Министерство сельского хозяйства Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Пермский государственный аграрно-технологический университет

имени академика Д.Н. Прянишникова»

Кафедра Информационных технологий

и программной инженерии

**ОТЧЁТ ПО УЧЕБНОЙ ПРАКТИКЕ**

(проектно-технологическая практика)

на тему: «Разработка информационной системы

для автоматизации бизнес-процесса»

Выполнил:

студент группы ПИб-21

направления подготовки

09.03.03 Прикладная информатика

Дегтярева Анастасия Сергеевна

Проверил:

доцент кафедры ИТиПИ, к.т.н., доцент

Беляков Андрей Юрьевич

2021

Содержание

[Постановка задачи на проектирование 3](#_Toc75987885)

[Анализ средств для разработки 4](#_Toc75987886)

[Разработка информационной системы 5](#_Toc75987887)

[Заключение 6](#_Toc75987888)

[Список источников 7](#_Toc75987889)

[ПРИЛОЖЕНИЕ А 8](#_Toc75987890)

[ПРИЛОЖЕНИЕ Б 9](#_Toc75987891)

# Постановка задачи на проектирование

Данная информационная система будет решать задачи по автоматизации процессов записи тем ВКР. Преимущественно ей будут пользоваться лаборанты и преподаватели, которые будут записывать темы ВКР, выбранные студентами.

1.Требования к функционалу программной реализации:

Данная программа должна выполнять регистрацию пользователя, запись регистрационных данных пользователя в БД, авторизацию пользователя, сохранение данных для заявления ВКР в Word и в БД.

2.Требования к интерфейсу:

Удобное размещение кнопок, логически понятные названия кнопок, удоюная последовательность при переключении кнопок.

3.Требования к защищённости:

Программа должна работать с заданными исходными данными в соответствии с алгоритмом функционирования, выдавать сообщения об ошибках при неверно заданных исходных данных, поддерживать диалоговый режим в рамках представляемых пользователю возможностей.

4.Требования к кроссплатформенности:

Программа должна быть кроссплатформлена.

5.Требования к операционной системе:

Выполнение функций эффективного управления ресурсами и обеспечения удобного интерфейса для пользователя и прикладных программ. ОС должна реализовывать мультипрограммную обработку, виртуальную память, свопинг, поддерживать многооконный интерфейс, а также выполнять многие другие совершенно необходимые функции. Кроме этих функциональных требований к операционным системам предъявляются не менее важные рыночные требования - расширяемость, переносимость, надежность и отказоустойчивость, совместимость, безопасность.

# Анализ средств для разработки

Для разработки информационной системы были рассматрены такие реляционные базы данных как SQLite, MySQL, PostgreSQL.

SQLite — это библиотека, встраиваемая в приложение, которое её использует. Будучи файловой БД, она предоставляет отличный набор инструментов для более простой (в сравнении с серверными БД) обработки любых видов данных.

Когда приложение использует SQLite, их связь производится с помощью функциональных и прямых вызовов файлов, что повышает скорость и производительность операций.

Поддерживаемые типы данных -Null, Integer, Real, Text, Blob.

Преимущества:

**-Файловая:**вся база данных хранится в одном файле, что облегчает перемещение.

**-Стандартизированная:**SQLite использует SQL; некоторые функции опущены (RIGHT OUTER JOIN или FOR EACH STATEMENT), однако, есть и некоторые новые.

**-Отлично подходит для разработки и даже тестирования:**во время этапа разработки большинству требуется масштабируемое решение.  Недостатки:

**-Отсутствие пользовательского управления:**продвинутые БД предоставляют пользователям возможность управлять связями в таблицах в соответствии с привилегиями, но у SQLite такой функции нет.

**-Невозможность дополнительной настройки:**опять-таки, SQLite нельзя сделать более производительной, поковырявшись в настройках — так уж она устроена.

MySQL— это самая популярная из всех крупных серверных БД. Разобраться в ней очень просто, да и в сети о ней можно найти большое количество информации. Хотя MySQL и не пытается полностью реализовать SQL-стандарты, она предлагает широкий функционал. Приложения общаются с базой данных через процесс-демон.

Поддерживаемые типы данных: Tinyint, Smallint, Mediumint, Integer, Begint, Float, Double Precision, Decimal, Numeric, Date, Datetime, Timestamp, Time, Year, Char, Varchar, Tinyblob, Tinytext, Mediablob, Mediatext, Longblob, Longtext, Enum, Set.

Преимущества:

**-Простота:**MySQL легко устанавливается. Существует много сторонних инструментов, включая визуальные, облегчающих начало работы с БД.

**-Много функций:**MySQL поддерживает большую часть функционала SQL.

**-Безопасность:**в MySQL встроено много функций безопасности.

**-Мощность и масштабируемость:**MySQL может работать с действительно большими объёмами данных, и неплохо походит для масштабируемых приложений.

**-Скорость:**пренебрежение некоторыми стандартами позволяет MySQL работать производительнее, местами срезая на поворотах.

Недостатки:

**Известные ограничения:**по определению, MySQL не может сделать всё, что угодно, и в ней присутствуют определённые ограничения функциональности.

**Вопросы надёжности:**некоторые операции реализованы менее надёжно, чем в других РСУБД.

PostgreSQL— это самая продвинутая РСУБД, ориентирующаяся в первую очередь на полное соответствие стандартам и расширяемость. PostgreSQL пытается полностью соответствовать SQL-стандартам ANSI/ISO.

PostgreSQL отличается от других РСУБД тем, что обладает объектно-ориентированным функционалом, в том числе полной поддержкой концепта ACID.

Будучи основанным на мощной технологии Postgres отлично справляется с одновременной обработкой нескольких заданий. Поддержка конкурентности реализована с использованием MVCC, что также обеспечивает совместимость с ACID.

Поддерживаемые типы данных: **bigint,** **bigserial,** **bit, bit varying, boolean,** **box,bytea, character varying, character,** **cidr, circle, date, double precision,** **inet**, **integerinterval,** **line, lseg, macaddr, money,** **path, point, polygon,** **real, smallint, serial,** **text,** **time, timestamp, tsquery**, **tsvector, txid\_snapshot,** **uuid**, **xml.**

Преимущества

**-Полная SQL-совместимость**.

**-Сообщество:**PostgreSQL поддерживается опытным сообществом 24/7.

**-Поддержка сторонними организациями:**несмотря на очень продвинутые функции, PostgreSQL используется в многих инструментах, связанных с РСУБД.

**-Расширяемость:** PostgreSQL можно программно расширить за счёт хранимых процедур.

**-Объектно-ориентированность:**PostgreSQL — не только реляционная, но и объектно-ориентированная СУБД.

Недостатки

**-Производительность:**В простых операциях чтения PostgreSQL может уступать своим соперникам.

**-Популярность:**из-за своей сложности инструмент не очень популярен.

**-Хостинг:**из-за вышеперечисленных факторов проблематично найти подходящего провайдера.

На основании того, что нам необходимо создать многопользовательское приложение с быстрой обработкой данных, с поддержкой SQL, обеспечением высокого уровеня «противовзломности», а также использованием в связке с PHP, то MySQL является наилучшим вариантом.

Далее, рассмотрим языки программирования, которые можно использовать для разработки ИС - C#, Python, Node.js.

C# — современный объектно-ориентированный и типобезопасный язык программирования. C# позволяет разработчикам создавать множество типов безопасных и надежных приложений, работающих в экосистеме .NET. C# относится к широко известному семейству языков C

Преимущества:

-Поддержка подавляющего большинства продуктов Microsoft

-Бесплатность ряда инструментов для небольших компаний и некоторых индивидуальных разработчиков — Visual Studio, облако Azure, Windows Server, Parallels Desktop для Mac Pro и др.

-Типы данных имеют фиксированный размер (32-битный int и 64-битный long), что повышает «мобильность» языка и упрощает программирование, так как вы всегда знаете точно, с чем вы имеете дело.

-Автоматическая «сборка мусора» Это значит, что нам в большинстве случаев не придётся заботиться об освобождении памяти. Вышеупомянутая общеязыковая среда CLR сама вызовет сборщик мусора и очистит память.

-Большое количество «синтаксического «сахара» — специальных конструкций, разработанных для понимания и написания кода. Они не имеют значения при компиляции.

-Низкий порог вхождения. Синтаксис C# имеет много схожего с другими языками программирования, благодаря чему облегчается переход для программистов. Язык C# часто признают наиболее понятным и подходящим для новичков.

-С помощью Xamarin на C# можно писать программы и приложения для таких операционных систем, как iOS, Android, MacOS и Linux.

Недостатки:

-Приоритетная ориентированность на платформу Windows.

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным.

Достоинства:

-Простой минималистичный синтаксис: код легко писать, читать и поддерживать.

-Большая стандартная библиотека и много дополнительных библиотек.

-Большой выбор фреймворков.

-Поддерживает объектно-ориентированное программирование и другие парадигмы.

-Кроссплатформенность и поддержка почти всех современных систем.

Недостатки:

-Низкая скорость.

-Плохо подходит для разработки мобильных приложений.

-Из-за динамической типизации выше вероятность ошибки при запуске, нужно больше тестов.

-Не подходит для работы с памятью на низком уровне.

Node.js это — среда выполнения JavaScript.

Достоинства:

-Легкость и скорость написания

-Легковесность

-Относительная простота

-Node package manager (огромное количество библиотек которые могут быть установлены в одну строку)

-Каждая библиотека попадает в дерево зависимостей

Недостатки:

-Требует четкой архитектуры.

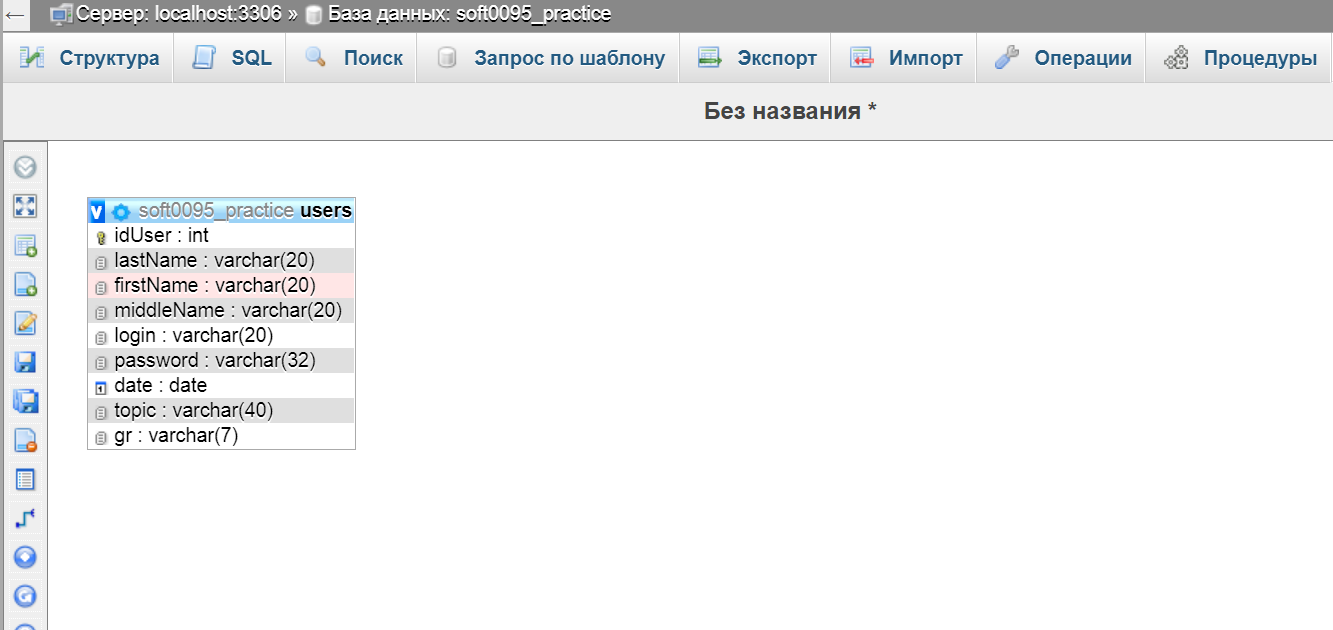
-Не может поддерживать задачи с высокой загрузкой процессора.

-Не развитая документация.

Таким образом, для разработки Windows –приложения наилучшим вариантом будет язык C# с разнообразными библиотеками и с удобными конструкциями.

# Разработка информационной системы

**Логическая модель базы данных**



На данной логической модели представлена таблица пользователей, состоящая из таких столбцов как идентификатор пользователя, фамилия, имя, отчество, логин, пароль, дата, название темы ВКР, номер группы.

**Запросы к базе данных**

1.Для проверки логина и пароля на форме авторизации. SQL-запрос "SELECT \* FROM `users` WHERE `login`= @log AND `password` = @pas"

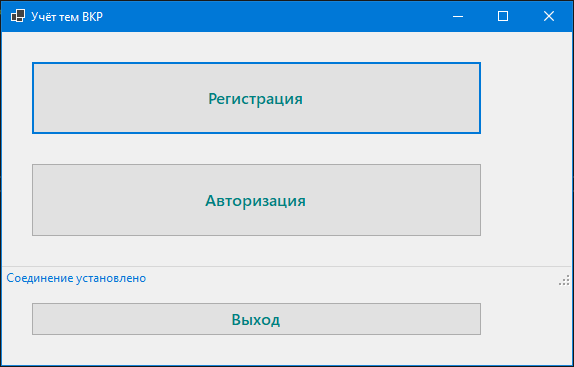
2. Для добавления в базу данных нового пользователя на форме регистрации. SQL-запрос "INSERT INTO `users` " +"(`lastName`, `firstName`, `middleName`, `Group`, `theme`, `date`, `login`,`password`) " + "VALUES(@ln,@fn, @mn, @gr, @th, @dt, @log, @pas)"

3. Для выбора id пользователя и передачи его на форму для заполнения её данными.SQL-запрос "SELECT \* FROM `users` WHERE idUser = @id"

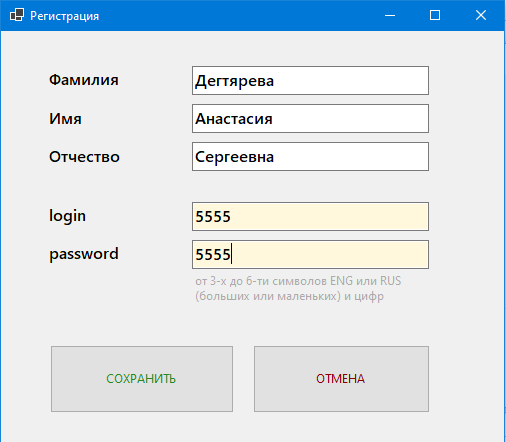
4. Для обновления информации в базе после ее изменения на форме. SQL-запрос "UPDATE `users` SET `group` = @gr, Date = @dt, theme = @th WHERE `idUser` = @id"

**Интерфейс пользователя**

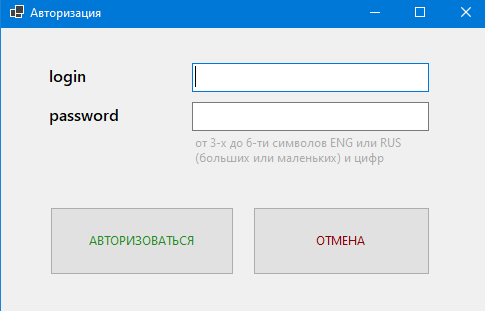
После запуска программы открывается стартовое окно с кнопками «Регистрация», «Авторизация» и «Выход».



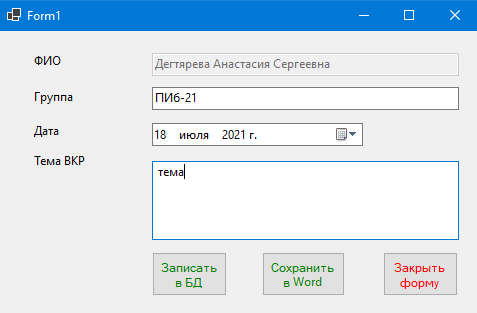
После нажатия кнопки «Регистрация» открывается окно регистрации пользователя, где пользователю нужно указать фамилию, имя, отчество и придумать логин и пароль.



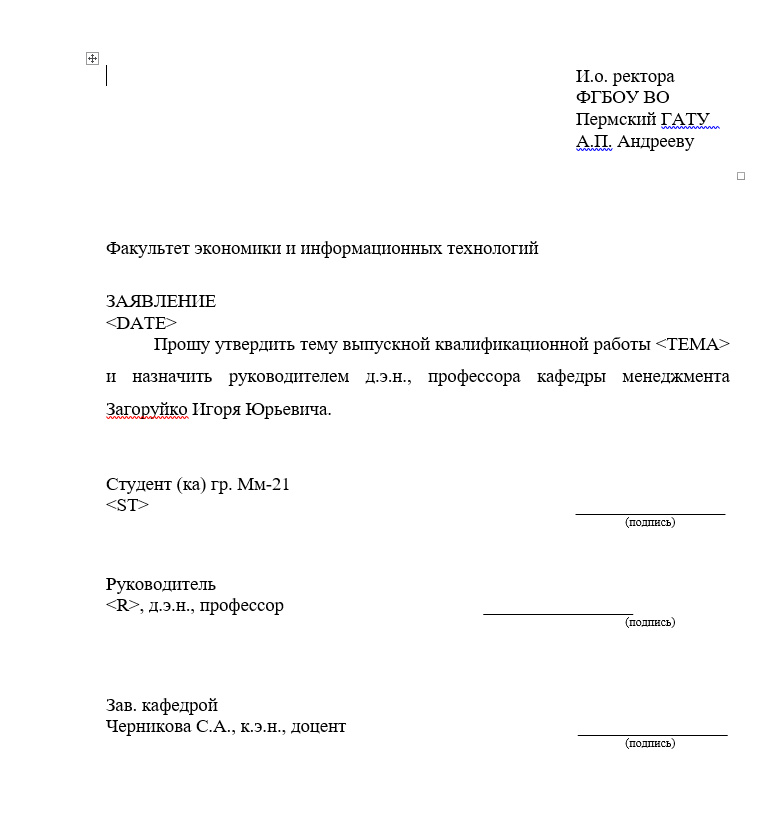
После регистрации или при нажатии кнопки «Авторизация» открывается окно авторизации, в котором нужно ввести свой логин и пароль.



После авторизации пользователя открывается окно заполнения заявления, в которое нужно записать ФИО, группу, дату, и тему ВКР. После нажатия кнопки «Записать в БД» происходит запись или изменение введенной информации в базу данных. После нажатия кнопки «Сохранить в Word» введенные данные записываются в шаблон документа в формате. При нажатии кнопки «Закрыть форму» форма заявления закроется и откроет форму авторизации.



Далее представлен шаблон заявления.



**Регистрация и авторизация**

Регистрация выполняется в окне регистрации, в котором пользователь вводит информацию и нажимает кнопку «Сохранить», затем программа добавляет введенную информацию в базу данных с помощью SQL запроса «INSERT». (Листинг 1)

Авторизация выполняется в окне авторизации, в котором пользователь вводит свои логин и пароль, после нажатия кнопки «Авторизоваться», при условии, что в базе данных есть такая пара логин-пароль, открывается окно заполнения заявления с данными, взятыми из базы данных по id студента. (Листинг 2).

Листинг 1 – Регистрация

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

namespace Practice

{

public partial class FormReg : Form

{

public FormReg()

{

InitializeComponent();

}

private void btnCancel\_Click(object sender, EventArgs e)

{

this.Close();

}

private void btnSave\_Click(object sender, EventArgs e)

{

var conn = DbHelper.GetConn();

string query = "INSERT INTO `users`" +

"(`lastName`,`firstName`,`middleName`,`login`,`password`,`date`, `topic`, `gr`) " +

"VALUES(@ln, @fn, @mn, @log, @pas, @date, @topic, @gr)";

try

{

conn.Open();

MySqlCommand cmd = new MySqlCommand(query, conn);

cmd.Parameters.Add("@ln", MySqlDbType.VarChar).Value = txtName1.Text;

cmd.Parameters.Add("@fn", MySqlDbType.VarChar).Value = txtName2.Text;

cmd.Parameters.Add("@mn", MySqlDbType.VarChar).Value = txtName3.Text;

cmd.Parameters.Add("@log", MySqlDbType.VarChar).Value = txtLogin.Text;

cmd.Parameters.Add("@pas", MySqlDbType.VarChar).Value = GenHesh.CalculateMD5Hash(txtPassword.Text);

cmd.Parameters.Add("@date", MySqlDbType.VarChar).Value = " 2000.10.12";

cmd.Parameters.Add("@topic", MySqlDbType.VarChar).Value = "''";

cmd.Parameters.Add("@gr", MySqlDbType.VarChar).Value = "''";

cmd.ExecuteNonQuery();

MessageBox.Show("Пользователь зарегистрирован");

}

catch (Exception ex)

{

MessageBox.Show("Нет соединения с БД\n\r" + ex.ToString());

}

finally

{

conn.Close();

}

}

}

}

Листинг 2 – Авторизация

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

namespace Practice

{

public partial class FormAutho : Form

{

public FormAutho()

{

InitializeComponent();

}

private void btnCancel\_Click(object sender, EventArgs e)

{

FormStart formStart = new FormStart();

this.Close();

formStart.Show();

}

private void btnAutho\_Click(object sender, EventArgs e)

{

var conn = DbHelper.GetConn();

string query = "SELECT idUser FROM `users` WHERE login = @lg AND password = @pass";

try

{

conn.Open();

MySqlCommand cmd = new MySqlCommand(query, conn);

cmd.Parameters.Add("@lg", MySqlDbType.VarChar).Value = txtLogin.Text;

cmd.Parameters.Add("@pass", MySqlDbType.VarChar).Value = GenHesh.CalculateMD5Hash(txtPassword.Text);

cmd.ExecuteNonQuery();

int id = 0;

id = Convert.ToInt32(cmd.ExecuteScalar());

if (id == 0)

{

MessageBox.Show("Неверный логин/пароль");

return;

}

this.Hide();

Form1 f1 = new Form1();

f1.fio(id);

f1.Show();

f1.FormClosed += this.FormAutho\_FormClosed;

}

catch (Exception ex)

{

MessageBox.Show("Нет соединения с БД\n\r" + ex.ToString());

//Form1 form1 = new Form1(id);

}

finally

{

conn.Close();

}

}

private void FormAutho\_FormClosed(object sender, FormClosedEventArgs e)

{

this.Show();

}

}

}

**Генерация документов**

Генерация документа происходит после нажатия на кнопку «Сохранить в Word» на форме заполнения заявления. Генерация реализована в классе «WordHelper». (Листинг 3)

Листинг 3 – Генерация документов

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Word = Microsoft.Office.Interop.Word;

namespace WordsChanger

{

public class WordHelper

{

private FileInfo fileInfo;

public WordHelper(string fileName)

{

if (File.Exists(fileName))

{

fileInfo = new FileInfo(fileName);

}

else

{

throw new ArgumentException("File not found");

}

}

internal bool Proccess(Dictionary<string, string> items)

{

Word.Application app = null;

try

{

app = new Word.Application();

Object file = fileInfo.FullName;

Object missing = Type.Missing;

app.Documents.Open(file);

foreach (var item in items)

{

Word.Find find = app.Selection.Find;

find.Text = item.Key;

find.Replacement.Text = item.Value;

Object wrap = Word.WdFindWrap.wdFindContinue;

Object replace = Word.WdReplace.wdReplaceAll;

find.Execute(FindText: Type.Missing,

MatchCase: false,

MatchWholeWord: false,

MatchWildcards: false,

MatchSoundsLike: missing,

MatchAllWordForms: false,

Forward: true,

Wrap: wrap,

Format: false,

ReplaceWith: missing,

Replace: replace);

}

Object newFileName = Path.Combine("C:\\Users\\Alexey\\Source\\Repos\\Practice — копия\\WordsChanger");

app.ActiveDocument.SaveAs2(newFileName);

app.ActiveDocument.Close();

return true;

}

catch (Exception ex)

{

//Console.WriteLine(ex.Message);

MessageBox.Show(ex.ToString());

}

finally

{

if (app != null)

{

app.Quit();

}

}

return false;

}

}

}

**Шифрование**

Шифрование пароля происходит с помощью хеширования, которое реализовано в классе «GenHash». (Листинг 4)

Листинг 4

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Word = Microsoft.Office.Interop.Word;

namespace WordsChanger

{

public class WordHelper

{

private FileInfo fileInfo;

public WordHelper(string fileName)

{

if (File.Exists(fileName))

{

fileInfo = new FileInfo(fileName);

}

else

{

throw new ArgumentException("File not found");

}

}

internal bool Proccess(Dictionary<string, string> items)

{

Word.Application app = null;

try

{

app = new Word.Application();

Object file = fileInfo.FullName;

Object missing = Type.Missing;

app.Documents.Open(file);

foreach (var item in items)

{

Word.Find find = app.Selection.Find;

find.Text = item.Key;

find.Replacement.Text = item.Value;

Object wrap = Word.WdFindWrap.wdFindContinue;

Object replace = Word.WdReplace.wdReplaceAll;

find.Execute(FindText: Type.Missing,

MatchCase: false,

MatchWholeWord: false,

MatchWildcards: false,

MatchSoundsLike: missing,

MatchAllWordForms: false,

Forward: true,

Wrap: wrap,

Format: false,

ReplaceWith: missing,

Replace: replace);

}

Object newFileName = Object newFileName = Path.Combine("C:\\Users\\Maxaki\\Desktop\\Practice — копия\\WordsChanger");

app.ActiveDocument.SaveAs2(newFileName);

app.ActiveDocument.Close();

return true;

}

catch (Exception ex)

{

//Console.WriteLine(ex.Message);

MessageBox.Show(ex.ToString());

}

finally

{

if (app != null)

{

app.Quit();

}

}

return false;

}

}

}

# Заключение

Таким образом, в результате работы на проектно-технологической практике было создано приложение с сетевой базой данных для автоматизации учета тем ВКР. Оно имеет такие возможности как о добавление студентов в базу данных, авторизация по логину и паролю, хэширование паролей, вывод данных в шаблон документа .docx.

Перспективами развития данного приложения является научных руководителей в базу данных, а также возможность интегрирования данного приложения на Портал ПГАТУ.

# Список источников

1. ГОСТ 19.401-78 Текст программы. требования к содержанию и оформлению.
2. ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы.
3. ГОСТ 19.701-90 Схемы алгоритмов, программ данных и систем.
4. Алгоритм хеширования MD5. [Электронный ресурс] – Электрон. текстовые дан. Режим доступа: https://ru.wikipedia.org/wiki/MD5
5. МЕТОДИЧЕСКОЕ ПОСОБИЕ по выполнению курсового проекта по дисциплине Объектно-ориентированное программирование. Режим доступа: https://pcoding.ru/pdf/CourseProject.pdf
6. Git Repositories Documentation [Электронный ресурс] – Электрон. текстовые дан. – Microsoft, Режим доступа: https://docs.microsoft.com/ru-ru/azure/devops/repos/git, свободный.
7. MSDN. Техническая документация. MD5 Класс. [Электронный ресурс] Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/api/system.security.cryptography.md5?view=netframework-4.8
8. MSDN. Техническая документация. HashAlgorithm.ComputeHash Метод. [Электронный ресурс] Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/api/system.security.cryptography.hashalgorithm.computehash?view=netframework-4.8
9. Паничев С. Язык программирования С#: история, специфика, место на рынке. [Электронный ресурс] Режим доступа: https://gb.ru/posts/yazyk-programmirovaniya-c-sharp-istoriya-specifika-mesto-na-rynke
10. itGap.ru. Python vs Node.js. Какой язык программирования выбрать? [Электронный ресурс] Режим доступа: https://zen.yandex.ru/media/itgap/python-vs-nodejs-kakoi-iazyk-programmirovaniia-vybrat-5da882a1ba281e00b00e6fed

# ПРИЛОЖЕНИЕ А

**Техническое задание**

ФГБОУ ВО Пермский ГАТУ

наименование организации – разработчика ТЗ на АС

УТВЕРЖДАЮ

Руководитель Беляков Андрей Юрьевич

(должность, наименование предприятия – заказчика АС)

Личная подпись Расшифровка подписи

(печать)

Дата

УТВЕРЖДАЮ

Руководитель Дегтярева Анастасия Сергеевна

(должность, наименование предприятия – разработчик АС)

Личная подпись Расшифровка подписи

(печать)

Дата

Автоматизированные процессы записи тем ВКР

наименование вида АС

Пермский Государственный аграрно-технологический университет

наименование объекта автоматизации

Автоматизация записи тем ВКР

сокращённое наименование АС

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

На 3 листах

Действует с 10.07.2021

2021

1. Общие сведения

1.1 Наименование системы

Полное наименование разрабатываемой системы – «Автоматизированные процессы записи тем ВКР».

Краткое наименование – «Автоматизация записи тем ВКР».

1.2 Наименование заказчика и исполнителя

Организация: ФГБОУ ВО Пермский ГАТУ.

Адрес: ул. Петропавловская, 23.

Телефон: +7 (342) 217-90-66;

Исполнитель: Дегтярева Анастасия Сергеевна.

1.3 Плановые сроки начала и окончания работ

Дата начала работ: 05.07.2021.

Дата окончания работ: 16.07.2021.

2. Назначение и цели создания системы

К целям создания модуля «Автоматизированные процессы записи тем ВКР» можно отнести:

-Улучшение оперативного взаимодействия и интеграция модулей;

-Автоматизировать процесс записи тем ВКР.

Достижение целей приведёт к следующим положительным результатам:

-Исчезнет необходимость заполнять таблицы с темами ВКР «вручную» лаборантам и преподавателям.

-Руководителям ВКР будет удобно отслеживать изменение тем ВКР, а также даты изменения тем.

3. Характеристика объекта автоматизации

Объектом автоматизации является подсистема организационной ком-муникации корпоративного портала ФГБОУ ВО Пермский ГАТУ.

4. Требования к системе

Общие требования к модулю «Автоматизированные процессы записи тем ВКР» являются:

-Надёжность и работоспособность;

-Интуитивно понятный интерфейс;

-Лицензионная чистота – применение средств в рамках общего лицензионного соглашения касательно корпоративного портала;

-Соблюдение информационной безопасности и разграничение прав доступа к данным.

4.1 Требования к способам и средствам связи для информационного об-мена между компонентами

Для обеспечения информационного обмена компоненты подсистемы должны взаимодействовать с объединённой информационной базой данных. Благодаря хранению данных в различных схемах, таких как мероприятия или поручения модуль «Автоматизированные процессы записи тем ВКР» может объединить эти данные представив их как единый информационный поток.

4.1.2 Требования к надёжности технических средств и программного обеспечения

Надёжность по отношению к техническим средствам должна обеспечи-ваться использованием в системе средств повышенной отказоустойчивости и их резервированием, а также дублированием носителей информационных банков данных.

Надёжность программного комплекса обеспечивается использованием сертифицированных операционных систем, общесистемных программных средств и инструментальных программных систем, используемых при разра-ботке программного обеспечения. Само программное обеспечение должно обеспечивать защиту от некорректных действий пользователей и ошибочных исходных данных.

4.1.4 Требования к безопасности

Разрабатываемый информационный модуль «Автоматизированные процессы записи тем ВКР» должен обеспечивать безопасный доступ к данным, предотвращая несанкционированный доступ или модифицирование данных. Модуль аутентификации дол-жен обеспечивать защищённый доступ ко всему программному интерфейсу приложения, за исключением статичной формы авторизации в системе предоставляя возможность пройти аутентификацию в корпоративном портале при помощи логина и пароля.

Также при разработке модуля необходимо соблюдать разграничение прав на публикацию информации отправителю должны быть доступны толь-ко те адресаты, которые относятся к его зоне ответственности.

4.1.5 Требования по эргономике и технической эстетике

Модуль должен иметь удобный и интуитивно понятный графический пользовательский интерфейс. Диалоговый интерфейс должен соблюдать кон-текст подсистемы организационной коммуникации университета и управле-ния в целом, тем самым действия конечного пользователя должны быть ясны и знакомы.

Пользовательский интерфейс модуля также должен аккомпанировать цветовой гамме и общему стилю корпоративного портала.

4.1.6 Требования к программному обеспечению

При проектировании информационного модуля «Автоматизированные процессы записи тем ВКР» необходимо эффективно использовать используемое на данный момент в проекте корпоративного портала применяется в качестве серверного окружения используется программная платформа phpMyAdmin, а для хранения данных применяется СУБД MySQL.

4.1.7 Требования к техническому обеспечению

Техническое обеспечение системы должно максимально и наиболее эффективно использовать существующее в отделе автоматизации оборудова-ние:

-Процессор – 2х Intel Xeon 3.7 ГГц;

-Оперативная память – 32 ГБ;

-Дисковая система – 2 х 1ТБ;

-Сетевой адаптер – 1 Гб/с.

5. Порядок контроля и приёмки системы

Приёмо-сдаточные испытания системы проводятся с привлечением сотрудников отдела автоматизации. По результатам опытной эксплуатации оформляется акт о приёме работ. Акт содержит заключение о соответствии системы техническому заданию.

5.1 Требования к составу и содержанию работ подготовки объекта автоматизации к вводу системы в действие

При подготовке к вводу в эксплуатацию модуля «Автоматизированные процессы записи тем ВКР» отдел управления информатизации должен обеспечить выполнение следующих ра-бот:

-Определить подразделение и ответственных должностных лиц для внедрения информационного модуля;

-Обеспечить пользователей руководством, которое поможет быстрее освоить внедрённый модуль;

-Провести опытную эксплуатацию модуля «Автоматизированные процессы записи тем ВКР».

# ПРИЛОЖЕНИЕ Б

**Листинг кода формы авторизации – FormAutho.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

namespace Practice

{

public partial class FormAutho : Form

{

public FormAutho()

{

InitializeComponent();

}

private void btnCancel\_Click(object sender, EventArgs e)

{

FormStart formStart = new FormStart();

this.Close();

formStart.Show();

}

private void btnAutho\_Click(object sender, EventArgs e)

{

var conn = DbHelper.GetConn();

string query = "SELECT idUser FROM `users` WHERE login = @lg AND password = @pass";

try

{

conn.Open();

MySqlCommand cmd = new MySqlCommand(query, conn);

cmd.Parameters.Add("@lg", MySqlDbType.VarChar).Value = txtLogin.Text;

cmd.Parameters.Add("@pass", MySqlDbType.VarChar).Value = GenHesh.CalculateMD5Hash(txtPassword.Text);

cmd.ExecuteNonQuery();

int id = 0;

id = Convert.ToInt32(cmd.ExecuteScalar());

if (id == 0)

{

MessageBox.Show("Неверный логин/пароль");

return;

}

this.Hide();

Form1 f1 = new Form1();

f1.fio(id);

f1.Show();

f1.FormClosed += this.FormAutho\_FormClosed;

}

catch (Exception ex)

{

MessageBox.Show("Нет соединения с БД\n\r" + ex.ToString());

//Form1 form1 = new Form1(id);

}

finally

{

conn.Close();

}

}

private void FormAutho\_FormClosed(object sender, FormClosedEventArgs e)

{

this.Show();

}

}

}

**Листинг кода формы регистрации – FormReg.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

namespace Practice

{

public partial class FormReg : Form

{

public FormReg()

{

InitializeComponent();

}

private void btnCancel\_Click(object sender, EventArgs e)

{

this.Close();

}

private void btnSave\_Click(object sender, EventArgs e)

{

var conn = DbHelper.GetConn();

string query = "INSERT INTO `users`" +

"(`lastName`,`firstName`,`middleName`,`login`,`password`,`date`, `topic`, `gr`) " +

"VALUES(@ln, @fn, @mn, @log, @pas, @date, @topic, @gr)";

try

{

conn.Open();

MySqlCommand cmd = new MySqlCommand(query, conn);

cmd.Parameters.Add("@ln", MySqlDbType.VarChar).Value = txtName1.Text;

cmd.Parameters.Add("@fn", MySqlDbType.VarChar).Value = txtName2.Text;

cmd.Parameters.Add("@mn", MySqlDbType.VarChar).Value = txtName3.Text;

cmd.Parameters.Add("@log", MySqlDbType.VarChar).Value = txtLogin.Text;

cmd.Parameters.Add("@pas", MySqlDbType.VarChar).Value = GenHesh.CalculateMD5Hash(txtPassword.Text);

cmd.Parameters.Add("@date", MySqlDbType.VarChar).Value = " 2000.10.12";

cmd.Parameters.Add("@topic", MySqlDbType.VarChar).Value = "''";

cmd.Parameters.Add("@gr", MySqlDbType.VarChar).Value = "''";

cmd.ExecuteNonQuery();

MessageBox.Show("Пользователь зарегистрирован");

}

catch (Exception ex)

{

MessageBox.Show("Нет соединения с БД\n\r" + ex.ToString());

}

finally

{

conn.Close();

}

}

}

}

**Листинг кода стартовой формы – FormStart.cs**

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Practice

{

public partial class FormStart : Form

{

public FormStart()

{

InitializeComponent();

}

private void btnReg\_Click(object sender, EventArgs e)

{

this.Hide();

FormReg formReg = new FormReg();

formReg.Show();

formReg.FormClosed += this.formClosed;

}

private void formClosed(object sender, FormClosedEventArgs e)

{

this.Show();

}

private void btnAuto\_Click(object sender, EventArgs e)

{

this.Hide();

FormAutho formAutho= new FormAutho();

formAutho.Show();

formAutho.FormClosed += this.formClosed;

}

private void FormStart\_Shown(object sender, EventArgs e)

{

Task.Delay(0).ContinueWith(CheckConnection);

}

private void CheckConnection(Task obj)

{

var conn = DbHelper.GetConn();

try

{

conn.Open();

this.status1.Text = "Соединение установлено";

}

catch (Exception ex)

{

this.status1.Text = "Нет соединения";

MessageBox.Show("Нет соединения c БД\n\r" + ex.ToString());

}

finally

{

conn.Close();

}

}

private void tableLayoutPanel1\_Paint(object sender, PaintEventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

**Листинг кода формы записи темы – Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Threading;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

using Word = Microsoft.Office.Interop.Word;

using System.IO;

using WordsChanger;

namespace Practice

{

public partial class Form1 : Form

{

string fio\_student;

int id;

public Form1()

{

InitializeComponent();

}

public void fio(int id)

{

var conn = DbHelper.GetConn();

string query = "SELECT lastName, firstName, middleName FROM `users` WHERE idUser = @id;";

this.id = id;

try

{

conn.Open();

MySqlCommand cmd = new MySqlCommand(query, conn);

cmd.Parameters.Add("@id", MySqlDbType.Int32).Value = id;

MySqlDataReader r = cmd.ExecuteReader();

fio\_student = "";

while (r.Read())

fio\_student += String.Format("{0} {1} {2}", r[0], r[1], r[2]);

textBox1.Text = fio\_student;

}

catch(Exception ex)

{

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Hide();

WordsChanger.Form1 f1 = new WordsChanger.Form1();

f1.parametres(textBox4.Text, textBox1.Text);

f1.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

var conn = DbHelper.GetConn();

string query = "UPDATE `users` SET topic = @tp, gr = @grp where idUser = @id;";

try

{

conn.Open();

MySqlCommand cmd = new MySqlCommand(query, conn);

cmd.Parameters.Add("@tp", MySqlDbType.VarChar).Value = textBox4.Text;

cmd.Parameters.Add("@grp", MySqlDbType.VarChar).Value = textBox2.Text;

cmd.Parameters.Add("@id", MySqlDbType.VarChar).Value = id;

cmd.ExecuteNonQuery();

MessageBox.Show("Пользователь зарегистрирован");

}

catch (Exception ex)

{

MessageBox.Show("Нет соединения с БД\n\r" + ex.ToString());

}

finally

{

conn.Close();

}

}

}

}

**Листинг кода класса записи в документ – WordHelper.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Word = Microsoft.Office.Interop.Word;

namespace WordsChanger

{

public class WordHelper

{

private FileInfo fileInfo;

public WordHelper(string fileName)

{

if (File.Exists(fileName))

{

fileInfo = new FileInfo(fileName);

}

else

{

throw new ArgumentException("File not found");

}

}

internal bool Proccess(Dictionary<string, string> items)

{

Word.Application app = null;

try

{

app = new Word.Application();

Object file = fileInfo.FullName;

Object missing = Type.Missing;

app.Documents.Open(file);

foreach (var item in items)

{

Word.Find find = app.Selection.Find;

find.Text = item.Key;

find.Replacement.Text = item.Value;

Object wrap = Word.WdFindWrap.wdFindContinue;

Object replace = Word.WdReplace.wdReplaceAll;

find.Execute(FindText: Type.Missing,

MatchCase: false,

MatchWholeWord: false,

MatchWildcards: false,

MatchSoundsLike: missing,

MatchAllWordForms: false,

Forward: true,

Wrap: wrap,

Format: false,

ReplaceWith: missing,

Replace: replace);

}

/// Object newFileName = Path.Combine(\_fileInfo.DirectoryName, DateTime.Now.ToString("yyyyMMdd") + \_fileInfo.Name);

//Object newFileName = Path.Combine("C:\\Users\\max\\source\\repos\\app\\app\\bin\\Debug\\doc1.doc");

Object newFileName = Path.Combine("C:\\Users\\Maxaki\\Desktop\\Practice — копия\\WordsChanger");

app.ActiveDocument.SaveAs2(newFileName);

app.ActiveDocument.Close();

return true;

}

catch (Exception ex)

{

//Console.WriteLine(ex.Message);

MessageBox.Show(ex.ToString());

}

finally

{

if (app != null)

{

app.Quit();

}

}

return false;

}

}

**Листинг кода класса хеширования пароля – GenHash.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Practice

{

class GenHesh

{

public static string CalculateMD5Hash(string input)

{

using (var md5 = System.Security.Cryptography.MD5.Create())

{

byte[] inputBytes = Encoding.UTF8.GetBytes(input);

byte[] hash = md5.ComputeHash(inputBytes);

var sb = new StringBuilder(16 \* 2);

for (int i = 0; i < hash.Length; i++)

{

sb.Append(hash[i].ToString("X2"));

}

return sb.ToString();

}

}

}

}