МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«Новосибирский национальный исследовательский государственный университет»

(Новосибирский государственный университет, НГУ)

Структурное подразделение Новосибирского государственного университета –

Высший колледж информатики Университета (ВКИ НГУ)

КАФЕДРА ИНФОРМАТИКИ

ОТЧЕТ КУРСОВОГО ПРОЕКТА

ПМ.01. Разработка программных модулей

Приложение «Win32»

|  |  |
| --- | --- |
| Преподаватель  доцент ВКИ НГУ | Голкова Н.В.  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2022 г. |
| Студент 3 курса  гр. 107сб2 | Косинова А.И  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_2022 г. |

Новосибирск

2022

**СОДЕРЖАНИЕ**

[ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, УСЛОВНЫХ ОБОЗНАЧЕНИЙ И ТЕРМИНОВ 3](#_Toc122989654)

[ВВЕДЕНИЕ 4](#_Toc122989655)

[1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc122989656)

[2 ПОСТАНОВКА ЗАДАЧИ 7](#_Toc122989657)

[3 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К ПРОГРАММНОМУ ПРОДУКТУ 9](#_Toc122989658)

[4 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К ПРОГРАММНОМУ ПРОДУКТУ 10](#_Toc122989659)

[4.1 Требования к программному обеспечению 10](#_Toc122989660)

[4.2 Требования к аппаратному обеспечению 10](#_Toc122989661)

[4.3 Требования к надёжности 11](#_Toc122989662)

[5 ХАРАКТЕРИСТИКА ВЫБРАННЫХ ПРОГРАММНЫХ СРЕД И СРЕДСТВ 13](#_Toc122989663)

[6 АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ 16](#_Toc122989664)

[6.1 Алгоритмы реализации программного средства 16](#_Toc122989665)

[6.3 Схема функционирования программного средства 17](#_Toc122989666)

[7 ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ 19](#_Toc122989667)

[8 ТЕСТИРОВАНИЕ И ОТЛАДКА 28](#_Toc122989668)

[9 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 30](#_Toc122989669)

[ЗАКЛЮЧЕНИЕ 33](#_Toc122989670)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 34](#_Toc122989671)

[ПРИЛОЖЕНИЕ A 35](#_Toc122989672)

[ПРИЛОЖЕНИЕ В 55](#_Toc122989673)

# 

# ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, УСЛОВНЫХ ОБОЗНАЧЕНИЙ И ТЕРМИНОВ

WPF – Windows Presentation Foundation;

MS SQL Server – Microsoft SQL Server Management;

VS – Visual Studio;

# ****ВВЕДЕНИЕ****

Актуальность

На данный момент существует тысячи приложений для операционной системы Windows под самые разные цели.

Приложение для вывода информации о системе компьютера необходимо для того, чтобы знать возможности машины, ознакомиться с архитектурой системы. Такое приложение нужно каждому, кто хоть раз задумывался сколько ядер в процессоре компьютера или же как называется устройство.

С развитием информационных технологий и телекоммуникаций жизнь становится все более мобильной и информативной, новые технологии прочно входят в различные отрасли хозяйствования, сферы жизни и несут новые нормы в них. В связи с реформирование экономики, с взятием курса на инновационное развитие экономики, всё чаще и чаще в повседневной работе в большинстве предприятий и организаций используют различные средства информационно вычислительной техники и соответственно программного обеспечения. Но необходимо заметить, что спонтанное, не спланированное развитие в любой деятельности малоэффективно. Опираясь на данное утверждение и выбрана данная тематика курсового проекта.

Приложение, где будут показаны основные параметры системы это хорошее средство популярно среди IT-специалистов, компьютерных техников и ремонтников, [геймеров](https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%B9%D0%BC%D0%B5%D1%80) и [оверклокеров](https://ru.wikipedia.org/wiki/%D0%9E%D0%B2%D0%B5%D1%80%D0%BA%D0%BB%D0%BE%D0%BA%D0%B5%D1%80), но и обычным пользователям. Такие приложения очень популярны уже долгое время. Например, CPU-Z - небольшая утилита существует уже 20 лет, создана она была французским разработчиком Франком Делатром. Данное приложение с его функционалом также разрабатывается под Android 1.41.

# ****1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ****

Компьютерная программа —

1) комбинация компьютерных [инструкций](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) и [данных](https://ru.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5_(%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0)), позволяющая [аппаратному обеспечению вычислительной системы](https://ru.wikipedia.org/wiki/%D0%90%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) выполнять [вычисления](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5) или функции управления (стандарт [ISO](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B6%D0%B4%D1%83%D0%BD%D0%B0%D1%80%D0%BE%D0%B4%D0%BD%D0%B0%D1%8F_%D0%BE%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D0%BF%D0%BE_%D1%81%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8)/[IEC](https://ru.wikipedia.org/wiki/IEC)/[IEEE](https://ru.wikipedia.org/wiki/IEEE) 24765:2010)[[1]](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0#cite_note-ISO24765-1);

2) [синтаксическая](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) единица, которая соответствует правилам определённого [языка программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), состоящая из определений и [операторов или инструкций](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), необходимых для определённой функции, задачи или решения проблемы (стандарт ISO/IEC 2382-1:1993)[[2]](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0#cite_note-ISO2382-2).

Первое определение соответствует понятию «[исполняемая программа](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BF%D0%BE%D0%BB%D0%BD%D1%8F%D0%B5%D0%BC%D1%8B%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB)», второе относится к понятию «[исходный текст](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4)».

Другие определения из нормативных документов:

данные, предназначенные для управления конкретными компонентами системы [обработки данных](https://ru.wikipedia.org/w/index.php?title=%D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85&action=edit&redlink=1) в целях реализации определённого [алгоритма](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) ([ГОСТ](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B6%D0%B3%D0%BE%D1%81%D1%83%D0%B4%D0%B0%D1%80%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D1%81%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82) 19781—90).

представленная в объективной форме совокупность данных и команд, предназначенных для функционирования [ЭВМ](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%BE-%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%B0) и других компьютерных устройств с целью получения определённого результата, включая подготовительные материалы, полученные в ходе разработки программы для ЭВМ, и порождаемые ею аудиовизуальные отображения ([Гражданский кодекс Российской Федерации](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D0%B6%D0%B4%D0%B0%D0%BD%D1%81%D0%BA%D0%B8%D0%B9_%D0%BA%D0%BE%D0%B4%D0%B5%D0%BA%D1%81_%D0%A0%D0%BE%D1%81%D1%81%D0%B8%D0%B9%D1%81%D0%BA%D0%BE%D0%B9_%D0%A4%D0%B5%D0%B4%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8)).

Компьютерные программы как объект [авторского права](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D1%80%D1%81%D0%BA%D0%BE%D0%B5_%D0%BF%D1%80%D0%B0%D0%B2%D0%BE) и других прав [интеллектуальной собственности](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%BB%D0%BB%D0%B5%D0%BA%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%BE%D0%B1%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D1%8C) относятся к категории [нематериальных активов](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%BC%D0%B0%D1%82%D0%B5%D1%80%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D1%8B).

Самый простой и точный вариант ответа: «Программирование – это акт инструктирования компьютеров для выполнения задач». Еще его называют разработкой или **кодингом**.

Итак, что такое **компьютерная программа**? ПО представляет собой последовательность инструкций, выполняемых ПК. Компьютер же – это любое устройство, способное обрабатывать код. Сюда относятся стационарные ПК, ноутбуки, планшеты, банкоматы, Raspberry Pi, серверы etc.

SQL Server является одной из наиболее популярных систем управления базами данных (СУБД) в мире. Данная СУБД подходит для самых различных проектов: от небольших приложений до больших высоконагруженных проектов.

SQL Server был создан компанией Microsoft. Первая версия вышла в 1987 году. А текущей версией является версия 2022, которая вышла в ноябре 2022 году и которая будет использоваться в текущем руководстве.

SQL Server долгое время был исключительно системой управления базами данных для Windows, однако начиная с версии 16 эта система доступна и на Linux.

SQL Server характеризуется такими особенностями как:

Таблица 1 – характеристики MS SQL Server

|  |
| --- |
| Производительность. SQL Server работает очень быстро. |
| Надежность и безопасность. SQL Server предоставляет шифрование данных. |
| Простота. С данной СУБД относительно легко работать и вести администрирование. |

# ****2 ПОСТАНОВКА ЗАДАЧИ****

Целью проекта является:

1. Изучение инструментов разработки для операционной системы Windows;
2. Изучение построения архитектуры проекта;
3. Работа с ядром СУБД MS SQL Server;
4. Изучение языка XAML;
5. Изучение структуры WPF;
6. Изучение библиотеки Microsift.Data.Sqlclient;
7. Изучение библиотеки System.Management.

Таблица 2 – Постановка задачи

|  |
| --- |
| Задача |
| Реализация WPF-дизайна для страниц и окон приложения. |
| Реализация страницы для входа в приложение через базу данных. |
| Реализация страницы авторизации для входа в приложение через базу данных. |
| Реализация окна панели администратора, где присутствует возможность просмотреть все таблицы базы данных. |
| Реализация основного окна приложения с работающим функционалом. |
| Реализация записи данных в базу данных. |
| Реализация вывода данных из базы данных. |
| Реализация работающих кнопок в приложении. |
| Реализация проверок для пользователя при регистрации. |
| Реализация входа по роли «Администратор» и «Юзер». |
| Реализация подключения проекта к базе данных SQL Server. |
| Реализация «резиновой» верстки проекта при помощи WPF |
| Использование User Control в приложении |
| Реализовать вывод классов «Процессор», «Сеть», «Материнская плата», «Клавиатура», «Память», «Монитор». |
| Кнопка «Очистить лист» |
| Сделать иконку приложения. |
| Собрать проект в .exe файл |

# ****3 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К ПРОГРАММНОМУ ПРОДУКТУ****

Функциональные требования объясняют, что должно быть сделано. Они идентифицируют задачи или действия, которые должны быть выполнены. Функциональные требования определяют действия, которые система должна быть способной выполнить, связь входа/выхода в поведении системы. Диаграмма функциональных требований представлена на рисунке 3.

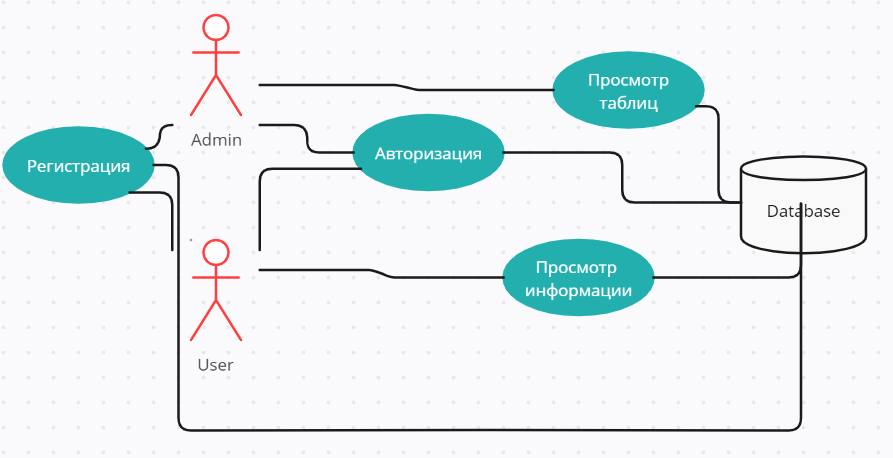


Рисунок 1 – Диаграмма претендентов

Таблица 2 – Use cases

|  |  |
| --- | --- |
| Use cases | Запустить приложение |
| Нажать на кнопку «Sign up» |
| Ввести логин, пароль и повторить пароль |
| Нажать на кнопку «Back» |
| Ввести логин и пароль |
| Нажать на кнопку «Proccesor» |
| Очистить список |

# ****4 НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К ПРОГРАММНОМУ ПРОДУКТУ****

## 4.1 Требования к программному обеспечению

Требования к программному обеспечению — совокупность запросов/утверждений относительно атрибутов, свойств или качеств программной системы, подлежащей реализации. Создаются в процессе проработки (анализа и синтеза) задания на разработку/модернизацию программного обеспечения (ПО).

Таблица 3 – Требование к программному обеспечению

|  |  |
| --- | --- |
| Требование | Средство |
| Операционная система | Windows 8.1 и выше |
| Драйверы | Не требуют дополнительной установки |
| Загрузчик | Application Manifest (.application) |
| Подключение к серверу | Microsoft SQL Server Tool |
| Права администратора | Не требуются |

## 4.2 Требования к аппаратному обеспечению

Аппаратные требования, предъявляемые к конфигурации аппаратного оборудования, зависят от операционной системы (ОС).

Таблица 4 – Требование к аппаратному обеспечению

|  |  |
| --- | --- |
| Требование | Средство |
| Количество серверов | 1 |
| Количество вычислительных потоков процессоров (шт.) | 2 |
| Тактовая частота процессора (Ггц) | 2,5 |
| Оперативная память (Гб) | 0,03 |
| Свободное дисковое пространство (Гб) | 0,04 |
| Тип дисковых накопителей | SATA |

## 4.3 Требования к надёжности

ГОСТ 27.003-90 трактует требования по надежности как совокупность количественных и (или) качественных требований к безотказности, долговечности, ремонтопригодности, сохраняемости, выполнение которых обеспечивает эксплуатацию изделий с заданными показателями эффективности, безопасности, экологичности, живучести и других ...

Таблица 5 – Требования к надёжности

|  |  |
| --- | --- |
| Безотказность | Приложение способно без отказа работать на протяжении долгого времени. |
| Долговечность | Проект способен работать без правок от 12 месяцев. |
| Ремонтопригодность | Приложение открыто для предупреждения и обнаружения причин возникновения отказов, повреждений и поддержания и восстановления работоспособного состояния |
| Сохраняемость | Приложение выполняет необходимые функции и параметры. |

# ****5 ХАРАКТЕРИСТИКА ВЫБРАННЫХ ПРОГРАММНЫХ СРЕД И СРЕДСТВ****

Для разработки приложения были выбраны программные средства:

**Среда разработки VS**

Visual Studio (интегрированная среда разработки Visual Studio) — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений.

VS было выбрано потому как она позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов.

**Объектно-ориентированный язык программирования C#**

C# — это язык программирования от компании Microsoft. Изначально его создавали для проектов под Windows, но теперь это по-настоящему универсальный язык: на нём пишут игры, десктопные приложения, веб-сервисы, нейросети и даже графику для метавселенных.

Язык был выбран для проекта поскольку C# — это язык от Microsoft, на нём удобно писать настольные приложения для Windows. Для этого разработчики придумали специальную платформу — называется [.NET Framework](https://skillbox.ru/media/code/kak_rabotaet_net_i_zachem_on_nuzhen/). Она умеет переводить код из понятного человеку в понятный процессору — то есть **компилировать**.

Преимущество платформы в том, что она может исполнить любую C#-команду на любом процессоре — а на Windows работает не меньше тысячи разных моделей. Если бы не .NET, пришлось бы компилировать код под каждое железо отдельно. А ещё .NET поддерживает много плагинов, библиотек и шаблонов для наглядной разработки интерфейсов — это фактически целый мир и экосистема для программиста на Windows.

Windows Presentation Foundation

Windows Presentation Foundation (WPF) — это библиотека для создания пользовательских интерфейсов для интеллектуальных клиентских приложений. В этом разделе будет предложена обширная информация о важнейших концепциях WPF.

Платформа WPF предоставляет широкий и гибкий набор графических функций, который обладает перечисленными ниже преимуществами, поэтому она была выбрана для проекта.

1. Независимость графики от разрешения и устройства.
2. Повышение точности.
3. Расширенная поддержка графики и анимации.
4. Аппаратное ускорение.

**Microsoft SQL Server**

SQL, или Structured Query Language, — это язык структурированных запросов, использующийся для работы с базой данных: извлечения, обновления, добавления и удаления информации из нее. То есть, SQL — язык запросов для “общения” с данными.

СУБД SQL Server используются для создания, размещения, хранения и управления реляционными (табличными) базами данных на специальных серверах или в облаке. Они работают через настольные приложения и web-сайты.

Основные преимущества Microsoft SQL Server 2014:

1. Полная Web ориентированность.
2. Масштабируемость и надежность.
3. Скорость создания решений.
4. Возможность обработки вычислений в оперативной памяти (in-memory OLTP).
5. Возможность взаимодействия с публичным облаком Windows Azure.

**Библиотека System.Data.Sqlclient**

SqlConnectionStringBuilder Класс (System. Data. SqlClient) Удобное средство, с помощью которого можно создавать используемые классом SqlConnection строки соединений и управлять их содержимым.

**Библиотека System.Management**

Средства доступа к обширному набору сведений и событий управления, относящихся к системе, устройствам и приложениям, поддерживающим инфраструктуру WMI (Windows Management Instrumentation — инструментарий управления Windows).

Приложения и службы могут запрашивать важные сведения об управлении (например, об объеме свободного места на диске, текущем уровне загрузки процессора, о том, к какой базе данных подключено конкретное приложение и т. п.) с помощью классов, производных от [ManagementObjectSearcher](https://learn.microsoft.com/ru-ru/dotnet/api/system.management.managementobjectsearcher?view=netframework-4.8) и [ManagementQuery](https://learn.microsoft.com/ru-ru/dotnet/api/system.management.managementquery?view=netframework-4.8), а также осуществлять подписку на ряд управляющих событий с помощью класса [ManagementEventWatcher](https://learn.microsoft.com/ru-ru/dotnet/api/system.management.managementeventwatcher?view=netframework-4.8).

Доступные данные могут быть получены в распределенной среде как от управляемых, так и от неуправляемых компонентов.

# ****6**** АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ

## 6.1 Алгоритмы реализации программного средства

Алгоритм реализации представлен на рисунке 2.

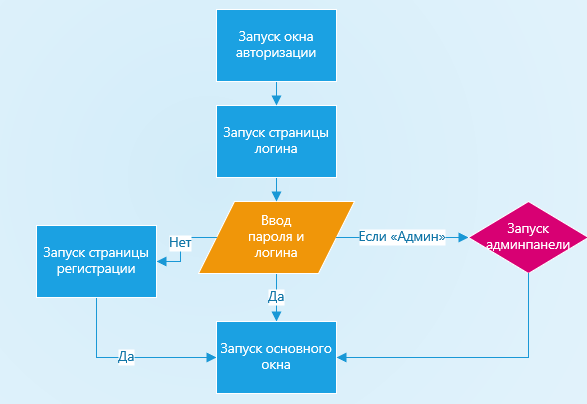


Рисунок 2 – Блок-схема алгоритма реализации

Сначала запускается окно авторизации, внутри которого помещены страницы для авторизации и регистрации. Чтобы зарегистрироваться нужно нажать на кнопку «Sign up» и откроется страница регистрации. При вводе пароля и логина данные из Textbox и Passwordbox сравниваются с данными в базе данных.

При успешном сравнивании открывается окно основного приложения, если роль указана «Пользователь», а если роль «Администратор», открывается окно Админпанели, из которой можно перейти в окно приложения.

## 6.3 Схема функционирования программного средства

Схема диаграммы классов представлена на рисунке 3.

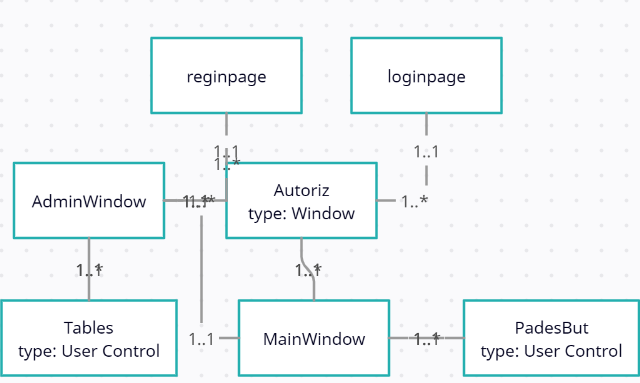
****

Рисунок 3 – Схема диаграммы классов

Таблица 6 – структура проекта «Win32»

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| AdminWindow | Window | Окно, внутрь которого помещены User Control. Нужно для управления таблицами баз данных. |
| Autoriz | Window | Окно с Frame, из которого открываются страницы loginpage и reginpage |
| MainWindow | Window | Окно, в которое помещены элементы User Control |
| PadesBut | User Control | Элемент управления, который используется для отображения информации о системе |
| loginpage | Page | Страница, которая нужна для входа в приложение |
| reginpage | Page | Страница, которая нужна для регистрации в приложении |
| Tables | User Control | Элемент пользовательского управления, нужный для отображения DataGrid |

Для вывода информации о системе используется класс System.Management, который устанавливается как пакет NuGet. Код использования класса находится в Приложении А.

Стартовым окном является Autoriz.

# ****7 ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ****

Входные данные для приложения «Win32»

Таблица 6 – Входные данные для приложения

|  |  |
| --- | --- |
| Данные | Описание |
| Логин | Пользователь вводит логин |
| Пароль | Пользователь вводит пароль |

Выходные данные для приложения

|  |  |
| --- | --- |
| Monitor | uint16   AcceleratorCapabilities[];  string   AdapterCompatibility;  string   AdapterDACType;  uint32   AdapterRAM;  uint16   Availability;  string   CapabilityDescriptions[];  string   Caption;  uint32   ColorTableEntries;  uint32   ConfigManagerErrorCode;  boolean  ConfigManagerUserConfig;  string   CreationClassName;  uint32   CurrentBitsPerPixel;  uint32   CurrentHorizontalResolution;  uint64   CurrentNumberOfColors;  uint32   CurrentNumberOfColumns;  uint32   CurrentNumberOfRows;  uint32   CurrentRefreshRate;  uint16   CurrentScanMode;  uint32   CurrentVerticalResolution;  string   Description;  string   DeviceID;  uint32   DeviceSpecificPens;  uint32   DitherType;  datetime DriverDate;  string   DriverVersion;  boolean  ErrorCleared;  string   ErrorDescription;  uint32   ICMIntent;  uint32   ICMMethod;  string   InfFilename;  string   InfSection;  datetime InstallDate;  string   InstalledDisplayDrivers;  uint32   LastErrorCode;  uint32   MaxMemorySupported;  uint32   MaxNumberControlled;  uint32   MaxRefreshRate;  uint32   MinRefreshRate;  boolean  Monochrome;  string   Name;  uint16   NumberOfColorPlanes;  uint32   NumberOfVideoPages;  string   PNPDeviceID;  uint16   PowerManagementCapabilities[];  boolean  PowerManagementSupported;  uint16   ProtocolSupported;  uint32   ReservedSystemPaletteEntries;  uint32   SpecificationVersion;  string   Status;  uint16   StatusInfo;  string   SystemCreationClassName;  string   SystemName;  uint32   SystemPaletteEntries;  datetime TimeOfLastReset;  uint16   VideoArchitecture;  uint16   VideoMemoryType;  uint16   VideoMode;  string   VideoModeDescription;  string   VideoProcessor; |
| Proccesor | uint16   AddressWidth;  uint16   Architecture;  string   AssetTag;  uint16   Availability;  string   Caption;  uint32   Characteristics;  uint32   ConfigManagerErrorCode;  boolean  ConfigManagerUserConfig;  uint16   CpuStatus;  string   CreationClassName;  uint32   CurrentClockSpeed;  uint16   CurrentVoltage;  uint16   DataWidth;  string   Description;  string   DeviceID;  boolean  ErrorCleared;  string   ErrorDescription;  uint32   ExtClock;  uint16   Family;  datetime InstallDate;  uint32   L2CacheSize;  uint32   L2CacheSpeed;  uint32   L3CacheSize;  uint32   L3CacheSpeed;  uint32   LastErrorCode;  uint16   Level;  uint16   LoadPercentage;  string   Manufacturer;  uint32   MaxClockSpeed;  string   Name;  uint32   NumberOfCores;  uint32   NumberOfEnabledCore;  uint32   NumberOfLogicalProcessors;  string   OtherFamilyDescription;  string   PartNumber;  string   PNPDeviceID;  uint16   PowerManagementCapabilities[];  boolean  PowerManagementSupported;  string   ProcessorId;  uint16   ProcessorType;  uint16   Revision;  string   Role;  boolean  SecondLevelAddressTranslationExtensions;  string   SerialNumber;  string   SocketDesignation;  string   Status;  uint16   StatusInfo;  string   Stepping;  string   SystemCreationClassName;  string   SystemName;  uint32   ThreadCount;  string   UniqueId;  uint16   UpgradeMethod;  string   Version;  boolean  VirtualizationFirmwareEnabled;  boolean  VMMonitorModeExtensions;  uint32   VoltageCaps |
| Keyboard | uint16 Availability;  string Caption;  uint32 ConfigManagerErrorCode;  boolean ConfigManagerUserConfig;  string CreationClassName;  string Description;  string DeviceID;  boolean ErrorCleared;  string ErrorDescription;  datetime InstallDate;  boolean IsLocked;  uint32 LastErrorCode;  string Layout;  string Name;  uint16 NumberOfFunctionKeys;  uint16 Password;  string PNPDeviceID;  uint16 PowerManagementCapabilities[];  boolean PowerManagementSupported;  string Status;  uint16 StatusInfo;  string SystemCreationClassName;  string SystemName; |
| Motherboard | uint16 Availability;  string Caption;  uint32 ConfigManagerErrorCode;  boolean ConfigManagerUserConfig;  string CreationClassName;  string Description;  string DeviceID;  boolean ErrorCleared;  string ErrorDescription;  datetime InstallDate;  uint32 LastErrorCode;  string Name;  string PNPDeviceID;  uint16 PowerManagementCapabilities[];  boolean PowerManagementSupported;  string PrimaryBusType;  string RevisionNumber;  string SecondaryBusType;  string Status;  uint16 StatusInfo;  string SystemCreationClassName;  string SystemName; |
| Memory | uint16 Access;  uint8 AdditionalErrorData[];  uint16 Availability;  uint64 BlockSize;  string Caption;  uint32 ConfigManagerErrorCode;  boolean ConfigManagerUserConfig;  boolean CorrectableError;  string CreationClassName;  string Description;  string DeviceID;  uint64 EndingAddress;  uint16 ErrorAccess;  uint64 ErrorAddress;  boolean ErrorCleared;  uint8 ErrorData[];  uint16 ErrorDataOrder;  string ErrorDescription;  uint16 ErrorGranularity;  uint16 ErrorInfo;  string ErrorMethodology;  uint64 ErrorResolution;  datetime ErrorTime;  uint32 ErrorTransferSize;  datetime InstallDate;  uint32 LastErrorCode;  string Name;  uint64 NumberOfBlocks;  string OtherErrorDescription;  string PNPDeviceID;  uint16 PowerManagementCapabilities[];  boolean PowerManagementSupported;  string Purpose;  uint64 StartingAddress;  string Status;  uint16 StatusInfo;  string SystemCreationClassName;  boolean SystemLevelAddress;  string SystemName; |
| Network | string   AdapterType;  uint16   AdapterTypeID;  boolean  AutoSense;  uint16   Availability;  string   Caption;  uint32   ConfigManagerErrorCode;  boolean  ConfigManagerUserConfig;  string   CreationClassName;  string   Description;  string   DeviceID;  boolean  ErrorCleared;  string   ErrorDescription;  string   GUID;  uint32   Index;  datetime InstallDate;  boolean  Installed;  uint32   InterfaceIndex;  uint32   LastErrorCode;  string   MACAddress;  string   Manufacturer;  uint32   MaxNumberControlled;  uint64   MaxSpeed;  string   Name;  string   NetConnectionID;  uint16   NetConnectionStatus;  boolean  NetEnabled;  string   NetworkAddresses[];  string   PermanentAddress;  boolean  PhysicalAdapter;  string   PNPDeviceID;  uint16   PowerManagementCapabilities[];  boolean  PowerManagementSupported;  string   ProductName;  string   ServiceName;  uint64   Speed;  string   Status;  uint16   StatusInfo;  string   SystemCreationClassName;  string   SystemName;  datetime TimeOfLastReset; |

# ****8 ТЕСТИРОВАНИЕ И ОТЛАДКА****

Для проверки работоспособности игрового приложения было проведено функциональное тестирование и юзабилити-тестирование.

Функциональное тестирование – это тестирование ПО в целях проверки реализуемости функциональных требований, то есть способности ПО в определенных условиях решать задачи, нужные пользователям.

1. Авторизация

|  |  |
| --- | --- |
| Тест | Авторизация |
| Ожидание | При вводе пароля и логина, если он существует в базе данных, высвечивается сообщение «Пользователь авторизовался» и включает окно с основным приложением. Если же в бд у пользователя роль «Администратор», то включается админпанель |
| Реальность | При вводе пароля и логина, если он существует в базе данных, высвечивается сообщение «Пользователь авторизовался» и включает окно с основным приложением. Если же в бд у пользователя роль «Администратор», то включается админпанель |

1. Регистрация

|  |  |
| --- | --- |
| Тест | Регистрация |
| Ожидание | При вводе пароля и логина, они проверяются и записываются в бд |
| Реальность | При вводе пароля и логина, они проверяются и записываются в бд |

1. Вывод информации о системе

|  |  |
| --- | --- |
| Тест | Вывод информации о системе |
| Ожидание | При нажатии кнопки на окне, в листбокс высвечивается нужная информация о системе, а информация оттуда записывается в базу данных |
| Реальность | При нажатии кнопки на окне, в листбокс высвечивается нужная информация о системе, а информация оттуда не записывается в базу данных, все значения NULL. |

1. Вывод таблиц из базы данных в DataGrid

|  |  |
| --- | --- |
| Тест | Вывод таблиц из базы данных |
| Ожидание | В окне админпанели при нажатии кнопки высвечивается содержание из таблицы в DataGrid |
| Реальность | В окне админпанели при нажатии кнопки высвечивается содержание из таблицы в DataGrid |

# ****9**** РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

При входе в приложение, пользователь видит страницу авторизации

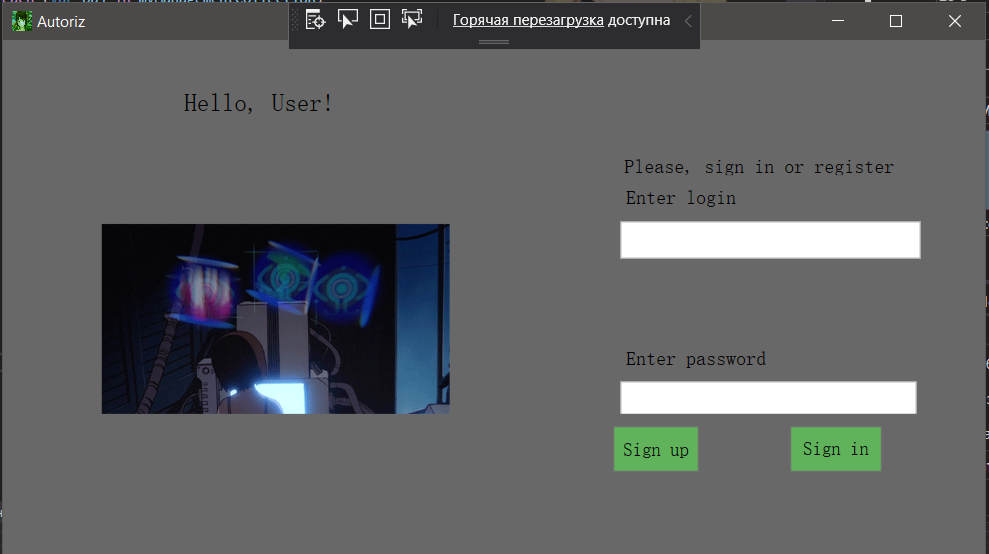


Рисунок 8 – Скриншот страницы авторизации

В странице авторизации представлены два текстбокса, а также две кнопки. Введя пароль и логин, пользователь нажимает кнопку «Sign in», система оповещает его MessageBoхсом, что вход выполнен и переходит на следующее окно.

На рисунке 9 представлена страница регистрации.

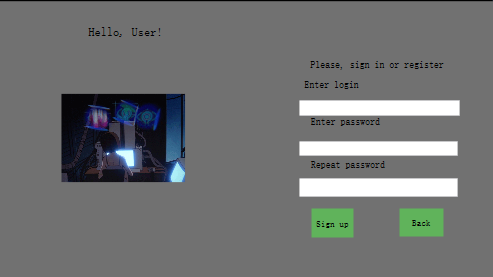


Рисунок 9 – страница регистрации

После регистрации открывается главная страница приложения.

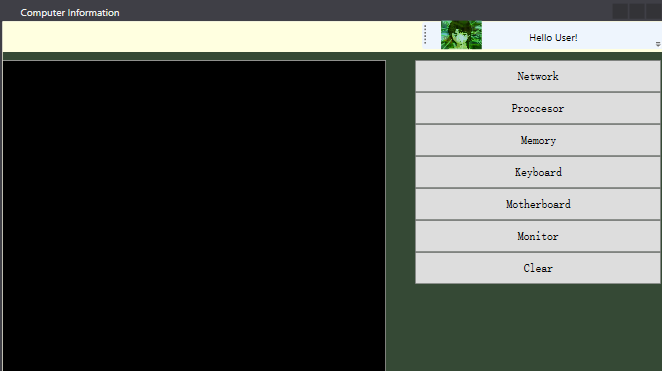
****

Рисунок 9 – Скриншот приложения «Win32»

Админ панель представлено на рисунке 10.



Рисунок 10 – Скриншот панели администратора

Меню паузы имеет стандартные кнопки, как и меню игры.

Окно результаты игры представлены на рисунке 11.

Пользователь должен установить приложение на свой компьютер. Запустить .exe файл «Win32». Пользователь при входе в приложение должен пройти авторизацию или же регистрацию. Если не зарегистрироваться или не ввести пароль, то вход в приложение не произойдет.

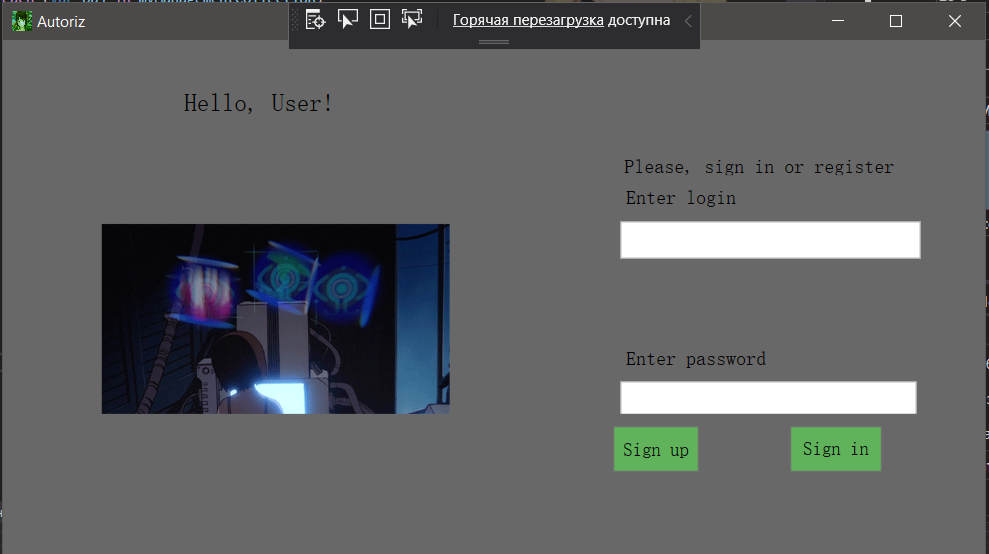


Рисунок 11 – Скриншот страницы авторизации

Попав в окно приложения, пользователь должен нажать на кнопки справа чтобы получить информацию о своей системе.

Чтобы очистить список с информацией нужно нажать на кнопку «Clear».

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы были выполнены такие виды работ как:

|  |  |
| --- | --- |
| Изучены технологии | Работа с WPF |
| Изучение User Control |
| Работа с MC SQL Server |
| Изучение XAML |

Оценка приложения представлены в таблице ниже

|  |  |
| --- | --- |
| Дизайн приложения | 4/10  Мало деталей, нет «резиновой» верстки |
| Функциональность | 5/10  Функция добавления записей о системе в базу данных не работает должным образом |
| Удобство использования | 6/10  Удобно и понятно пользоваться приложением |
| Архитектура приложения | 3/10  Архитектура очень запутана, нет паттерна mvvm, отдельных классов или же отдельного проекта для view. |
| Чистота кода | 4/10  Переменные названы «некрасиво», комментариев почти нет |

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. «Разработка приложений» URL: [<https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/app-development/?view=netframeworkdesktop-4.8>]
2. «Окна в приложениях» URL: [<https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/app-development/windows-in-wpf-applications?view=netframeworkdesktop-4.8> ]

# ПРИЛОЖЕНИЕ A

PadesBut

namespace Win32

{

/// <summary>

/// Логика взаимодействия для PadesBut.xaml

/// </summary>

public partial class PadesBut : UserControl

{

public PadesBut()

{

InitializeComponent();

}

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

{

NetworkInfo();

}

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

{

ProcessorInfo();

}

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

{

MemoryInfo();

}

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

{

KeyboardInfo();

}

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

{

MotherboardInfo();

}

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

{

MonitorInfo();

}

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

{

InfoList.Items.Clear();

}

private void ProcessorInfo()

{

SqlConnection = new SqlConnection("server=LAPTOP-NDOBE5PK; Trusted\_Connection=Yes; DataBase=Win32Project;");

SqlCommand = sqlConnection.CreateCommand();

sqlConnection.Open();

Dictionary<object, object> myPropertyResults = new Dictionary<object, object>();

ManagementClass myManagementClass = new ManagementClass("Win32\_Processor");

ManagementObjectCollection myManagementCollection = myManagementClass.GetInstances();

PropertyDataCollection myProperties = myManagementClass.Properties;

foreach (var obj in myManagementCollection)

{

foreach (var myProperty in myProperties)

{

bool keyExists = myPropertyResults.ContainsKey(myProperty.Name);

if (keyExists == false)

{

myPropertyResults.Add(myProperty.Name,

obj.Properties[myProperty.Name].Value);

}

else

{

myPropertyResults[Name] = obj.Properties[myProperty.Name].Value;

}

}

foreach (var myPropertyResult in myPropertyResults)

{

InfoList.Items.Add(myPropertyResult.Key + ":" + myPropertyResult.Value);

string cmdText = "INSERT \* INTO ProccesorTab VALUES (myPropertyResults)";

SqlCommand cmd = new SqlCommand(cmdText, sqlConnection);

cmd.Parameters.AddWithValue("myProperty.Name", myPropertyResults);

}

}

}

private void NetworkInfo()

{

SqlConnection = new SqlConnection("server=LAPTOP-NDOBE5PK; Trusted\_Connection=Yes; DataBase=Win32Project;");

SqlCommand = sqlConnection.CreateCommand();

sqlConnection.Open();

Dictionary<object, object> myPropertyResults = new Dictionary<object, object>();

ManagementClass myManagementClass = new ManagementClass("Win32\_NetworkAdapter");

ManagementObjectCollection myManagementCollection = myManagementClass.GetInstances();

PropertyDataCollection myProperties = myManagementClass.Properties;

foreach (var obj in myManagementCollection)

{

foreach (var myProperty in myProperties)

{

bool keyExists = myPropertyResults.ContainsKey(myProperty.Name);

if (keyExists == false)

{

myPropertyResults.Add(myProperty.Name,

obj.Properties[myProperty.Name].Value);

}

else

{

myPropertyResults[Name] = obj.Properties[myProperty.Name].Value;

}

}

string insertSql = "INSERT INTO NetworkTab VALUES (@myPropertyResults)";

var types = new[] { 1, 2, 3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40 };

using (SqlCommand command = new SqlCommand(insertSql, sqlConnection))

{

foreach (var myPropertyResult in types)

{

command.Parameters.AddWithValue("@myPropertyResults", myPropertyResults);

command.ExecuteNonQuery();

}

}

foreach (var myPropertyResult in myPropertyResults)

{

InfoList.Items.Add(myPropertyResult.Key + ":" + myPropertyResult.Value);

}

}

}

private void MemoryInfo()

{

SqlConnection = new SqlConnection("server=LAPTOP-NDOBE5PK; Trusted\_Connection=Yes; DataBase=Win32Project;");

SqlCommand = sqlConnection.CreateCommand();

sqlConnection.Open();

Dictionary<object, object> myPropertyResults = new Dictionary<object, object>();

ManagementClass myManagementClass = new ManagementClass("Win32\_MemoryArray");

ManagementObjectCollection myManagementCollection = myManagementClass.GetInstances();

PropertyDataCollection myProperties = myManagementClass.Properties;

foreach (var obj in myManagementCollection)

{

foreach (var myProperty in myProperties)

{

bool keyExists = myPropertyResults.ContainsKey(myProperty.Name);

if (keyExists == false)

{

myPropertyResults.Add(myProperty.Name,

obj.Properties[myProperty.Name].Value);

}

else

{

myPropertyResults[Name] = obj.Properties[myProperty.Name].Value;

}

}

foreach (var myPropertyResult in myPropertyResults)

{

InfoList.Items.Add(myPropertyResult.Key + ":" + myPropertyResult.Value);

for (int i = 0; i < myProperties.Count; i++)

{

string cmdText = "INSERT \* INTO [MemoryTab] VALUES (myPropertyResults)";

SqlCommand cmd = new SqlCommand(cmdText, sqlConnection);

cmd.Parameters.AddWithValue("myPropertyResult.Key", myPropertyResults);

}

}

}

}

private void MonitorInfo()

{

SqlConnection = new SqlConnection("server=LAPTOP-NDOBE5PK; Trusted\_Connection=Yes; DataBase=Win32Project;");

SqlCommand = sqlConnection.CreateCommand();

sqlConnection.Open();

Dictionary<object, object> myPropertyResults = new Dictionary<object, object>();

ManagementClass myManagementClass = new ManagementClass("Win32\_VideoController");

ManagementObjectCollection myManagementCollection = myManagementClass.GetInstances();

PropertyDataCollection myProperties = myManagementClass.Properties;

foreach (var obj in myManagementCollection)

{

foreach (var myProperty in myProperties)

{

bool keyExists = myPropertyResults.ContainsKey(myProperty.Name);

if (keyExists == false)

{

myPropertyResults.Add(myProperty.Name,

obj.Properties[myProperty.Name].Value);

}

else

{

myPropertyResults[Name] = obj.Properties[myProperty.Name].Value;

}

}

foreach (var myPropertyResult in myPropertyResults)

{

for (int i = 0; i < myProperties.Count; i++)

{

string cmdText = "INSERT \* INTO [MonitorTab] VALUES (myPropertyResults)";

SqlCommand cmd = new SqlCommand(cmdText, sqlConnection);

cmd.Parameters.AddWithValue("myPropertyResult.Key", myPropertyResults);

}

InfoList.Items.Add(myPropertyResult.Key + ":" + myPropertyResult.Value);

}

}

}

private void MotherboardInfo()

{

SqlConnection = new SqlConnection("server=LAPTOP-NDOBE5PK; Trusted\_Connection=Yes; DataBase=Win32Project;");

SqlCommand = sqlConnection.CreateCommand();

sqlConnection.Open();

Dictionary<object, object> myPropertyResults = new Dictionary<object, object>();

ManagementClass myManagementClass = new ManagementClass("Win32\_MotherboardDevice");

ManagementObjectCollection myManagementCollection = myManagementClass.GetInstances();

PropertyDataCollection myProperties = myManagementClass.Properties;

foreach (var obj in myManagementCollection)

{

foreach (var myProperty in myProperties)

{

bool keyExists = myPropertyResults.ContainsKey(myProperty.Name);

if (keyExists == false)

{

myPropertyResults.Add(myProperty.Name,

obj.Properties[myProperty.Name].Value);

}

else

{

myPropertyResults[Name] = obj.Properties[myProperty.Name].Value;

}

}

foreach (var myPropertyResult in myPropertyResults)

{

for (int i = 0; i < myProperties.Count; i++)

{

string cmdText = "INSERT \* INTO [MotherTab] VALUES (myPropertyResults)";

SqlCommand cmd = new SqlCommand(cmdText, sqlConnection);

cmd.Parameters.AddWithValue("myPropertyResult.Key", myPropertyResults);

}

InfoList.Items.Add(myPropertyResult.Key + ":" + myPropertyResult.Value);

}

}

}

private void KeyboardInfo()

{

SqlConnection = new SqlConnection("server=LAPTOP-NDOBE5PK; Trusted\_Connection=Yes; DataBase=Win32Project;");

SqlCommand = sqlConnection.CreateCommand();

sqlConnection.Open();

Dictionary<object, object> myPropertyResults = new Dictionary<object, object>();

ManagementClass myManagementClass = new ManagementClass("Win32\_Keyboard");

ManagementObjectCollection myManagementCollection = myManagementClass.GetInstances();

PropertyDataCollection myProperties = myManagementClass.Properties;

foreach (var obj in myManagementCollection)

{

foreach (var myProperty in myProperties)

{

bool keyExists = myPropertyResults.ContainsKey(myProperty.Name);

if (keyExists == false)

{

myPropertyResults.Add(myProperty.Name,

obj.Properties[myProperty.Name].Value);

}

else

{

myPropertyResults[Name] = obj.Properties[myProperty.Name].Value;

}

}

foreach (var myPropertyResult in myPropertyResults)

{

for (int i = 0; i < myProperties.Count; i++)

{

string cmdText = "INSERT \* INTO [KeyboardTab] VALUES (myPropertyResults)";

SqlCommand cmd = new SqlCommand(cmdText, sqlConnection);

cmd.Parameters.AddWithValue("myPropertyResult.Key", myPropertyResults);

}

InfoList.Items.Add(myPropertyResult.Key + ":" + myPropertyResult.Value);

}

}

}

internal void InfoList\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

}

Loginpage

using System.Configuration;

using System.Data;

using System.Data.SqlClient;

using System.Windows;

using System.Windows.Controls;

namespace Win32

{

/// <summary>

/// Логика взаимодействия для login.xaml

/// </summary>

public partial class loginpage : Page

{

public Autoriz autor;

string connectionString;

public loginpage(Autoriz \_autor)

{

InitializeComponent();

connectionString = ConfigurationManager.ConnectionStrings["SQLConnection"].ConnectionString;

autor = \_autor;

}

private void login\_TextChanged(object sender, TextChangedEventArgs e)

{

}

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

{

autor.OpenPage(Autoriz.pages.reginpage);

}

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

{

SqlConnection connection = new SqlConnection("server = LAPTOP-NDOBE5PK; Trusted\_Connection = Yes; DataBase = Win32Project;");

connection.Open();

{

if (login.Text.Length > 0) // проверяем введён ли логин

{

if (password.Password.Length > 0) // проверяем введён ли пароль

{ // ищем в базе данных пользователя с такими данными

DataTable dt\_user = Autoriz.Select("SELECT \* FROM [dbo].[Users] WHERE [login] = '" + login.Text + "' AND [password] = '" + password.Password + "'");

if (dt\_user.Rows.Count > 0) // если такая запись существует

{

MessageBox.Show("Пользователь авторизовался"); // говорим, что авторизовался

GetRole();

}

else MessageBox.Show("Пользователя не найден"); // выводим ошибку

}

else MessageBox.Show("Введите пароль"); // выводим ошибку

}

else MessageBox.Show("Введите логин"); // выводим ошибку

}

}

public void GetRole()

{

SqlConnection connection = new SqlConnection("server = LAPTOP-NDOBE5PK; Trusted\_Connection = Yes; DataBase = Win32Project;");

connection.Open();

AdminWindow a = new AdminWindow();

Autoriz a1 = new Autoriz();

MainWindow main = new MainWindow();

DataTable dt\_users = Autoriz.Select("SELECT \* FROM Users WHERE role = 'Админ'");

if (dt\_users.Rows.Count > 0) // если такая запись существует

{

a1.Close();

a.Show();

}

else

{

a1.Close();

main.Show();

}

}

}

}

Reginpage

using System.Windows;

using System.Windows.Controls;

namespace Win32

{

/// <summary>

/// Логика взаимодействия для reginpage.xaml

/// </summary>

public partial class reginpage : Page

{

public Autoriz autor;

public reginpage(Autoriz \_autor)

{

InitializeComponent();

autor = \_autor;

}

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

{

autor.OpenPage(Autoriz.pages.loginpage);

}

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

{

CheckPassword();

}

private void CheckPassword()

{

if (login.Text.Length > 0) // проверяем логин

{

if (password.Password.Length > 0) // проверяем пароль

{

if (password\_Copy.Password.Length > 0) // проверяем второй пароль

{

}

else MessageBox.Show("Повторите пароль");

}

else MessageBox.Show("Укажите пароль");

}

else MessageBox.Show("Укажите логин");

string[] dataLogin = login.Text.Split('@'); // делим строку на две части

if (dataLogin.Length == 2) // проверяем если у нас две части

{

string[] data2Login = dataLogin[1].Split('.'); // делим вторую часть ещё на две части

if (data2Login.Length == 2)

{

}

else MessageBox.Show("Укажите логин в форме х@x.x");

}

else MessageBox.Show("Укажите логин в форме х@x.x");

if (password.Password.Length >= 6)

{

bool en = true; // английская раскладка

bool symbol = false; // символ

bool number = false; // цифра

for (int i = 0; i < password.Password.Length; i++) // перебираем символы

{

if (password.Password[i] >= 'А' && password.Password[i] <= 'Я') en = false; // если русская раскладка

if (password.Password[i] >= '0' && password.Password[i] <= '9') number = true; // если цифры

if (password.Password[i] == '\_' || password.Password[i] == '-' || password.Password[i] == '!') symbol = true; // если символ

}

if (!en)

MessageBox.Show("Доступна только английская раскладка"); // выводим сообщение

else if (!symbol)

MessageBox.Show("Добавьте один из следующих символов: \_ - !"); // выводим сообщение

else if (!number)

MessageBox.Show("Добавьте хотя бы одну цифру"); // выводим сообщение

if (en && symbol && number) // проверяем соответствие

{

}

}

else MessageBox.Show("пароль слишком короткий, минимум 6 символов");

if (password.Password == password\_Copy.Password) // проверка на совпадение паролей

{

MessageBox.Show("Пользователь зарегистрирован");

}

else MessageBox.Show("Пароли не совподают");

}

private void login\_TextChanged(object sender, TextChangedEventArgs e)

{

}

}

}

# ПРИЛОЖЕНИЕ В

PadesBut

<UserControl x:Class="Win32.PadesBut"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Win32" Height="425" Width="800">

<Grid>

<Canvas x:Name="MCanvas" Background="#FF354935" Focusable="True" Grid.ColumnSpan="4" Margin="0,0,-0.4,-1" Grid.Row="2" Grid.Column="2" />

<StackPanel x:Name="Buttons" Margin="495,10,10,0" VerticalAlignment="Top" Height="369" >

<Button x:Name="NetworkBut" Height="38" FontFamily="SimSun" FontSize="14" Click="NetworkBut\_Click">Network</Button>

<Button x:Name="ProccesorBut" Content="Proccesor" Height="38" FontFamily="SimSun" FontSize="14" Click="ProccesorBut\_Click"/>

<Button x:Name="MemoryBut" Height="38" FontFamily="SimSun" FontSize="14" Click="MemoryBut\_Click">Memory</Button>

<Button x:Name="KeybBut" Height="38" FontFamily="SimSun" FontSize="14" Click="KeybBut\_Click">Keyboard</Button>

<Button x:Name="MotherBut" Height="38" FontFamily="SimSun" FontSize="14" Click="MotherBut\_Click">Motherboard</Button>

<Button x:Name="MonitorBut" Height="38" FontFamily="SimSun" FontSize="14" Click="MonitorBut\_Click">Monitor</Button>

<Button x:Name="Clear" Height="38" FontFamily="SimSun" FontSize="14" Click="Clear\_Click">Clear</Button>

</StackPanel>

<StackPanel Margin="0,10,340,-0.4" Background="LightYellow">

<ListBox x:Name="InfoList" Height="415" Background="Black" FontFamily="SimSun" FontSize="14" Foreground="White" SelectionChanged="InfoList\_SelectionChanged"/>

</StackPanel>

</Grid>

</UserControl>

Loginpage

<Page x:Class="Win32.loginpage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Win32"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="loginpage">

<Grid>

<Grid Background="#FF686868">

<Grid.RowDefinitions>

<RowDefinition Height="110\*"/>

<RowDefinition Height="187\*"/>

<RowDefinition Height="0\*"/>

<RowDefinition Height="9\*"/>

<RowDefinition Height="104\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="177\*" />

<ColumnDefinition Width="154\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="0" Grid.ColumnSpan="2" Margin="0,0,-0.4,191" Grid.RowSpan="2">

<Label Content="Hello, User!" FontFamily="SimSun" Margin="140,33,501,21" FontSize="20" Grid.Column="0"/>

<Label Content="Please, sign in or register" HorizontalAlignment="Center" FontFamily="SimSun" FontSize="16" Height="29" Margin="479,0,52,0" Width="229"/>

</StackPanel>

<Grid Grid.Row="1" HorizontalAlignment="Center" Width="358" Margin="57.8,0.2,46.6,0" Grid.Column="1">

<StackPanel Margin="0,0,23,0">

<Grid Height="103">

<Label Content="Enter login" FontFamily="SimSun" FontSize="16" Margin="64,0,0.2,0" />

<TextBox x:Name="login" Margin="64,34,30,39" TextChanged="login\_TextChanged" />

</Grid>

</StackPanel>

<StackPanel Margin="0,116,23,0">

<Grid Height="13" Grid.Column="1" Grid.Row="1"/>

<Label Content="Enter password" FontFamily="SimSun" FontSize="16" Height="33" Margin="64,0,70.2,0"/>

<PasswordBox x:Name="password" Height="28" Margin="64,0,33,0"></PasswordBox>

</StackPanel>

</Grid>

<StackPanel x:Name="buttonaut" Orientation="Horizontal" Grid.Row="2" Grid.Column="1" Margin="1.8,0,-0.4,5.6" Grid.RowSpan="2" />

<Grid Margin="67.8,0.4,39.6,67" Grid.Row="4" Grid.Column="1">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition Width="0\*"/>

</Grid.ColumnDefinitions>

<Button x:Name="regin" Content="Sign up" Width="69" FontFamily="SimSun" FontSize="15" Background="#FF60B35B" HorizontalAlignment="Left" Click="regin\_Click" />

<Button x:Name="enter" Content="Sign in" Width="73" FontFamily="SimSun" FontSize="15" Background="#FF60B35B" HorizontalAlignment="Right" Margin="0,0,43.2,0.4" Click="enter\_Click" />

<StackPanel x:Name="getrole"/>

</Grid>

<StackPanel HorizontalAlignment="Center" Grid.Row="1" Width="466" Margin="35,0.2,30.2,0">

<Image Source="lain.gif" Margin="30,36,18.6,18" ClipToBounds="True" Height="199"/>

</StackPanel>

</Grid>

</Grid>

</Page>

Reginpage

<Page x:Class="Win32.reginpage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Win32"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="reginpage">

<Grid>

<Grid Background="#FF717171">

<Grid.RowDefinitions>

<RowDefinition Height="110\*"/>

<RowDefinition Height="187\*"/>

<RowDefinition Height="0\*"/>

<RowDefinition Height="9\*"/>

<RowDefinition Height="104\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="177\*" />

<ColumnDefinition Width="154\*"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="0" Grid.ColumnSpan="2" Margin="0,0,-0.4,191" Grid.RowSpan="2">

<Label Content="Hello, User!" FontFamily="SimSun" Margin="140,33,501,21" FontSize="20" Grid.Column="0"/>

<Label Content="Please, sign in or register" HorizontalAlignment="Center" FontFamily="SimSun" FontSize="16" Height="29" Margin="479,0,52,0" Width="229"/>

</StackPanel>

<Grid Grid.Row="1" HorizontalAlignment="Center" Width="358" Margin="57.8,0.2,46.6,0" Grid.Column="1">

<StackPanel Margin="0,0,23,-104.4">

<Grid Height="103">

<Label Content="Enter login" FontFamily="SimSun" FontSize="16" Margin="49,0,0.2,0" />

<TextBox x:Name="login" Margin="19,38,30,39" TextChanged="login\_TextChanged" />

</Grid>

</StackPanel>

<StackPanel Margin="10,59,23,-104.4">

<Label Content="Enter password" FontFamily="SimSun" FontSize="16" Height="33" Margin="50,0,70.6,0"/>

<Grid Height="13" Grid.Column="1" Grid.Row="1"/>

<PasswordBox x:Name="password" Height="25" Margin="23,0,33,0"/>

<Label Content="Repeat password" FontFamily="SimSun" FontSize="16" Height="35" Margin="50,0,33.6,0"/>

<PasswordBox x:Name="password\_Copy" Height="32" Margin="23,0,33.2,0"/>

</StackPanel>

</Grid>

<StackPanel x:Name="buttonaut" Orientation="Horizontal" Grid.Row="2" Grid.Column="1" Margin="2,0.4,0,66.6" Grid.RowSpan="3" />

<Grid Margin="67.8,0.4,39.6,67" Grid.Row="4" Grid.Column="1">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition Width="0\*"/>

</Grid.ColumnDefinitions>

<StackPanel HorizontalAlignment="Center" Width="250" Margin="-426,-250,440.8,47">

<Image Source="lain.gif" Margin="30,36,18.6,18" ClipToBounds="True" Height="199"/>

</StackPanel>

<Button x:Name="back" Content="Back" Width="73" FontFamily="SimSun" FontSize="15" Background="#FF60B35B" HorizontalAlignment="Right" Margin="0,-1,40.8,1" Click="back\_Click" />

<Button x:Name="reg" Content="Sign up" Width="69" FontFamily="SimSun" FontSize="15" Background="#FF60B35B" HorizontalAlignment="Left" Click="reg\_Click" Margin="10,0,0,-1" />

</Grid>

</Grid>

</Grid>

</Page>

Tables

<UserControl x:Class="Win32.Tables"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Win32"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800">

<Grid>

<Canvas x:Name="MCanvas" Background="#FF354935" Focusable="True" Margin="0,0,0,9.6" >

<DataGrid x:Name="InfoGrid" HorizontalAlignment="Left" Height="374" VerticalAlignment="Top" Width="415" AutoGenerateColumns="True" Canvas.Left="210" Canvas.Top="10" SelectionChanged="InfoGrid\_SelectionChanged\_1" Background="Black" Foreground="Black"/>

<StackPanel x:Name="Paging" Height="190" Width="110">

<Button x:Name="Users" Content="Users" Width="90" FontFamily="SimSun" FontSize="15" Background="#FF60B35B" HorizontalAlignment="Left" Click="Users\_Click" />

<Button x:Name="Proccesor" Content="Proccesor" Width="90" FontFamily="SimSun" FontSize="15" Background="#FF60B35B" HorizontalAlignment="Left" Click="Proccesor\_Click" />

<Button x:Name="Keyboard" Content="Keyboard" Width="90" FontFamily="SimSun" FontSize="15" Background="#FF60B35B" HorizontalAlignment="Left" Click="Keyboard\_Click" />

<Button x:Name="Monitor" Content="Monitor" Width="90" FontFamily="SimSun" FontSize="15" Background="#FF60B35B" HorizontalAlignment="Left" Click="Monitor\_Click" />

<Button x:Name="Memory" Content="Memory" Width="90" FontFamily="SimSun" FontSize="15" Background="#FF60B35B" HorizontalAlignment="Left" Click="Memory\_Click" />

<Button x:Name="Network" Content="Network" Width="90" FontFamily="SimSun" FontSize="15" Background="#FF60B35B" HorizontalAlignment="Left" Click="Network\_Click" />

<Button x:Name="Motherboard" Content="Motherboard" Width="90" FontFamily="SimSun" FontSize="15" Background="#FF60B35B" HorizontalAlignment="Left" Click="Motherboard\_Click" />

<Button x:Name="BackTo" Content="Back to app" Width="90" FontFamily="SimSun" FontSize="15" Background="#FF60B35B" HorizontalAlignment="Left" Click="BackTo\_Click"></Button>

</StackPanel>

<StackPanel Height="60" Width="100" Canvas.Left="665" Canvas.Top="350" HorizontalAlignment="Center">

<Button x:Name="upd" Content="Update" Width="90" FontFamily="SimSun" FontSize="15" Background="#FF60B35B" Click="upd\_Click"></Button>

<Button x:Name="delet" Content="Delete" Width="90" FontFamily="SimSun" FontSize="15" Background="#FF60B35B"/>

</StackPanel>

</Canvas>

</Grid>

</UserControl>