|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Министерство образования и науки Пермского края  государственное бюджетное профессиональное образовательное учреждение  «Пермский химико-технологический техникум»   |  |  | | --- | --- | |  | Специальность 09.02.07 «Информационные системы и программирование» |   **Отчет по учебной практике УП.01**  **ПМ.01 «**Разработка модулей программного обеспечения для компьютерных систем»    Сроки прохождения практики 8 апреля – 13 апреля 2024 года (рассредоточено)  13 мая – 18 мая 2024 года (рассредоточено)  Место прохождения практики ГБПОУ «ПХТТ»   |  |  | | --- | --- | |  | Выполнил студент гр. ИС-21-3/ \_\_\_\_\_\_ / А.А. Каменских/  группа подпись… расшифровка подписи  Руководитель практики  ГБПОУ «ПХТТ» \_\_\_\_\_\_\_\_\_\_\_\_ / П.Н. Токарева /  подпись расшифровка подписи |   2022 |

**СОДЕРЖАНИЕ ОТЧЕТА ПО УЧЕБНОЙ ПРАКТИКЕ**

|  |  |
| --- | --- |
| Характеристика | 3 |
| Программа практики | 5 |
| 1. Отчет о выполненных работах | 6 |
| * 1. Формулировка задания №1 и результат выполнения | 6 |
| * 1. Формулировка задания №2 и результат выполнения | 15 |
| * 1. Формулировка задания №3 и результат выполнения | 19 |
| * 1. Формулировка задания №4 и результат выполнения | 24 |
| Список литературы |  |
| Дневник прохождения практики |  |

# **ХАРАКТЕРИСТИКА**

Обучающийся (аяся) Каменских Анна Алексеевна Группы ИС-21-3

специальности СПО 09.02.07 «Информационные системы и программирование» проходил(а) учебную практику по профессиональному модулю **ПМ.01 «**Разработка модулей программного обеспечения для компьютерных систем» в период с «8» апреля 2024 г. по «13» апреля 2024 г (рассредоточено) и в период с «13» мая 2024 г. по «18» мая 2024 г (рассредоточено) на предприятии / в организации ГБПОУ «ПХТТ»

За время работы проявил(а) себя как ответственный/безответственный, исполнительный/неисполнительный, коммуникабельный/замкнутый, доброжелательный/наглый сотрудник.

В ходе выполнения всех видов работ обучающийся(аяся) показал(ла) сформированность следующих компетенций:

|  |  |
| --- | --- |
| **Наименование компетенции** | **Уровень (оптимальный/средний/ допустимый уровень)** |
| OK 0l. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. | оптимальный/средний/  допустимый / |
| ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности. | оптимальный/средний/  допустимый / |
| ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях. | оптимальный/средний/  допустимый / |
| ОК 04. Эффективно взаимодействовать и работать в коллективе и команде. | оптимальный/средний/  допустимый / |
| ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста. | оптимальный/средний/  допустимый / |
| ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения. | оптимальный/средний/  допустимый / |
| ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях. | оптимальный/средний/  допустимый / |
| ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности. | оптимальный/средний/  допустимый / |
| ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках. | оптимальный/средний/  допустимый / |
| ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. | оптимальный/средний/  допустимый / |
| ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием. | оптимальный/средний/  допустимый / |
| ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств. | оптимальный/средний/  допустимый / |
| ПК 1.4. Выполнять тестирование программных модулей. | оптимальный/средний/  допустимый |
| ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода. | оптимальный/средний/  допустимый |

К работе относился\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Цели и задачи практики достигнуты / достигнуты не в полном объёме.

Руководитель практики от ГБПОУ «ПХТТ»

преподаватель, Токарева Полина Николаевна

|  |  |  |
| --- | --- | --- |
| Дата 18.05.2024 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | / П.Н. Токарева/  Расшифровка подписи |

# **ПРОГРАММА УЧЕБНОЙ ПРАКТИКИ**

**по модулю ПМ.01 «**Разработка модулей программного обеспечения для компьютерных систем»

**по специальности 09.02.07 «Информационные системы и программирование»**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование вида работ** | **Количество часов** |
|  | Инструктаж по технике безопасности и пожарной безопасности.  Разработка приложения с разграничением прав доступа: механизм авторизации, реализация функционала по работе с БД (добавление, изменение, удаление данных) | 22 |
|  | Разработка приложения в соответствии с функциональным назначением программы, работа по созданию интерфейса программы, создание модуля обработки данных, создание модуля формирования штрихкода. Формирование документов по шаблону с возможностью сохранения в разных форматах (.docx /.xlsx/.jpg/.pdf) | 14 |
|  | Разработка дополнительного функционала в соответствии с техническим заданием. Создание модуля по работе с API по HTTP, работа с JSON-файлом | 12 |
|  | Разработка алгоритмов программных модулей в соответствии с техническим заданием | 4 |
|  | Разработка технической документации по программному продукту | 12 |
|  | Тестирование и отладка программных модулей. | 4 |
|  | Оформление отчетной документации. Презентация проекта. | 4 |

Руководитель практики от ГБПОУ «ПХТТ»

преподаватель, Токарева Полина Николаевна

|  |  |  |
| --- | --- | --- |
| Дата 18.05.2024 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | /П.Н. Токарева/  Расшифровка подписи |

# **ОТЧЕТ О ВЫПОЛНЕННЫХ РАБОТАХ**

**Задание 1**

**База данных и импорт**

Создайте базу данных, используя предпочтительную платформу (MySQL / Microsoft SQL Server), на сервере баз данных, который вам предоставлен.

Создайте таблицы основных сущностей, атрибуты, отношения и необходимые ограничения. После создания базы данных требуется импортировать предоставленные данные из папки «Импорт». Возможно, вам понадобится отформатировать данные, прежде чем загрузить их в таблицы, которые вы только что создали. В любом случае созданные таблицы должны содержать начальные тестовые данные.

Порядок работы лаборатории: на каждую единицу принятого биоматериала создается заказ, который может содержать в себе услуги (одну или несколько) – исследования биоматериала. У одного пациента может быть несколько заказов. Хранение данных о всех пациентах и заказах позволит формировать все необходимые отчеты, отслеживать динамику показателей и состояние здоровья пациента, а также автоматизировать работу сотрудников лаборатории.

Обеспечьте хранение в базе данных:

• 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)

Разработанная вами база данных должна быть в 3 НФ.

*К разработанной баз данных создайте словарь данных (пример словаря данных в папке с ресурсами).*

**Окно входа**

При запуске приложения окно входа – первое, что видит пользователь. На ней пользователю предлагается ввести свой логин и пароль. Только после удачной авторизации пользователь получает доступ к остальным модулям системы.

При вводе пароль должен быть скрыт маской ввода, но также должна быть реализована возможность просмотра введенного пароля.

При входе система выводит фото пользователя, фамилию и имя пользователя, его роль.

После авторизации пользователь получает доступ к нужному функционалу:

• 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.

Реализуйте окна для будущей разработки интерфейсов пользователей системы. После входа в любую учетную запись должна быть реализована возможность выхода на главный экран – окно входа.

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.

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.

**Проверка данных**

Для проверки ввода данных входа необходимо создать библиотеку для проверки электронной почты, сложности пароля и верности логина. Название библиотеки Mail\_LIB.dll. Создайте три метода:

1. *bool check\_mail(string mail)* должен проверить конструкцию почты.

2. *bool check\_password (string password)* должен проверить сложность пароля: пароль не должен быть простым, содержать хотя бы одну букву, хотя бы один символ,хотя бы одну цифру, иметь длину минимум 8 символов.

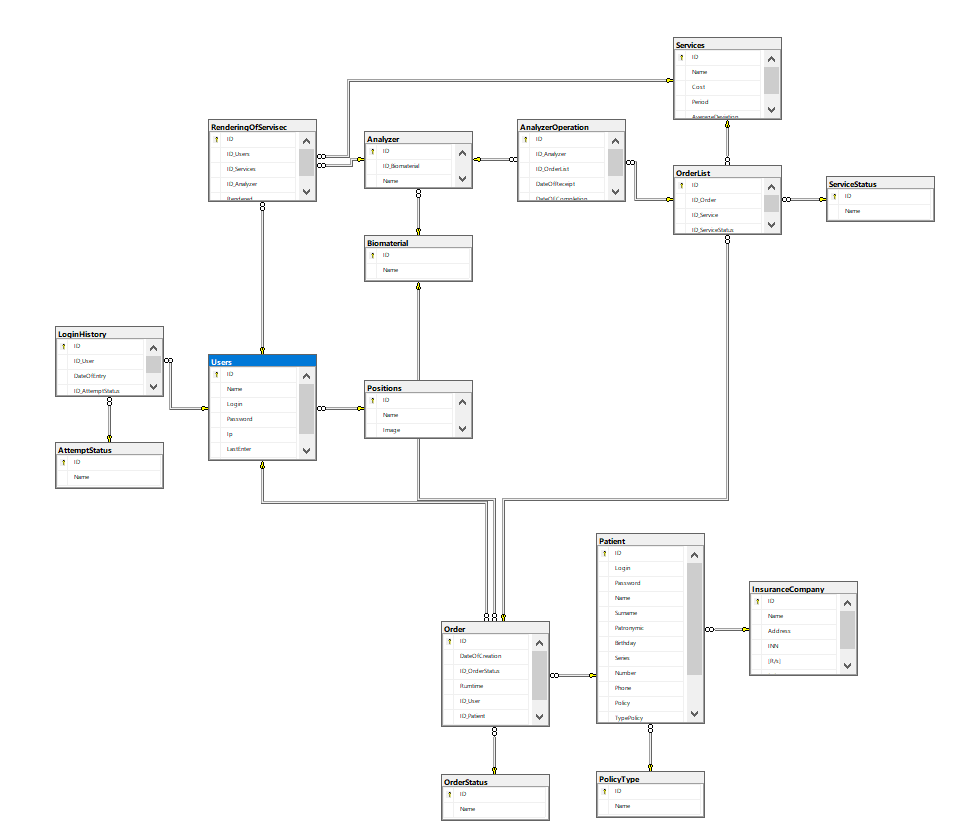
3*. bool check\_login(string login)* должен проверять корректность введённого логина. Логин может содержать только латинские буквы и цифры, иметь длину минимум 6 символов.

Библиотеку необходимо использовать при вводе любых электронных почт, паролей и логинов.

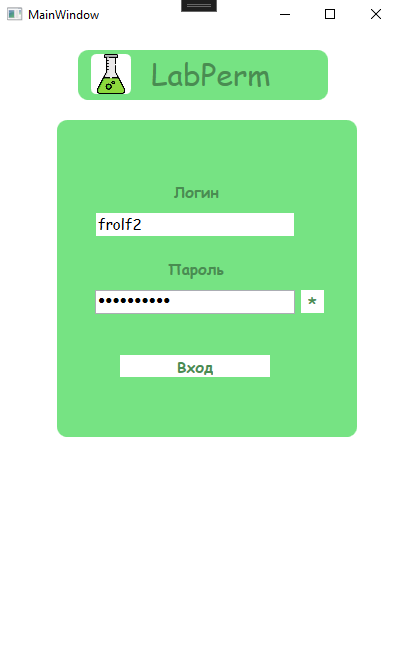
**История входа**

Приложение должно хранить историю входа в систему, так как в системе будут храниться медицинские данные пациентов. Окно для просмотра истории должно быть доступно администратору системы. В этом окне 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.

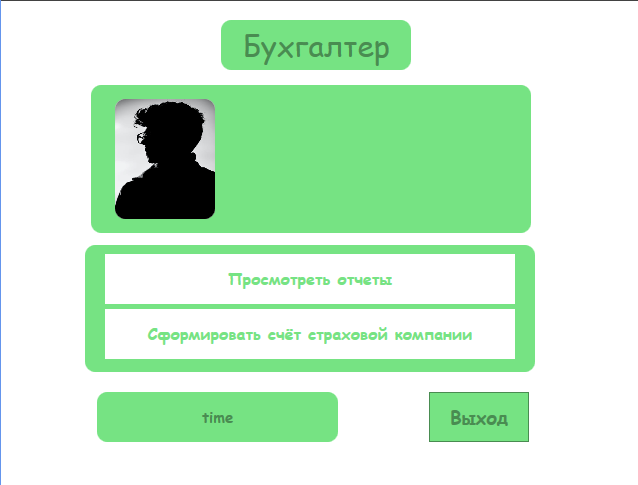
**Физическая модель данных (диаграмма из SQL):**



**Скриншот форм/ы авторизации с заполненными данными:**



**Скриншот форм/ы Бухгалтера с заполненными данными:**



**Скриншот форм/ы администратора с заполненными данными:**



**Программный код формирования Капчи:**

//КАПЧА

private string GenerateRandomText(int length)

{

const string chars = "ABSDEFGHIJKLMNOPQRSTUVWXYZabcdefghjklmnopqrstuvwxyz0123456789";

Random random = new Random();

StringBuilder captchaText = new StringBuilder();

for (int i = 0; i < length; i++)

{

captchaText.Append(chars[random.Next(chars.Length)]);

}

return captchaText.ToString();

}

private const string CaptchaText = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";

private readonly Random \_random = new Random();

private Bitmap GenerateCaptchaImage()

{

string capcthaText = GenerateRandomText(6);

Font font = new Font("Comic Sans MS", 20);

Bitmap bitmap = new Bitmap(200, 50);

C.Content = capcthaText;

using (Graphics g = Graphics.FromImage(bitmap))

{

g.Clear(System.Drawing.Color.White);

int letterWidth = bitmap.Width / capcthaText.Length;

int initialX = 0;// Начальное значение x

Random random = new Random();

foreach (char letter in capcthaText)

{

int y = random.Next(0, 10); // Фиксированный y

int offsetX = random.Next(-3, 6);

int offsetY = random.Next(-3, 6);

// Получаем значения компонентов цвета из шестнадцатеричного представления

byte red = Convert.ToByte(0x49); // 0x49 соответствует 73

byte green = Convert.ToByte(0x8C); // 0x8C соответствует 140

byte blue = Convert.ToByte(0x51); // 0x51 соответствует 81

byte alpha = Convert.ToByte(0xFF); // 0xFF соответствует 255

// Создаем новый цвет на основе полученных компонентов

System.Drawing.Color newColor = System.Drawing.Color.FromArgb(alpha, red, green, blue);

// Создаем новую кисть с этим цветом

System.Drawing.Brush brush = new System.Drawing.SolidBrush(newColor);

float x = initialX + offsetX; //Вычисляем x на основе начального значения и случайного смещения

g.DrawString(letter.ToString(), font, brush, x, y + offsetY);

initialX += letterWidth;

}

}

return bitmap;

}//конец капчи

private void AddNoiseToCaptcha(Bitmap captchaImage)

{

Random random = new Random();

using (Graphics g = Graphics.FromImage(captchaImage))

{

for (int i = 0; i < 20; i++)

{

System.Drawing.Pen pen = new System.Drawing.Pen(System.Drawing.Color.Black, 1);

int x1 = random.Next(captchaImage.Width);

int y1 = random.Next(captchaImage.Height);

int x2 = random.Next(captchaImage.Width);

int y2 = random.Next(captchaImage.Height);

g.DrawLine(pen, x1, y1, x2, y2);

//добавление случайных точек

for (int j = 0; j < 50; j++)

{

int x = random.Next(captchaImage.Width);

int y = random.Next(captchaImage.Height);

captchaImage.SetPixel(x, y, System.Drawing.Color.Gray);

}

}

}

}

private void DisplayCaptchaImage()

{

Bitmap captchaImage = GenerateCaptchaImage();

AddNoiseToCaptcha(captchaImage);

BitmapSource bitmapSource = ConvertBitmapToBitmapSource(captchaImage);

CaptchaImage.Source = bitmapSource;

}

private BitmapSource ConvertBitmapToBitmapSource(Bitmap captchaImage)

{

throw new NotImplementedException();

}

**Программный код проверки времени работы лаборанта:**

public int Hours = 0, Mins = 0, Sec = 0;

private int secSave = 0, minSave = 0, hourSave = 0;

private DispatcherTimer Time;

public LaborantAnalizator(Users users)

{

InitializeComponent();

Time = new DispatcherTimer();

TimeFromRegistracia();

Time.Interval = TimeSpan.FromSeconds(1);

Time.Tick += TimerLabel\_Tick;

Time.Start();

}

//Время сеанса

private void TimerLabel\_Tick(object sender, EventArgs e)

{

Sec++;

if (Sec == 60)

{

Mins++;

Sec = 0;

}

if (Mins == 60)

{

Hours++;

Mins = 0;

}

TimerLabel.Content = $"{Hours}:{Mins}:{Sec}";

if (Hours == 2 && Mins == 25)

MessageBox.Show("Через 5 минут сеанс будет закончен!");

if (Hours == 2 && Mins == 30)

{

SetBlock();

Application.Current.Shutdown();

}

}

private void SetBlock()

{

secSave = GetSec();

minSave = GetMins();

hourSave = GetHours();

CreateTimeReg(secSave, minSave, hourSave);

}

private void CreateTimeReg(int secSave, int minSave, int hourSave)

{

RegistryKey regTime = Registry.CurrentUser;

RegistryKey REG\_TIME = regTime.CreateSubKey("REG\_TIME");

REG\_TIME.SetValue("seconds", Sec.ToString());

REG\_TIME.SetValue("minutes", Mins.ToString());

REG\_TIME.SetValue("hours", Hours.ToString());

REG\_TIME.Close();

}

private int GetMins()

{

return Convert.ToInt32(DateTime.Now.ToString().Substring((DateTime.Now.ToString().Length - 5), 2));

}

private int GetHours()

{

return Convert.ToInt32(DateTime.Now.ToString().Substring((DateTime.Now.ToString().Length - 8), 2));

}

private int GetSec()

{

return Convert.ToInt32(DateTime.Now.ToString().Substring((DateTime.Now.ToString().Length - 2), 2)); ;

}

private void TimeFromRegistracia()

{

RegistryKey time = Registry.CurrentUser.OpenSubKey("Timer");

if (time != null)

{

Hours = (int)time.GetValue("Houers", 0);

Mins = (int)time.GetValue("Minutes", 0);

Sec = (int)time.GetValue("Seconds", 0);

time.Close();

}

}

**Программный код библиотеки классов:**

public class Class1

{

public static bool Check\_Mail(string Mail)//почта

{

bool IsTrue = false;

string pattern = @"(\w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*)";

if (Regex.IsMatch(Mail, pattern, RegexOptions.IgnoreCase))

IsTrue = true;

return IsTrue;

}

public static bool Check\_Password(string Password)//пароль

{

bool IsTrue = false;

string Pattern = @"^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*[0-9])(?=.\*[^a-zA-Z0-9]).{1,16}$";

if (Regex.IsMatch(Password, Pattern, RegexOptions.IgnoreCase))

IsTrue = true;

return IsTrue;

}

public static bool Check\_Login(string Login)//логин

{

bool IsTrue = false;

string Pattern = @"^[a-zA-Z0-9]{6,}$";

if (Regex.IsMatch(Login, Pattern, RegexOptions.IgnoreCase))

IsTrue = true;

return IsTrue;

**Задание 2**

**Прием биоматериала и формирование заказ**а

Лаборант получает от курьера биоматериал из поликлиник города. Каждый биоматериал приходит с листом сопровождения. Часть пробирок приходит со штрих-кодом, часть - без кода, поэтому такой код нужно ввести самостоятельно.

При приеме биоматериала лаборант формирует заказ. В данном случае заказа – это занесение в базу данных кода биоматериала и перечень услуг (далее - исследований), которые могут быть произведены пациентам.

Реализуйте интерфейс для приема биоматериала и формирования заказа.

По нажатию на элемент интерфейса для принятия биоматериала открывается окно формирования заказа, в котором первое, что необходимо – это ввести код пробирки.

Код пробирки можно ввести вручную (если пробирка пришла без штрих-кода) или считать сканером.

Реализуйте поле для ручного ввода кода пробирки, при этом номер пробирки должен появиться в виде подсказки в поле ввода – последний номер заказа в БД+1. Лаборант может подтвердить его нажатием Enter или ввести другой номер, не дублирующий уже существующий в БД, кроме заказов в архиве. По нажатию на Enter формируется штрих-код и сохраняется в формате pdf для печати на клейкой бумаге.

Когда лаборант формирует штрих-код для пробирки с биоматериалом, штрих-код должен содержать в себе цифры из уникального идентификатора заказа и даты его создания + уникальный код из 6 символов (например: 5 1 4 0 9 2 0 2 0 1 2 3 4 5 6). Штрих-код должен быть представлен в числовом виде и графически (в виде полосок). Сгенерированный штрих код должен быть сохранен в pdf файл, так как лаборант будет печатать его на клейкой бумаге. Номинальные размеры символа штрихового кодирования:

• высота символа - 25,93 мм;

• высота штриха - 22,85 мм;

• свободная зона слева - 3,63 мм;

• свободная зона справа - 2,31 мм;

• штрихи, которые образуют правый и левый ограничивающие знаки, а также центральный ограничивающий знак должны быть удлинены вниз на 1,65 мм.

• в символе номинального размера высота цифр - 2,75 мм;

• минимальный размер от верхнего края цифр до нижнего края штрихов - 0,165 мм;

• каждая цифра отражается в виде черной полоски, которая имеет ширину 0,15мм\*{цифра}. Например, если цифра 5, то ширина полоски будет 0.15\*5 мм;

• если цифрой является нуль, то необходимо оставлять белый штрих в 1.35мм.

• между штрихами должно быть расстояние в 0.2мм

Цветовое исполнение: возможные цвета штрихов черный, синий, темно-зеленый, темно-коричневый; цвет пробелов (и фона) белый, желтый, оранжевый, светло-коричневый. Не допускаются красный и желтый цвета для штрихов – они не считываются сканером.

После ввода кода пробирки в окне формирования заказа реализуйте возможность ввода ФИО пациента, услуги. Если пациента в базе данных лаборатории нет, его необходимо добавить. Для этого открывается дополнительное модальное окно добавления пациента, в котором нужно ввести ФИО, дата рождения, серия и номер паспорта, телефон, e-mail, номер страхового полиса, тип страхового полиса (выбор из списка), название страховой компании (выбор из списка).

Реализуйте возможность добавления нескольких услуг к заказу.

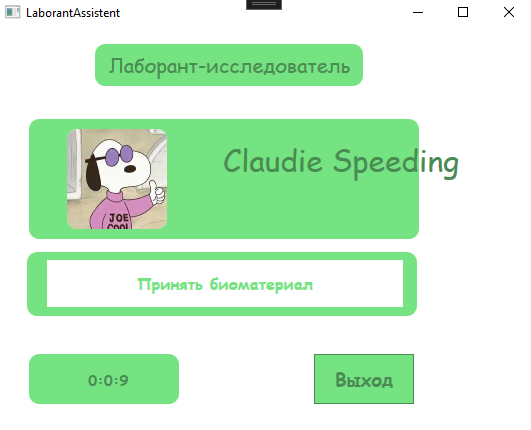
При формировании заказа стоимость рассчитывается автоматически.

**Добавление новых сотрудников**

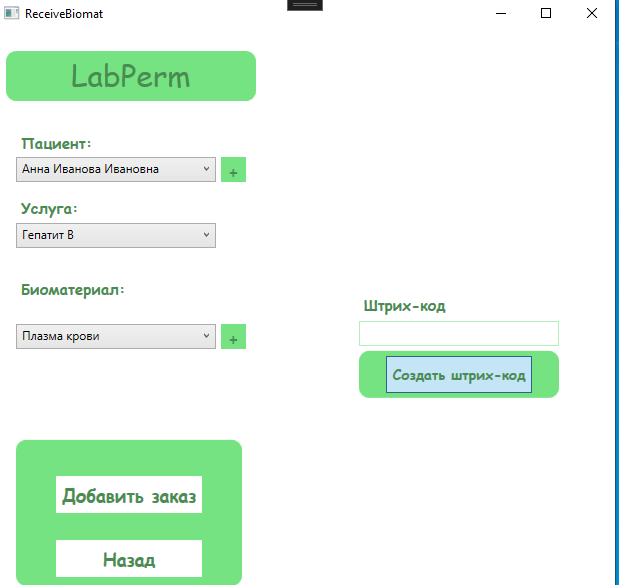
Реализуйте для администратора возможность добавлять новых сотрудников лаборатории. Для нового сотрудника необходимо будет обязательно указать: должность, ФИО, номер телефона, электронную почту, логин и пароль для входа в программу. Электронная почта, логин и пароль при окончании ввода или добавлении пользователя должны проверятся с помощью разработанной библиотеки.

Если администратор выбирает должность лаборанта, то должна быть реализована возможность указать набор услуг, которые он может оказывать.

**Скриншоты форм лаборанта с заполненными данными:**

****

**Скриншоты формы заказа (принятия биоматериала) с заполненными данными:**

****

**Программный код сохранения штрих-кода:**

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

{

System.Drawing.Image image = null;

BarcodeLib.Barcode b = new BarcodeLib.Barcode();

b.BackColor = System.Drawing.Color.White;

b.ForeColor = System.Drawing.Color.Black;

b.IncludeLabel = true;

b.Alignment = BarcodeLib.AlignmentPositions.CENTER;

b.LabelPosition = BarcodeLib.LabelPositions.BOTTOMCENTER;

b.ImageFormat = System.Drawing.Imaging.ImageFormat.Png;

System.Drawing.Font font = new System.Drawing.Font("verdana", 10f);

b.LabelFont = font;

b.Height = 100;

b.Width = 200;

image = b.Encode(BarcodeLib.TYPE.CODE128C, bioorder.ID);

image.Save($@"C:\tmp\barcode{bioorder.ID}.png");

MessageBox.Show("Штрих-код был добавлен!");

}

**Программный код добавления заказа в базу данных:**

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

{

if (comboBoxBiomat.SelectedItem != null)

{

var id = db.Biomaterial.FirstOrDefault(x => x.Name == comboBoxBiomat.SelectedItem.ToString());

var bioPatient = new Order

{

DateOfCreation = DateTime.Now.Date,

ID\_OrderStatus = 2,

ID\_User = Users.ID,

ID\_Patient = patient.ID,

ID\_Biomaterial = id.ID

};

db.Order.Add(bioPatient);

db.SaveChanges();

MessageBox.Show("Все прошло успешно!");

Window takeBio = new ReceiveBiomat(Users);

takeBio.Show();

this.Close();

}

}

**Задание 3**

**Работа с анализатором**

Работа с анализатором доступна только лаборанту-исследователю, который при входе в окно анализатора видит список невыполненных услуг.

Анализатор — это медицинский прибор, на котором проводится исследование биоматериала.

Биоматериал загружается в анализатор, специальная система дозирования отправляет биоматериал на реактив, сенсор реактива выдает некоторый показатель - число, текст, знак. Это и есть результат исследования.

Задача нашей информационной системы - автоматизировать отправление материала на анализатор и получение результатов.

Для эмуляции анализатора Вам необходимо запустить приложение LIMSAnalyzers.exe на своей виртуальной машине, методы по работе с которым будут описаны ниже.

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.

Отправка услуги на анализатор должна быть реализована с помощью API по HTTP в формате JSON. Максимальное время ожидания работы анализатора - 30 секунд. После отправки запроса на анализатор, необходимо изменить статус у услуги на “отправлена на исследование”.

Результаты всех исследований заносятся в базу данных.

Отправка услуги на анализатор не должна блокировать интерфейс (должна быть в

параллельном потоке).

Реализуйте возможность асинхронной работы лаборанта-исследователя с несколькими анализаторами, то есть лаборант-исследователь может перейти в окно другого анализатора и отправить услугу на исследование.

После отправки услуги на анализатор, необходимо периодически опрашивать анализатор на предмет готовности исследования и получения результатов. При этом, около конкретной услуги необходимо показывать анимацию loader для индикации процесса исследования, а также процентный статус исследования. При получении результата с анализатора, лаборант должен одобрить результат. Если значения, полученные с анализатора, отклоняются от среднего в 5 раз, должно появится сообщение о возможном сбое исследования или некачественном биоматериале.

Но данное сообщение лаборант может закрыть и одобрить правильность исследования.

Если лаборант одобрил результат, то услуга получает статус “выполнена”. Если лаборант не одобрил результат, то услуга получает статус - необходим повторный забор биоматериала.

Реализуйте в интерфейсе анализатора возможность просмотра списка невыполненных услуг (исследований) и выполняющихся услуг со статусом, полем для вывода результата и кнопки одобрения услуги. Анализатор выдает процентный статус выполнения исследования, отобразите это в интерфейсе выполняющихся услуг.

Если все анализаторы в работе (заняты), мы не можем отправить на анализатор новые услуги

Код исследования,Название,Стоимость,Тип результата,Доступные анализаторы

619,TSH,262.71, Integer, Ledetect | Biorad

311,Амилаза,361.88, Integer, Ledetect

548,Альбумин,234.09, Integer, Biorad

258,Креатинин,143.22, Integer, Biorad | Ledetect

176,Билирубин общий,102.85, Integer, Biorad

501,Гепатит В,176.83, String, Ledetect

543,Гепатит С,289.99, String, Ledetect | Biorad

557,ВИЧ,490.77, String, Ledetect

229,СПИД,341.78, String, Ledetect

415,Кальций общий,419.9, Integer, Ledetect

323,Глюкоза,447.65, Integer, Ledetect

855,Ковид IgM,209.78, String, Biorad

346,Общий белок,396.03, Integer, Ledetect

836,Железо,105.32, Integer, Biorad

659,Сифилис RPR,443.66, String, Ledetect | Biorad

797,АТ и АГ к ВИЧ 1/2,370.62, String, Biorad

287,Волчаночный антикоагулянт,290.11, String, Biorad

Описание:

Отправка заказа: POST http://localhost:5000/api/analyzer/{name}

Content-Type: application/json

Структура запроса: JSON { “patient”: “{id}”, “services”: [{ “serviceCode”: 000 }, { “serviceCode”: 000

}, ….] }

При успешном создании возвращается код 200. Если возникла ошибка код 400 с текстом ошибки.

Получение результатов/статуса: GET http://localhost:5000/api/analyzer/{name}

Content-Type: application/json

Структура ответа: JSON { “patient”: “{id}”, “services”: [{ “сode”: 000, “result”: “” }, { “code”: 000,

“result”: “” }, ….] }

Или статус заказа (если он еще не выполнен, в процентах) JSON { “progress”: 000 }

При успешном получении код 200.

**Комментарии**

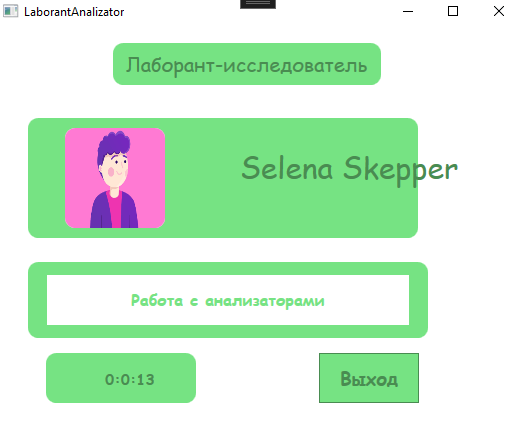
Пропишите для разработанной программы комментарии XML, благодаря которым в будущем будет возможно автоматически сгенерировать документацию.

**Загрузка разработанной программы в GOGS**

Загрузите разработанную программу в репозиторий GOGS.

Создайте файл ReadMe и заполните его в соответствующей структуре.

**Скриншоты формы лаборанта-исследователя с заполненными данными:**



**Программный код получения услуг с исследования:**

**Задание 4**

**Оформление диаграммы прецедентов**

Разработать диаграмму прецедентов для разработанной программы

**Оформление функциональной диаграммы**

Разработать функциональную диаграмму разработанной программы уровня А1

**Оформление блок-схемы**

Разработать блок-схему разработанной программы в соответствии с ГОСТ 19.701-90

**Оформление руководства системного программиста**

Разработать руководство системного программиста по разработанной программе в соответствии с ЕСПД

**Оформление руководства оператора**

Разработать руководство оператора по разработанной программе в соответствии с ЕСПД

**Оформление протоколов тестирования**

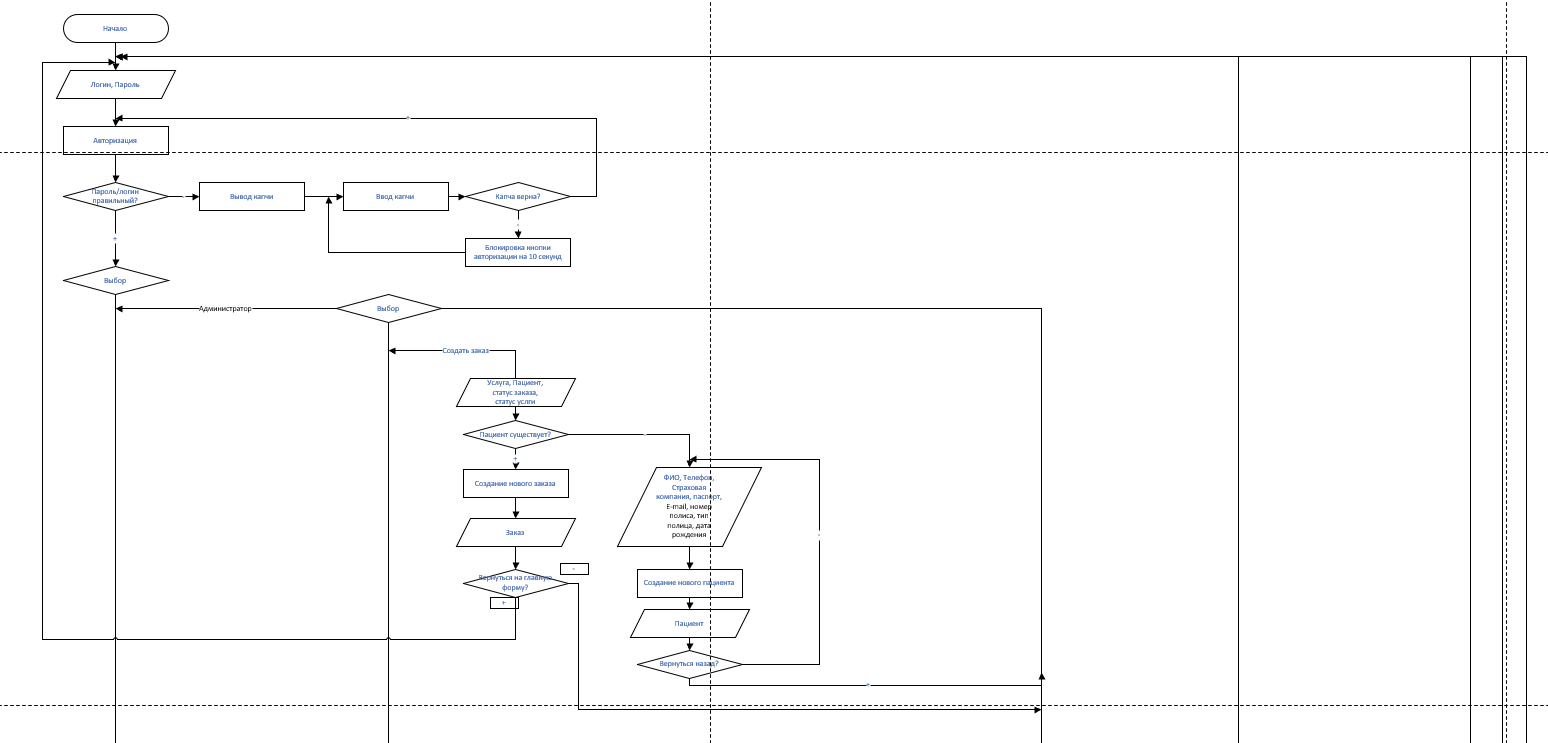
Проведите ручное тестирование разработанной программы. Заполните протоколы для трех функций в нормальных, экстремальных и исключительных условиях:

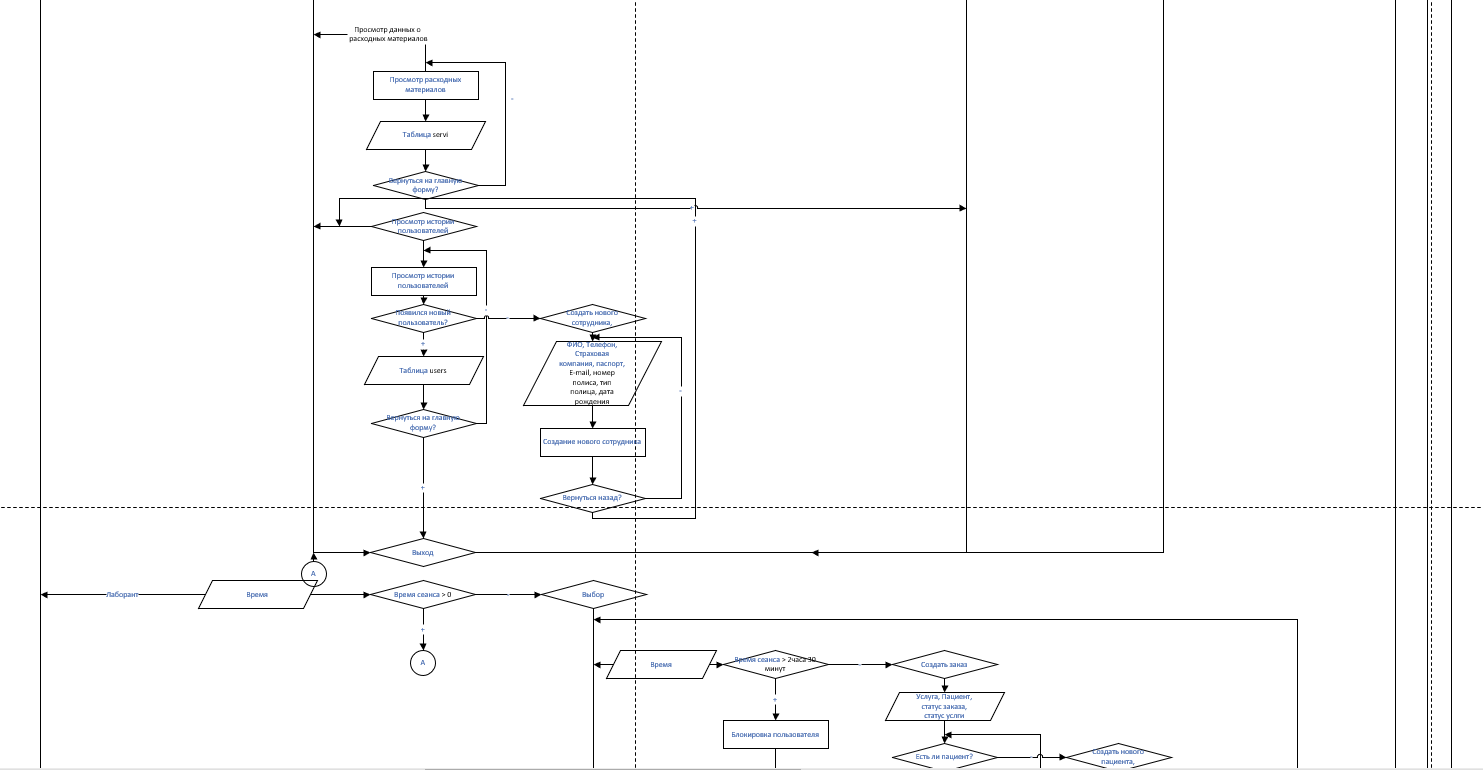
*Функция 1:* создание нового сотрудника администратором

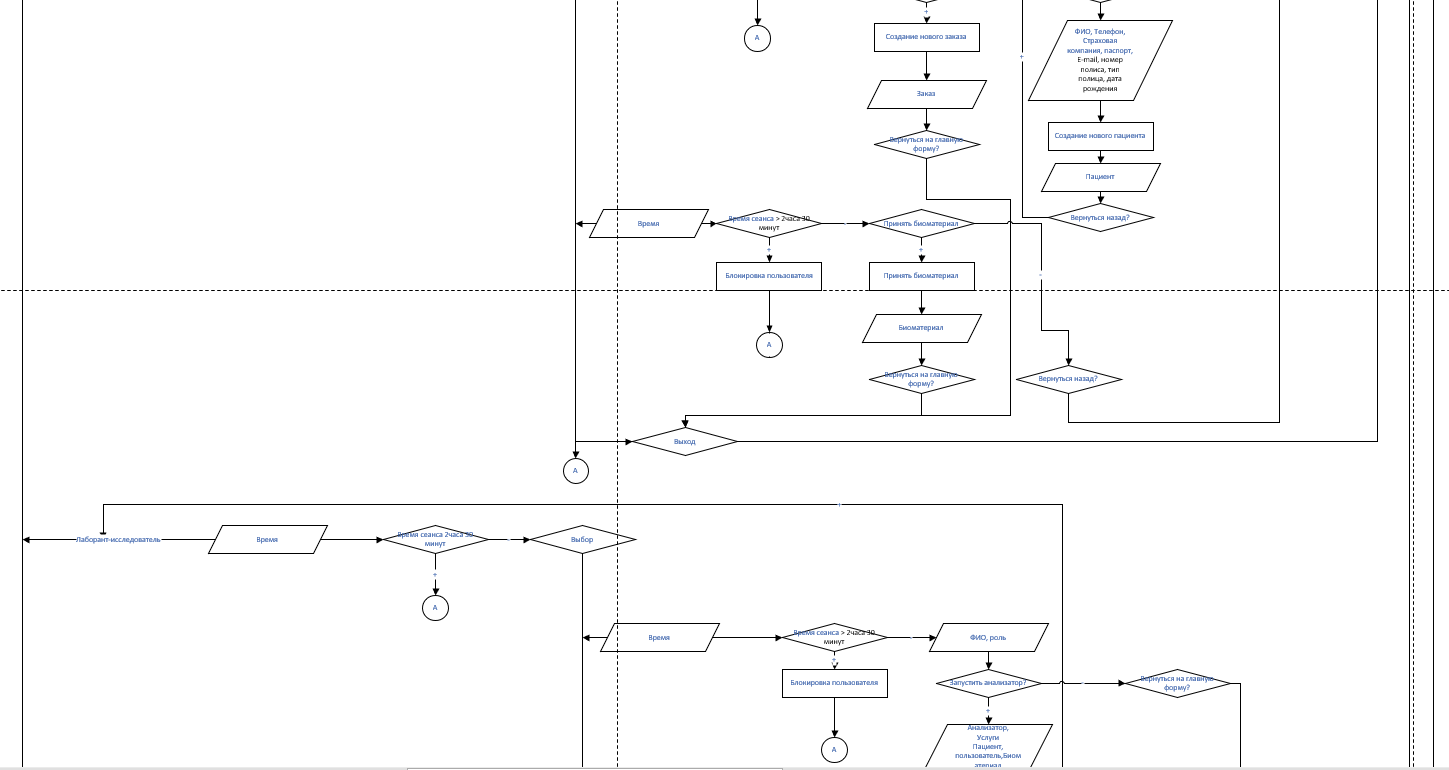
*Функция 2:* формирование заказа лаборантом

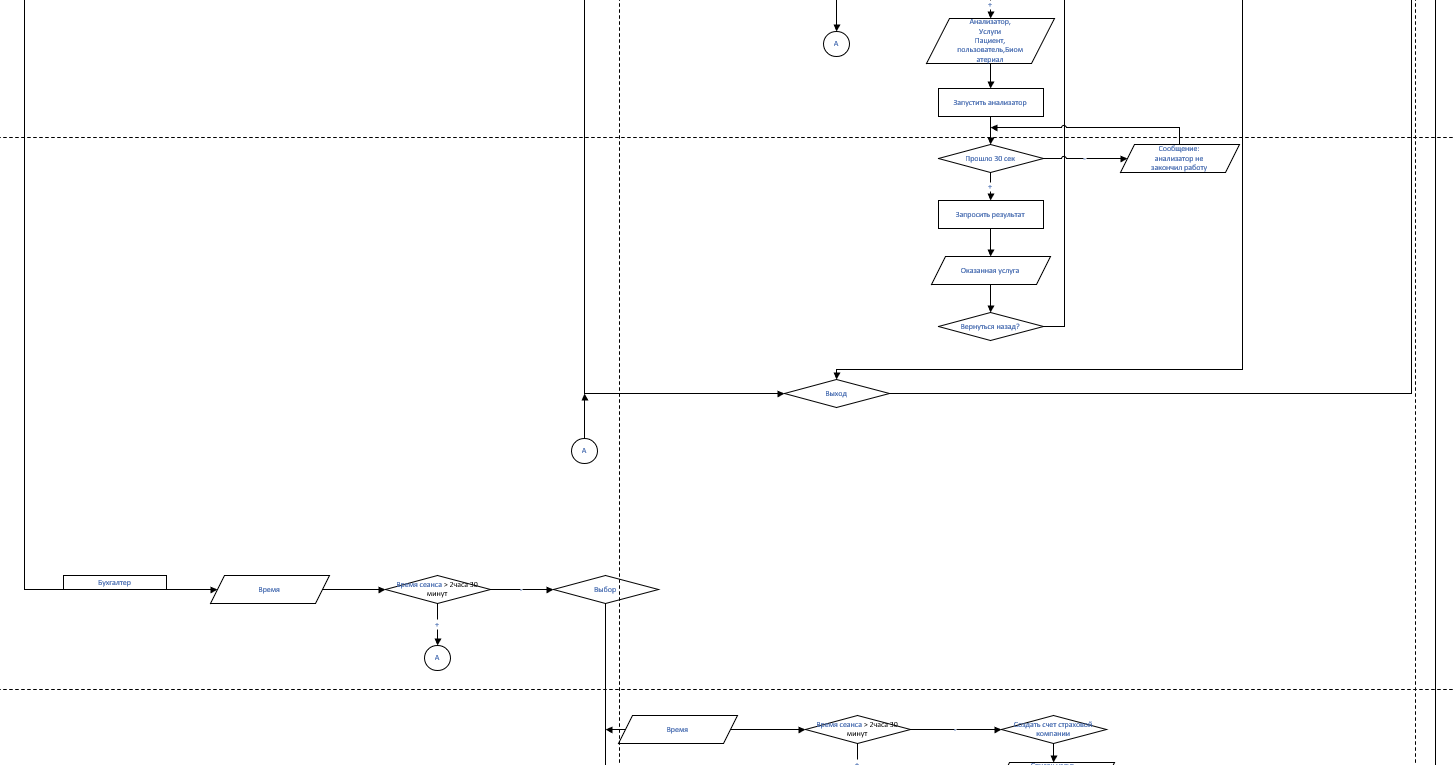
*Функция 3:* отправка биоматериала на анализ

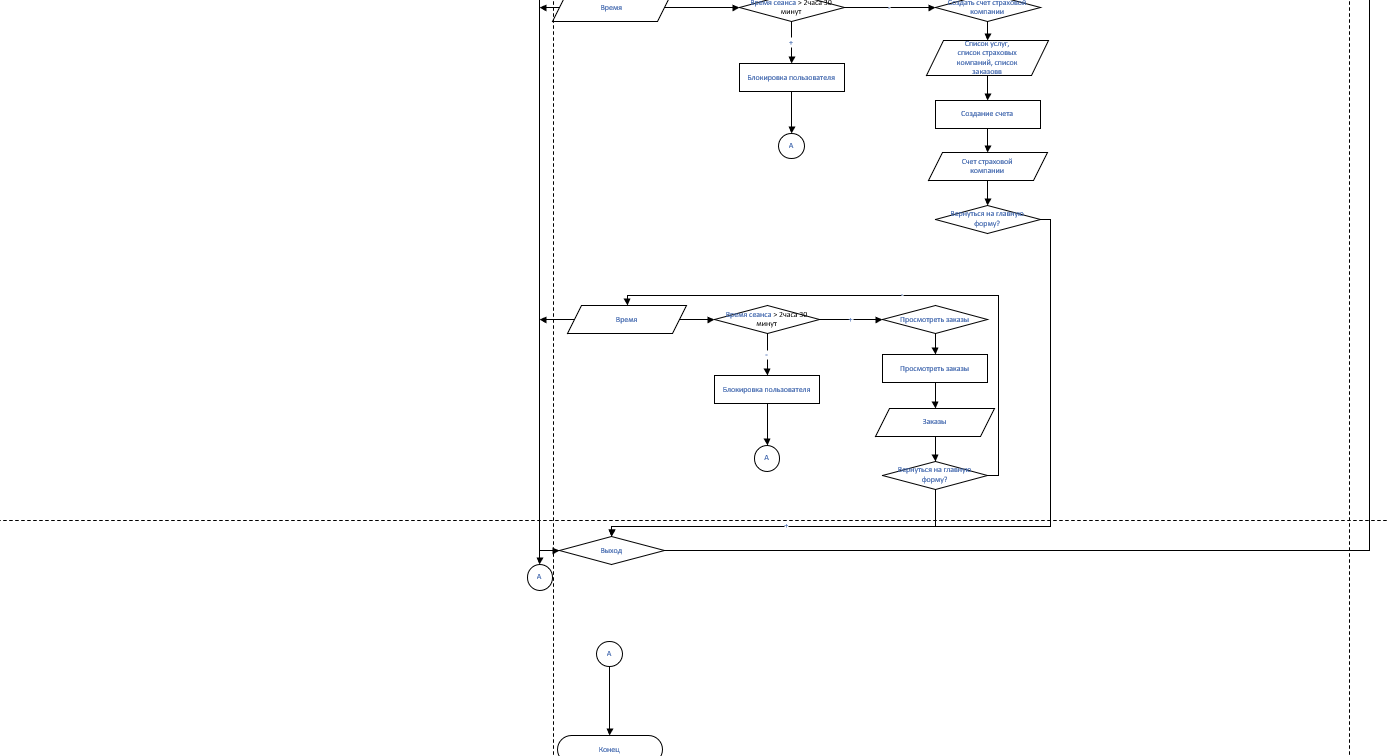
**Скриншот блок-схемы**



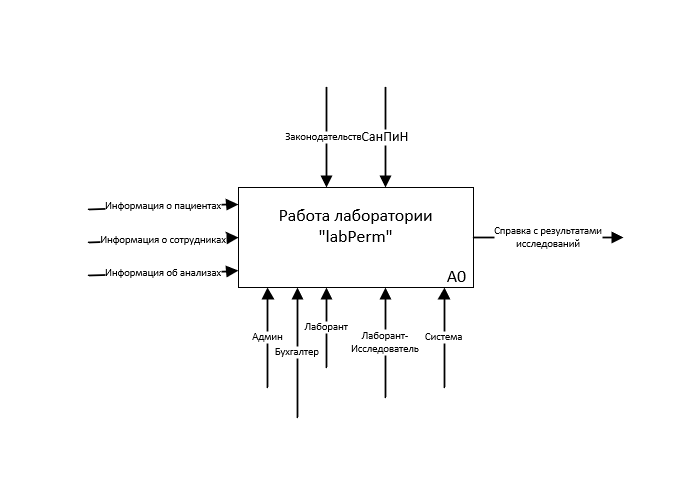


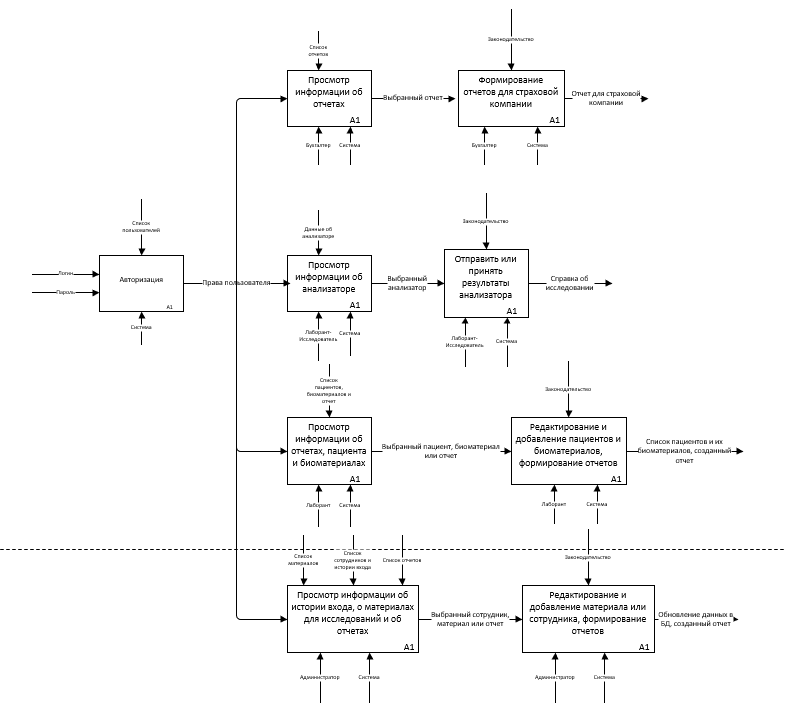




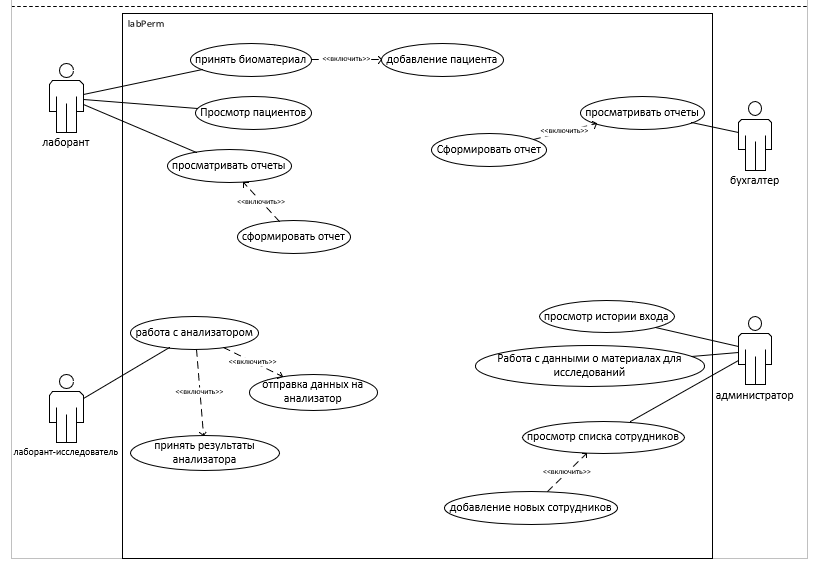


**Скриншот функциональной диаграммы (А0, А1)**





**Скриншот диаграммы прецедентов**



**Руководство оператора (можно скринами)**

**Руководство системного программиста (можно скринами)**

**Заполненные протоколы тестирования**

|  |  |
| --- | --- |
| **Наименование** | **Описание** |
| **Наименование проекта** | Практическое задание. Учебная Практика 01 |
| **Номер версии** | 1.0 |
| **Имя тестера** | Каменских Аня |
| **Даты тестирования** | 08.06.2024 |
| **Test Case #** | 1 |
| **Приоритет теста** *(низкий/средний/высокий)* | Высокий |
| **Наименование варианта тестирования** | Создание нового сотрудника администратором |
| **Резюме испытания** | При создании нового пользователя необходимо, чтобы его данные заносились в систему |
| **Шаги тестирования** | 1. Авторизоваться под должностью администратора 2. Перейти на форму «Добавить пользователя», после нажатия на соответствующую кнопку 3. Заполнить все поля 4. Нажать на кнопку «Добавить» |
| **Тестовые данные** | Логин и пароль для входа в систему, личные данные пользователя для добавления в систему |
| **Ожидаемый результат** | При добавлении пользователя, должно выйти окно «Пользователь успешно добавлен» и появится в списке пользователей системы |
| **Фактический результат** | При добавлении пользователя, должно выйти окно «Пользователь успешно добавлен» и появится в списке пользователей системы |
| **Предусловия** | Любые предварительные действия, которые должны быть выполнены перед проведением тестирования. Перечислите предварительные условия для успешного выполнения проекта |
| **Постусловия** | После выполнения тестирования данные нового пользователя будут записаны в системе и выйдет окно-сообщение об успешном выполнении операции: |
| **Статус** *(Pass/Fail)* | Pass |
| **Комментарии** |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Наименование** | **Описание** |
| **Наименование проекта** | Практическое задание. Учебная Практика 01 |
| **Номер версии** | 1.0 |
| **Имя тестера** | Каменских Аня |
| **Даты тестирования** | 08.06.2024 |
| **Test Case #** | 2 |
| **Приоритет теста** *(низкий/средний/высокий)* | Высокий |
| **Наименование варианта тестирования** | формирование заказа лаборантом |
| **Резюме испытания** | По окончанию формирования заказа лаборантом должно выйти окно-сообщение об успешном добавлении заказа |
| **Шаги тестирования** | 1. Авторизоваться под должностью лаборанта 2. Перейти на форму «Получить биоматериал», после нажатия на соответствующую кнопку 3. Заполнить все поля 4. Нажать на кнопку «Добавить заказ» |
| **Тестовые данные** | Логин и пароль для входа в систему, данные о биоматериалах и пациентах |
| **Ожидаемый результат** | После оформления заказа должно выйти окно-сообщение «Заказ успешно добавлен» |
| **Фактический результат** | Какой фактический результат после выполнения теста? Опишите любое соответствующее поведение системы после выполнения тестирования. |
| **Предусловия** | Любые предварительные действия, которые должны быть выполнены перед проведением тестирования. Перечислите предварительные условия для успешного выполнения проекта |
| **Постусловия** | Должно выйти окно-сообщение об успешном формировании заказа |
| **Статус** *(Pass/Fail)* | Pass |
| **Комментарии** | Если при заполнении полей за оформление заказа в списке пациентов не будет того, который вам нужен для оформления заказа, то необходимо будет самостоятельно его добавить. Для этого необходимо нажать на кнопку «+» возле выпадающего списка с пациентами лаборатории, после этого вас перенесет на форму «Добавление пациентов», где надо будет заполнить все необходимые поля и нажать на соответствующую кнопку добавления. По завершению регистрации пациента вы увидите окно-сообщение об успешном добавлении пациента. |

|  |  |
| --- | --- |
| **Наименование** | **Описание** |
| **Наименование проекта** | Практическое задание. Учебная Практика 01 |
| **Номер версии** | 1.0 |
| **Имя тестера** | Каменских Аня |
| **Даты тестирования** | 08.06.2024 |
| **Test Case #** | 3 |
| **Приоритет теста** *(низкий/средний/высокий)* | Средний |
| **Наименование варианта тестирования** | Отправка биоматериала на анализ |
| **Резюме испытания** | При отправке на анализатор должно выйти окно сообщение об успешном выполнении процесса |
| **Шаги тестирования** | 1. Авторизоваться под должностью лаборант-исследователя 2. Открыть файл LIMSAnalyzers.exe 3. Перейти на форму «Отправка анализатора», после нажатия на соответствующую кнопку 4. Нажать на отправку нужного нам биоматериала |
| **Тестовые данные** | Логин и пароль для входа под должностью лаборант-исследователя |
| **Ожидаемый результат** | Выбранный биоматериал должен быть успешно отправлен на анализ, об этом нам сообщит окно-сообщение. |
| **Фактический результат** | Какой фактический результат после выполнения теста? Опишите любое соответствующее поведение системы после выполнения тестирования. |
| **Предусловия** | Любые предварительные действия, которые должны быть выполнены перед проведением тестирования. Перечислите предварительные условия для успешного выполнения проекта |
| **Постусловия** | Какое состояние должно быть у системы после выполнения тестирования? |
| **Статус** *(Pass/Fail)* | Если фактический результат не соответствует ожидаемым результатам отметка, что тест провалился (fail). В противном случае как прошел (pass) |
| **Комментарии** |  |

**СПИСОК ЛИТЕРАТУРЫ**

1. METANIT. Сайт о программировании. [Электронный ресурс] URL: [Реляционные базы данных и язык SQL (metanit.com)](https://metanit.com/sql/?ysclid=lx8m04jitw654187101)
2. METANIT. Сайт о программировании. [Электронный ресурс] URL: [C# и .NET | Таймеры (metanit.com)](https://metanit.com/sharp/tutorial/11.9.php?ysclid=lx8mcncys707318350)
3. METANIT. Сайт о программировании. [Электронный ресурс] URL: [C# и Windows Forms | Метки и ссылки (metanit.com)](https://metanit.com/sharp/windowsforms/4.2.php?ysclid=lx8mdoo26a292264139)
4. METANIT. Сайт о программировании. [Электронный ресурс] URL: [C# и WPF | Элементы управления (metanit.com)](https://metanit.com/sharp/wpf/5.1.php?ysclid=lx8metzgf7678929040)

**ДНЕВНИК ПРОХОЖДЕНИЯ УЧЕБНОЙ ПРАКТИКИ**

Обучающийся (аяся) Каменских Анна Алексеевна Группы ИС-21-3

специальности СПО 09.02.07 «Информационные системы и программирование» проходил(а) учебную практику по профессиональному модулю   
**ПМ.01 «**Разработка модулей программного обеспечения для компьютерных систем» в период с «8» апреля 2024 г. по «13» апреля 2024 г (рассредоточено) и в период с «13» мая 2024 г. по «18» мая 2024 г (рассредоточено)

на предприятии / в организации ГБПОУ «ПХТТ»

За время практики выполнил(а) следующие виды работ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Дата** | **Кол-во часов** | **Наименование вида работ** | **Оценка** | **Подпись** |
| 26.03.2024  27.03.2024  28.03.2024 | 22 | Инструктаж по технике безопасности и пожарной безопасности.  Разработка приложения с разграничением прав доступа: механизм авторизации, реализация функционала по работе с БД (добавление, изменение, удаление данных) |  |  |
| 29.03.2024  30.03.2024 | 14 | Разработка приложения в соответствии с функциональным назначением программы, работа по созданию интерфейса программы, создание модуля обработки данных, создание модуля формирования штрихкода. Формирование документов по шаблону с возможностью сохранения в разных форматах (.docx /.xlsx/.jpg/.pdf) |  |  |
| 1.04.2024  2.04.2024 | 12 | Разработка дополнительного функционала в соответствии с техническим заданием. Создание модуля по работе с API по HTTP, работа с JSON-файлом |  |  |
| 3.04.2024 | 6 | Разработка алгоритмов программных модулей в соответствии с техническим заданием |  |  |
| 4.04.2024 | 6 | Разработка технической документации по программному продукту |  |  |
| 5.04.2024 | 6 | Тестирование и отладка программных модулей. |  |  |
| 6.04.2024 | 6 | Оформление отчетной документации. Презентация проекта. |  |  |

|  |  |
| --- | --- |
|  | Итоговая оценка за практику \_\_\_\_\_\_\_ |

Руководитель практики от ГБПОУ «ПХТТ»

преподаватель, Токарева Полина Николаевна

|  |  |  |
| --- | --- | --- |
| Дата 18.06.2024 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | /П.Н. Токарева /  Расшифровка подписи |