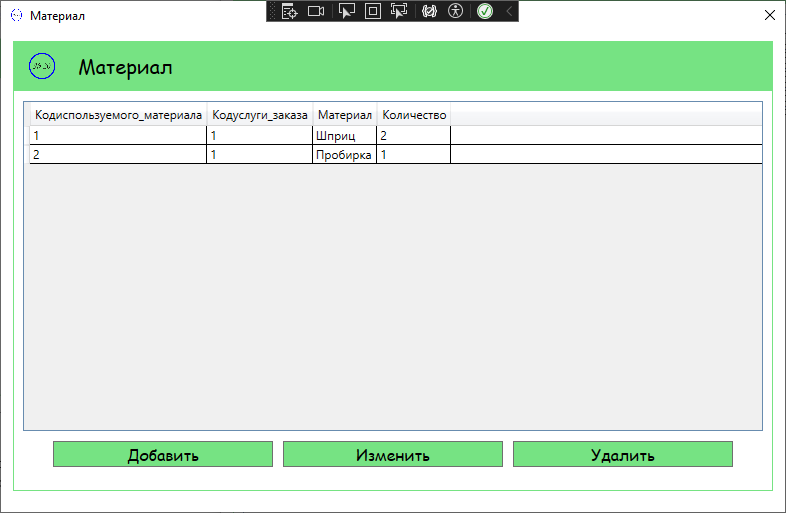


Рисунок 31 - Окно добавления расходных материалов

Окно добавления расходных материалов (Рис. 31) выполнено в едином стиле, и интуитивно понятно при его использовании.

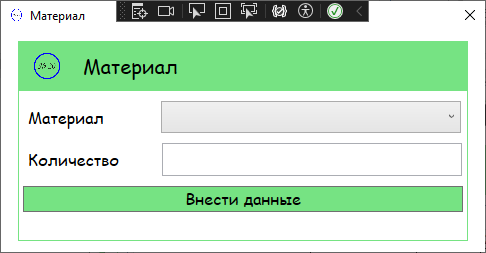


Рисунок 32 - Окно добавления материала

Окно добавления материала (Рис. 32) выполнено в едином стиле, и интуитивно понятно при его использовании. Предусматривает выбор материала и его количество нежное для выполнения услуги.

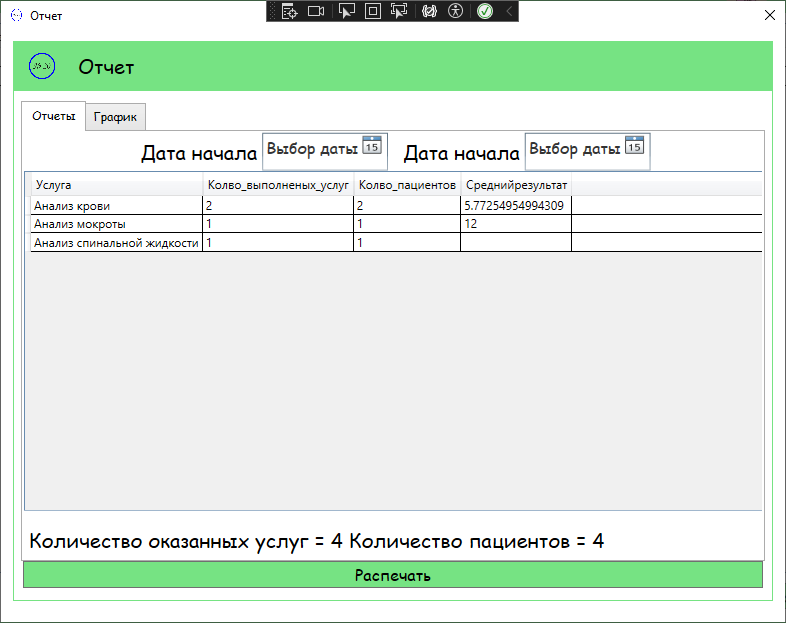


Рисунок 33 - Окно отчетов

Окно отчетов (Рис. 33) интуитивно понятно при использовании оно предусматривает построение графика (Рис. 34) и печати выводимых данных.

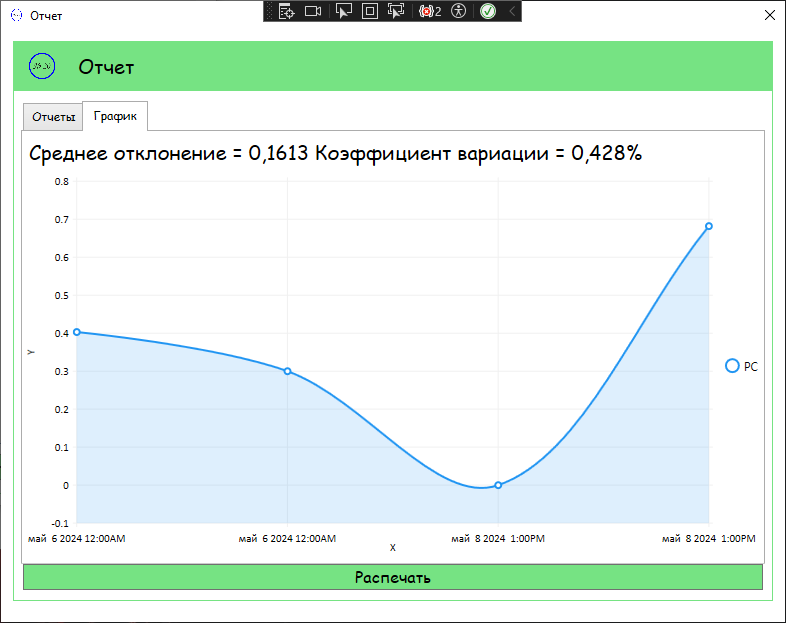


Рисунок 34 - Окно отчетов, просмотр графика

**UnitTest**

Код UnitTest

|  |
| --- |
| namespace UnitTest  {  [TestClass]  public class GenerationCapcha  {  [TestMethod]  public void GenerateTextCapcha()  {  Capcha capcha = new Capcha();  string checkKode = capcha.GenerateRandomText();  Assert.IsTrue(checkKode != null); |

Продолжение код UnitTest

|  |
| --- |
| }  [TestMethod]  public void DrawCaptchaText()  {  Capcha capcha = new Capcha();  var captchaImage = new Bitmap(380, 100);  var graphics = Graphics.FromImage(captchaImage);  capcha.DrawCaptchaText(graphics, captchaImage);  Assert.IsTrue(graphics.TextContrast > 0);  }  [TestMethod]  public void AddCaptchaNoise()  {  Capcha capcha = new Capcha();  var captchaImage = new Bitmap(380, 100);  var graphics = Graphics.FromImage(captchaImage);  capcha.AddNoise(graphics, captchaImage);  Assert.IsTrue(graphics.TextContrast > 0);  }  [TestMethod]  public void BitmapToImageSource()  {  Capcha capcha = new Capcha();  var captchaImage = new Bitmap(380, 100);  var graphics = Graphics.FromImage(captchaImage);  string captchaText = capcha.GenerateRandomText();  capcha.DrawCaptchaText(graphics, captchaImage); |

Продолжение кода UnitTest

|  |
| --- |
| var checkData = capcha.BitmapToImageSource(captchaImage);  Assert.IsTrue(checkData.StreamSource != null);  }  }  [TestClass]  public class TestDataBase  {  [TestMethod]  public void CheckData()  {  string connectionString = "Data Source=DESKTOP-09DGVTM\\SQLEXPRESS;Initial Catalog=MedLaboratory;Integrated Security=True";  using (SqlConnection connection = new SqlConnection(connectionString))  {  connection.Open();  SqlCommand command = new SqlCommand("SELECT COUNT(\*) FROM Пользователи", connection);  int rowCount = (int)command.ExecuteScalar();  Assert.AreEqual(rowCount, 8);  }  }  }  } |

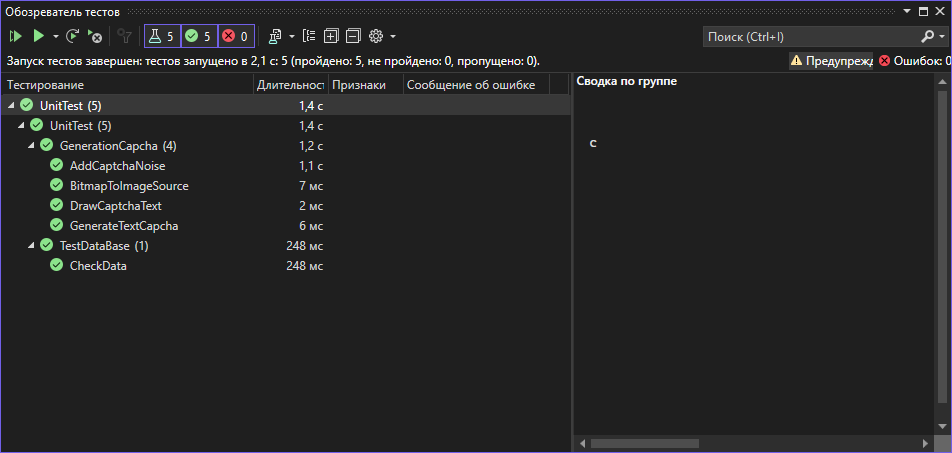


Рисунок 35 - UnitTest

**TestCase**

Таблица 2

|  |  |
| --- | --- |
| TestCase # | MedLaboratory #1 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Просмотр контактов |
| Резюме испытания | Нажатие на кнопку просмотр контактов |
| Шаги тестирования | 1. Запуск приложения |
| Данные тестирования |  |
| Ожидаемый результат | Вывод данных в таблицу |
| Фактический результат | Вывод данных в таблицу |
| Предпосылки | Открытие формы |
| Постусловия |  |
| Статус | Pass |
| Комментарии |  |

Таблица 3

|  |  |
| --- | --- |
| TestCase # | MedLaboratory #2 |
| Приоритет теста | Средний |
| Название тестирования/Имя | Поиск по ФИО |
| Резюме испытания | Ввод в поле текста |
| Шаги тестирования | 1. Заполнение поля поиска |
| Данные тестирования | Иванов |
| Ожидаемый результат | Вывод всех пользователей с Фамилией Иванов |
| Фактический результат | Вывод всех пользователей с Фамилией Иванов |
| Предпосылки |  |
| Постусловия |  |
| Статус | Pass |
| Комментарии |  |

Таблица 4

|  |  |
| --- | --- |
| TestCase # | MedLaboratory #3 |
| Приоритет теста | Выскоий |
| Название тестирования/Имя | Генерация штрих кода |
| Резюме испытания | Ввод в поле кода пробирки |
| Шаги тестирования | 1. Заполнение поля код пробирки |
| Данные тестирования | 050505050550 |
| Ожидаемый результат | Вывод штрих кода и сохранение в pdf |
| Фактический результат | Вывод штрих кода и сохранение в pdf |
| Предпосылки |  |
| Постусловия |  |
| Статус | Pass |
| Комментарии |  |

## Телефонный справочник

**Функциональное тестирование (Таблица 2)**

Таблица 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Тестируемая функция | Ожидаемый результат | Реальный результат | Вывод |
| 1 | Просмотр контактов | Вывод данных контактов | Вывод данных контактов | + |
| 2 | Выделение контактов у которых группа контактов Родственники Зеленым цветом | Выделение контактов у которых группа контактов Родственники  Зеленым цветом | Нет выделения | - |
| 3 | Добавление контакта | Запись добавлена в таблицу Контакты БД | Запись добавлена в таблицу БД | + |
| 4 | Редактирование контакта | Отредактированная запись в таблице Контакты БД | Отредактированная запись в таблице Контакты БД | + |
| 5 | Удаление контакта | Запись в таблице Контакты удалена | Запись в таблице Контакты удалена | + |
| 6 | Сортировка | При вводе поля для сортировки данные в таблице сортируются по введенному полю, результат сортировки выводиться в таблицу | При вводе поля для сортировки данные в таблице сортируются по введенному полю,  результат сортировки выводиться в таблицу | + |
| 7 | Поиск | При вводе ФИО выполняется выводе результат в таблицу | При вводе ФИО выполняется выводе результат в таблицу | + |
| 8 | Экспорт в csv | Вывод данных в csv файл | Не предусмотрена возможность экспорта в csv | - |
| 9 | Фильтрация по полю "группа" | Вывод в таблицу контактов, которые состоят в группе Коллеги | Вывод в таблицу контактов, которые состоят в группе Коллеги | + |
| 10 | Валидация поля "номер телефона" | При внесении телефона в неправильной формате система должна выдать предупреждение | Нет проверки формата телефона | - |
| 11 | Валидация поля "e-mail" | При внесении почти в неверном формате система должна выдать предупреждение | Нет проверки формы почты | - |

**Тестирование интерфейса**

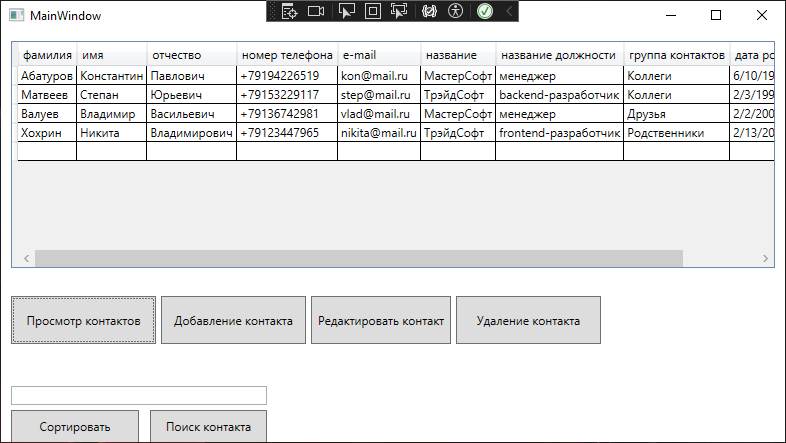


Рисунок 36 - Главное окно

Главное окно (Рис. 36) удобно в работе, за исключением выполнения сортировки, фильтрации и поиска контакта которое объединено в одном поисковом окне из-за чего нельзя одновременно выполнять фильтрацию и поиск.

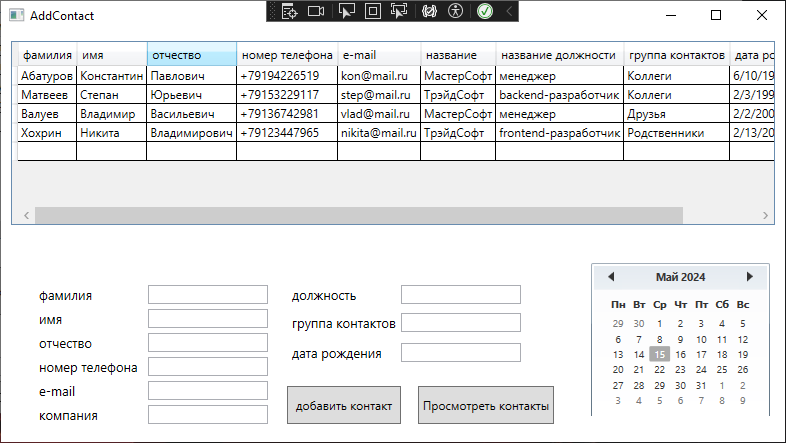


Рисунок 37 - Окно добавления контакта

Окно добавления контакта (Рис. 37) сделано удобно в плане использования.

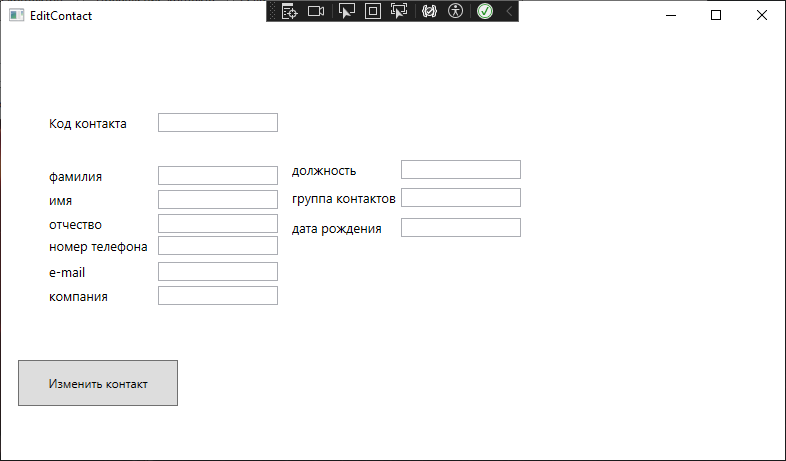


Рисунок 38 - Окно редактирования

Окно редактирования (Рис. 38) не очень удобно так-как данные заранее не подгружаться, также нет таблицы и кода, с помощью которого можно отредактировать запись в таблице.

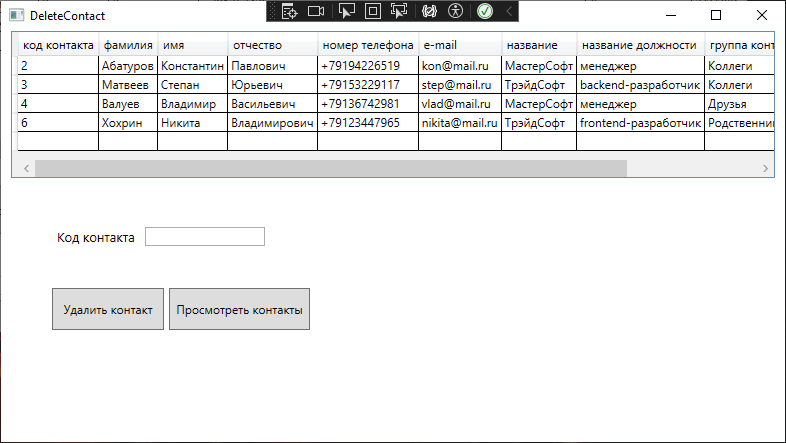


Рисунок 39 - Окно удаления контакта

Окно удаления контакта (Рис. 39) выполнено хорошо и в пользование удобно.

**UnitTest**

Код UnitTest

|  |
| --- |
| namespace UnitTest  {  [TestClass]  public class TestMainWindow  {  [TestMethod]  public void ShowData()  {  MainWindow mw = new MainWindow();  DataTable ds = mw.Show();  Assert.AreEqual(ds.Rows.Count, 6);  }  [TestMethod]  public void SortData()  {  MainWindow mw = new MainWindow();  DataTable ds = mw.Sort("фамилия");  Assert.AreEqual(ds.Rows.Count, 6);  }  [TestMethod]  public void SearchData()  {  MainWindow mw = new MainWindow();  DataTable ds = mw.Search("Никита");  Assert.AreEqual(ds.Rows.Count, 1);  }  } |

Продолжение кода UnitTest

|  |
| --- |
| [TestClass]  public class TestAddContact  {  [TestMethod]  public void ShowData()  {  AddContact ac = new AddContact();  DataTable ds = ac.Show();  Assert.AreEqual(ds.Rows.Count,6);  }  }  [TestClass]  public class TestDeleteContact  {  [TestMethod]  public void ShowData()  {  DeleteContact dc = new DeleteContact();  DataTable ds = dc.Show();  Assert.AreEqual(ds.Rows.Count, 6);  }  [TestMethod]  public void DeleteDataPass()  {  DeleteContact dc = new DeleteContact();  int checkDelete = dc.Delete("12");  Assert.IsTrue(checkDelete > 0);  dc.RemoveDeleteData();  } |

Продолжение кода UintTest

|  |
| --- |
| [TestMethod]  public void DeleteDataFailed()  {  DeleteContact dc = new DeleteContact();  int checkDelete = dc.Delete("50000");  Assert.IsTrue(checkDelete <= 0);  }  }  [TestClass]  public class TestEditContact  {  [TestMethod]  public void EditDataPass()  {  EditContact dc = new EditContact();  int checkEdit = dc.Edit("12", "Иванов", "Иван", "Иванович", "+7(958)456-12-12", "ivan@mail.ru", "1", "1", "Друзья", "12.02.2000");  Assert.IsTrue(checkEdit > 0);  }  [TestMethod]  public void EditDataFailed()  {  EditContact dc = new EditContact();  int checkEdit = dc.Edit("50000", "Иванов", "Иван", "Иванович", "+7(958)456-12-12", "ivan@mail.ru", "1", "1", "Друзья", "12.02.2000");  Assert.IsTrue(checkEdit <= 0);  }  }  } |

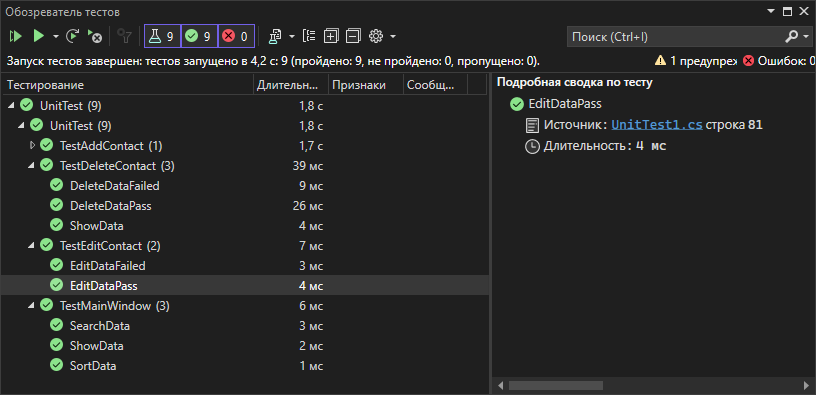
****

Рисунок 40 - Проведение UnitTest

**TestCase**

Таблица 6

|  |  |
| --- | --- |
| TestCase # | PhoneDirectory #1 |
| Приоритет теста | Высокий |
| Название тестирования/Имя | Просмотр контактов |
| Резюме испытания | Нажатие на кнопку просмотр контактов |
| Шаги тестирования | 1. Запуск приложения 2. Нажатие на кнопку “Просмотр контактов” |
| Данные тестирования |  |
| Ожидаемый результат | Вывод данных контактов в таблицу |
| Фактический результат | Вывод данных контактов в таблицу |
| Предпосылки |  |
| Постусловия |  |
| Статус | Pass |
| Комментарии |  |

Таблица 7

|  |  |
| --- | --- |
| TestCase # | PhoneDirectory #2 |
| Приоритет теста | Средний |
| Название тестирования/Имя | Сортировка контактов |
| Резюме испытания | Ввод в поле текста |
| Шаги тестирования | 1. Заполнение поля 2. Нажатие на кнопку “Сортировать” |
| Данные тестирования | Фамилия |
| Ожидаемый результат | Сортировка данных в таблице по выбранному полю |
| Фактический результат | Сортировка данных в таблице по выбранному полю |
| Предпосылки |  |
| Постусловия |  |
| Статус | Pass |
| Комментарии |  |

Таблица 8

|  |  |
| --- | --- |
| TestCase # | PhoneDirectory #3 |
| Приоритет теста | Средний |
| Название тестирования/Имя | Поиск контактов |
| Резюме испытания | Ввод в поле текста |
| Шаги тестирования | 1. Заполнение поля 2. Нажатие на кнопку “Поиск контакта” |
| Данные тестирования | Никита |
| Ожидаемый результат | Поиск данных в таблице и вывод результата |
| Фактический результат | Поиск данных в таблице и вывод результата |
| Предпосылки |  |
| Постусловия |  |
| Статус | Pass |
| Комментарии |  |

# Отладка программного модуля

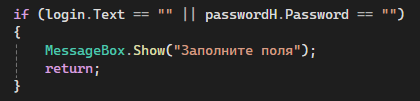


Рисунок 42 - Проверка на заполнение полей логина и пароля

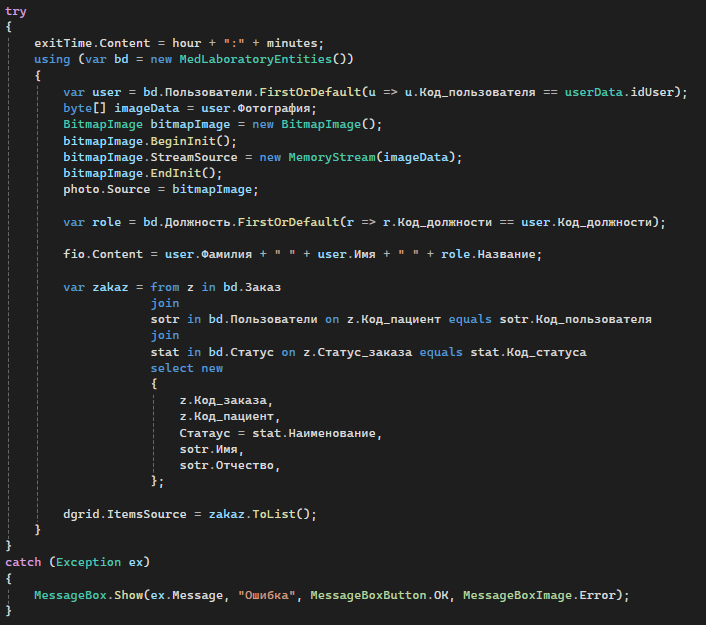


Рисунок 43 - Обработка ошибки связанной с выводом данных из БД

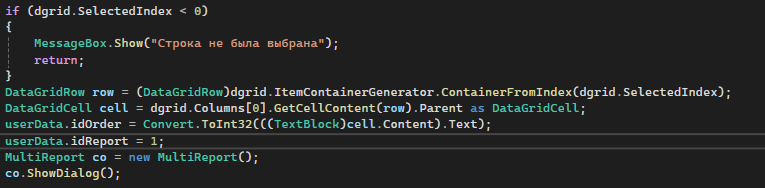


Рисунок 44 - Проверка на выбранную запись в dataGrid

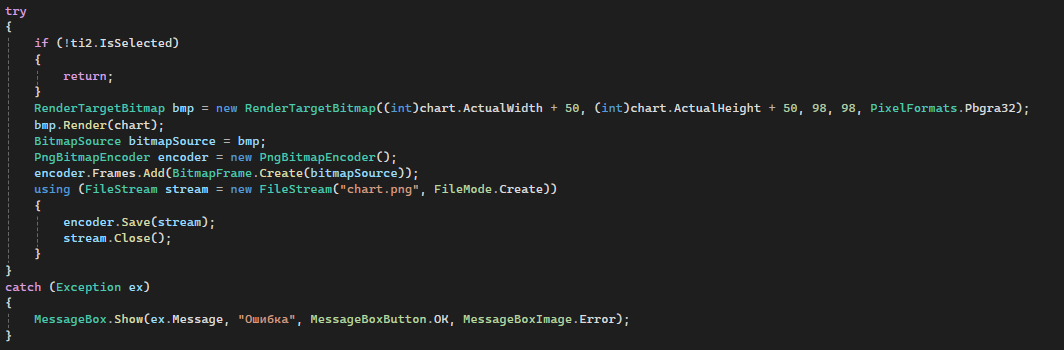


Рисунок 45 - Обработка ошибки связанной с сохранением изображения

# Заключение

Во время прохождения учебной практики я разработал систему MedLaboratory, протестировал и отладил программу для взаимодействия с базой данных. Разработал телефонный справочник для хранения информации о контактах, и разработал приложения для решения ЗЛУ графическим и симплексным методом.

Учебная практика научила меня выполнять правильную интеграцию модулей в готовую систему их тестирование и отладку.

# Приложения

**Приложение 1**

МИНИСТЕРСТВО образования Кировской области

Кировское областное государственное профессиональное образовательное бюджетное учреждение

"Слободской колледж педагогики и социальных отношений"

**Проект**

**Техническое задание**

**Листов 14**

2024

СОДЕРЖАНИЕ

[1. Введение 3](#_Toc107747478)

[2. Основания для разработки 4](#_Toc107747479)

[3. Назначение разработки 5](#_Toc107747480)

[4. Требования к программе или программному изделию 6](#_Toc107747481)

[4.1. Требования к функциональным характеристикам 6](#_Toc107747482)

[4.2. Требования к надежности 6](#_Toc107747483)

[4.2.1. Требования к обеспечению надежного функционирования программы 6](#_Toc107747484)

[4.2.2. Время восстановления после отказа 6](#_Toc107747485)

[4.2.3. Отказы из-за некорректных действий оператора 6](#_Toc107747486)

[4.3. Условия эксплуатации 7](#_Toc107747487)

[4.3.1. Климатические условия эксплуатации 7](#_Toc107747488)

[4.3.2. Требования к квалификации и численности персонала 7](#_Toc107747489)

[4.4. Требования к составу и параметрам технических средств 7](#_Toc107747490)

[4.5. Требования к информационной и программной совместимости 7](#_Toc107747491)

[4.5.1. Требования к информационным структурам и методам решения 7](#_Toc107747492)

[4.5.2. Требования к исходным кодам и языкам программирования 7](#_Toc107747493)

[4.5.3. Требования к программным средствам, используемым программой 8](#_Toc107747494)

[4.5.4. Требования к защите информации и программ 8](#_Toc107747495)

[4.5.5. Специальные требования 8](#_Toc107747496)

[4.6. Требования к маркировке и упаковке 8](#_Toc107747497)

[4.7. Требования к транспортированию и хранению 8](#_Toc107747498)

[4.8. Специальные требования 8](#_Toc107747499)

[5. Требования к программной документации 9](#_Toc107747500)

[6. Технико-экономические показатели 10](#_Toc107747501)

[7. Стадии и этапы разработки 11](#_Toc107747502)

[7.1. Стадии разработки 11](#_Toc107747503)

[7.2. Этапы разработки 11](#_Toc107747504)

[8. Порядок контроля и приемки 13](#_Toc107747505)

[8.1. Виды испытаний 13](#_Toc107747506)

[8.2. Общие требования к приемке работы 13](#_Toc107747507)

[Перечень принятых сокращений 14](#_Toc107747508)

1. Введение

Анализ предметной области

Основной задачей является разработка информационной системы для медицинской лаборатории № 20 города Санкт-Петербурга. Медицинская лаборатория № 20 работает на прием биоматериала из поликлиник города, проводит специализированные исследования с помощью новейшего оборудования, ведет базу данных исследований, предоставляет отчеты по проведенным исследованиям.

В системе будут состоять следующие члены персонала:

• лаборант

• лаборант-исследователь

• бухгалтер

• администратор

Каждый из членов персонала может выполнять определенные действия в системе:

• лаборант может принять биоматериал, получая штрих-код кода пробирки, формировать отчеты

• лаборант-исследователь может работать с анализатором

• бухгалтер может просмотреть отчеты, сформировать счет страховой компании

• администратор может сформировать отчеты, проконтролировать всех

Также для системы должны быть сформированы следующие ограничения:

При входе в учетную запись лаборанта и лаборанта-исследователя должен быть виден таймер (часы:минуты), который фиксирует время сеанса пользователя. Сеанс пользователя не должен превышать 2 ч 30 минут, так как через каждые 2 ч 30 минут необходимо выполнить кварцевание помещений. За 15 минут до окончания времени сеанса должно появиться сообщение об окончании времени сеанса. По окончании времени сеанса реализуйте выход из учетной записи и блокировку входа на 30 минут. Для удобства проверки экспертной группой - укажите время сеанса – 10 минут, появление сообщения – за 5 минут до окончания времени сеанса, блокировка входа – 1 минута.

После первой попытки неуспешной авторизации система выдает сообщение о неуспешной авторизации, а затем помимо ввода логина и пароля просит ввести captcha, состоящую из 4 символов (цифры и буквы латинского алфавита) и графического шума. CAPTCHA - должна содержать минимум 4 символа (буква или цифра), которые выведены не в одной линии. Символы должны быть либо перечеркнуты либо наложены друг на друга. Реализуйте возможность повторной генерации captcha, если пользователю непонятны символы из-за шума. После попытки неудачной авторизации с вводом captcha, система блокирует возможность входа на 10 секунд.

Функции системы:

В окне истории посещений необходимо реализовать просмотр всей истории входа, а также фильтрацию по логину пользователя. Кроме этого, необходимо добавить сортировку по дате попытки входа. Каждая запись истории должна содержать следующие данные: время, логин пользователя, успешная или ошибочная попытка входа.

При организации хранения данных вам необходимо учесть запрет на полное удаление данных, реализовав возможность отправки данных в архив. Кроме того, необходимо учесть, что данные о заказе не могут быть отправлены в архив, если не выполнена хотя-бы одна услуга в заказе.

Лаборант может выполнять поиск клиента по ФИО с помощью метода Левенштейна.

Лаборант-исследователь в окне должен видеть анализаторы лаборатории с возможными исследованиями на них. Выбрав один из представленных анализаторов, пользователь видит список невыполненных услуг, доступных на данном анализаторе. Некоторые услуги могут быть выполнены на нескольких анализаторах. Лаборант-исследователь выбирает услугу и отправляет биоматериал на анализатор с помощью нажатия специальной кнопки “отправить на исследование” около услуги.

Бухгалтер должен сформировать счет на каждую страховую компанию за определенный период времени и сохранить его в формате pdf.

Система должна иметь возможность формировать отчеты и график контроля качества.

В базе данных должны храниться следующие данные:

• услуги лаборатории (наименование, стоимость, код услуги, срок выполнения, среднее отклонение)

• данные пациентов (логин, пароль, ФИО, дата рождения, серия и номер паспорта, телефон, e-mail, номер страхового полиса, тип страхового полиса, страховая компания)

• данные о страховых компаниях (название страховой компании, адрес, ИНН, р/с, БИК)

• заказ (дата создания, которые входят в заказ, услуги, статус заказа, статус услуги в заказе, время выполнения заказа (в днях))

• оказанная услуга (услуга, когда и кем была и на каком анализаторе)

• данные о работе анализатора (дата и время поступления заказа на анализатор, дата и время выполнения (в секундах) услуг на анализаторе)

• данные лаборантов (логин, пароль, ФИО, последняя дата и время входа, набор услуг, которые он может оказывать)

• бухгалтер (логин, пароль, ФИО, последняя дата и время входа, набор услуг, выставленные счета страховым компаниям)

• администратор (логин и пароль)

• история входа в систему

1. Основания для разработки

Основанием для разработки системы является прохождение практики в Слободском колледже педагогики и социальных отношений находящиеся по адресу Слободской, Рождественская ул., 69

1. Назначение разработки

Информационная система разрабатывается для автоматизации и оптимизации процессов ведения медицинской лаборатории, а также для улучшения качества оказания лабораторных услуг персоналом учреждения.

1. Требования к программе или программному изделию
   1. Требования к функциональным характеристикам

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

Возможность отправки данных в архив (история)

Лаборант может принять биоматериал, сформировать отчеты

Лаборант-исследователь может работать с анализатором

Бухгалтер может просмотреть отчеты, сформировать счет страховой компании

Администратор может сформировать отчеты, проконтролировать всех пользователей по истории входа, работать с данными о расходных материалах, используемых в лаборатории

Приложение должно хранить историю входа в систему

* 1. Требования к надежности
     1. Требования к обеспечению надежного функционирования программы

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением Заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

а) организацией бесперебойного питания технических средств;

б) использованием лицензионного программного обеспечения;

в) регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г.

Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;

г) регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов

* + 1. Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 30-ти минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

* + 1. Отказы из-за некорректных действий оператора

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой.

Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

* 1. Условия эксплуатации
     1. Климатические условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации

* + 1. Требования к квалификации и численности персонала

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 2 штатных единиц — системный администратор и конечный пользователь программы — оператор.

Системный администратор должен иметь высшее профильное образование и сертификаты компании-производителя операционной системы. В перечень задач, выполняемых системным администратором, должны входить:

а) задача поддержания работоспособности технических средств;

б) задачи установки (инсталляции) и поддержания работоспособности системных программных средств — операционной системы;

в) задача установки (инсталляции) программы.

г) задача создания резервных копий базы данных.

* 1. Требования к составу и параметрам технических средств

В состав технических средств должен входить IВМ-совместимый персональный компьютер (ПЭВМ), выполняющий роль сервера, включающий в себя:

* процессор Pentium-2.0Hz, не менее;
* оперативную память объемом, 1 Гигабайт, не менее;
* операционную систему Windows 10;
  1. Требования к информационной и программной совместимости
     1. Требования к информационным структурам и методам решения

База данных работает под управлением Microsoft SQL Server. Используется много поточный доступ к базе данных. Необходимо обеспечить одновременную работу с программой с той же базой, данной модулей экспорта внешних данных.

* + 1. Требования к исходным кодам и языкам программирования

Дополнительные требования не предъявляются

* + 1. Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы Windows 10

* + 1. Требования к защите информации и программ

Требования к защите информации и программ не предъявляются

* + 1. Специальные требования

Специальные требования к данной программе не предъявляются

* 1. Требования к маркировке и упаковке
  2. Требования к транспортированию и хранению
  3. Специальные требования

1. Требования к программной документации

Руководство оператора

1. Технико-экономические показатели

Ориентировочная экономическая эффективность не рассчитываются. Аналогия не проводится ввиду уникальности предъявляемых требований к разработке.

1. Стадии и этапы разработки
   1. Стадии разработки

Разработка должна быть проведена в три стадии:

1. разработка технического задания;

2. рабочее проектирование;

3. внедрение.

* 1. Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

1. разработка программы;

2. разработка программной документации;

3. испытания программы.

На стадии внедрения должен быть выполнен этап разработки подготовка и передача программы

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1. постановка задачи;

2. определение и уточнение требований к техническим средствам;

3. определение требований к программе;

4. определение стадий, этапов и сроков разработки программы и документации на неё;

5. согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями к составу документации.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

1. разработка, согласование и утверждение, и методики испытаний;

2. проведение приемо-сдаточных испытаний;

3. корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах Заказчика.

1. Порядок контроля и приемки
   1. Виды испытаний

Приемо-сдаточные испытания должны проводиться на объекте Заказчика в оговоренные сроки.

Приемо-сдаточные испытания программы должны проводиться согласно разработанной Исполнителем и согласованной Заказчиком Программы и методик испытаний.

Ход проведения приемо-сдаточных испытаний Заказчик и Исполнитель документируют в Протоколе проведения испытаний

* 1. Общие требования к приемке работы

На основании Протокола проведения испытаний Исполнитель совместно с Заказчиком подписывает Акт приемки-сдачи программы в эксплуатацию.

Перечень принятых сокращений

|  |  |
| --- | --- |
| JSON | текстовый формат обмена данными, основанный на JavaScript. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |