Казань, 2021

СОДЕРЖАНИЕ

[Введение 2](#_Toc184145106)

[Цель и задачи 3](#_Toc184145107)

[1 разработка системного проекта 4](#_Toc184145108)

[2.2 Диаграмма вариантов использования 6](#_Toc184145109)

[3.2 Компоненты, использованные в программе 9](#_Toc184145110)

[3.4 Описание пользовательского интерфейса 11](#_Toc184145111)

[3.5 Запросы и обращение к базе данных 12](#_Toc184145112)

[Запросы для аналитики 14](#_Toc184145113)

[4. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 16](#_Toc184145114)

[4.1 Работа с программой 16](#_Toc184145115)

[5. ТЕСТИРОВАНИЕ 21](#_Toc184145116)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 25](#_Toc184145117)

[ПРИЛОЖЕНИЕ А 32](#_Toc184145118)

[ПРИЛОЖЕНИE B 35](#_Toc184145119)

### Введение

Современный подход к управлению оборудованием начинается с автоматизации и упрощения процессов. Превратите свою повседневную работу в эффективное и упорядоченное взаимодействие.

Однако, многие компании сталкиваются с трудностями, связанными с отсутствием удобных инструментов для управления данными и процессами. Наш программный продукт **«Учет техники»** предназначен для автоматизации процессов взаимодействия с пользователями, улучшения качества обслуживания клиентов за счёт сохранения информации о пользователях, их данных и управлении оборудованием, а также анализа проведённых операций.

Система предоставляет удобный интерфейс для работы с обширным каталогом данных и поддерживает следующие ключевые функции:

* добавление и редактирование пользователей;
* управление оборудованием и его назначением;
* редактирование профилей сотрудников;
* просмотр и управление списком техники;
* проведение аналитики и последующий анализ данных.

В рамках разработки проекта было реализовано создание программного продукта **«Учет техники»**, направленного на автоматизацию и упрощение бизнес-процессов.

Для достижения поставленной цели были сформулированы и решены следующие задачи:

* закрепление теоретических знаний по программированию;
* углубленное изучение возможностей языка программирования C#;
* использование дополнительных возможностей для создания графического интерфейса клиентского приложения с использованием технологии WPF;
* проектирование интуитивно понятного интерфейса для пользователя.

Центральным элементом системы является база данных, разработанная с использованием **MS SQL Server**, которая организует и хранит информацию о пользователях и оборудовании.

### Цель и задачи

Целью данного проекта является создание автоматизированной информационной системы **«Учет техники»** для управления данными о пользователях и оборудовании.

Задачи проекта включают:

* проектирование базы данных для хранения информации об оборудовании и пользователях;
* разработку и реализацию удобного интерфейса пользователя;
* изучение основ и возможностей технологии WPF для создания клиентских приложений;
* тестирование и внедрение автоматизированной системы управления.

Для реализации данного проекта была выбрана среда разработки **Visual Studio 2019**, а для хранения данных – **MS SQL Server**.

# 1 разработка системного проекта

1.1 Анализ предметной области

Целью разработки данного проекта является создание программного обеспечения для учёта и управления техникой на предприятии. Программа позволит автоматизировать процессы взаимодействия с пользователями и улучшить обслуживание за счёт централизованного хранения данных о пользователях и техники.

Для использования программы пользователям необходимо зарегистрироваться через специальное меню. Система будет сохранять данные, такие как:

Фамилия, имя, отчество (ФИО),

Логин и пароль для доступа к системе.

Ключевым элементом базы данных являются записи о технике, включая:

ID (уникальный идентификатор),

Название техники,

Стоимость,

Состав комплектующих (для понимания структуры товаров).

В проекте предусмотрены две основные роли:

Администратор – обладает полным доступом ко всем функциям системы.

Пользователь – ознакомиться с техникой и править личную информацию через интерфейс программы.

Обоснование выбора технологий

Система управления базами данных (СУБД):

Для реализации проекта была выбрана Microsoft SQL Server 2019 – надёжная и масштабируемая реляционная СУБД, используемая разработчиками по всему миру. Эта платформа отлично подходит для приложений, написанных на языке программирования C# и работающих в экосистеме .NET.

Преимущества SQL Server 2019:

Высокая производительность – сервер обеспечивает быструю обработку запросов.

Надёжность и безопасность – встроенные механизмы шифрования данных.

Удобство использования – простой интерфейс для администрирования и интеграции с другими продуктами Microsoft.

Язык программирования:

В качестве языка разработки выбран C#. Это объектно-ориентированный язык с C-подобным синтаксисом, который позволяет разрабатывать мощные и гибкие приложения.

Преимущества C#:

Поддержка наследования, полиморфизма, статической типизации и других функций ООП.

Совместимость с .NET Framework, что ускоряет разработку и интеграцию приложений.

Быстрое развитие и востребованность на рынке.

Первая версия C# была выпущена в 2002 году, а к 2019 году появилась версия C# 8.0, значительно расширившая возможности языка.

Среда разработки:

Для написания и тестирования проекта используется Visual Studio 2019, которая предоставляет:

Редактор исходного кода с автозавершением (IntelliSense) и инструментами рефакторинга.

Мощный встроенный отладчик.

Поддержку компиляции и графические конструкторы интерфейсов.

Visual Studio позволяет разрабатывать как консольные приложения, так и графические интерфейсы, обеспечивая гибкость и удобство в реализации проекта.

Эти инструменты и технологии обеспечивают надёжную и удобную платформу для разработки программного обеспечения для автоматизации учёта и управления продажами техники.

2 Проектирование программного продукта

2.1 Графическое описание программы

Проектирование является одним из ключевых этапов в разработке программного продукта. На данном этапе осуществляется моделирование основных элементов продукта — модели базы данных, процессов и кода. Эффективным способом моделирования продукта является графическое описание на основе Unified Modeling Language (UML).

UML – это графический язык моделирования, в котором используются диаграммы для визуализации и конструирования архитектуры проекта.

Все UML-диаграммы делятся на две группы: структурные и поведенческие.

К структурным диаграммам относится диаграмма отношений сущностей (ERD). Это визуальное представление базы данных, которое показывает, как элементы в базе данных связаны друг с другом. ERD состоит из двух типов объектов — сущностей и отношений.

К поведенческим диаграммам относятся диаграмма вариантов использования (Use case diagram) и диаграмма последовательности (Sequence diagram).

Диаграмма вариантов использования описывает функциональное назначение системы, то, что система будет делать в процессе своего функционирования.

Диаграмма последовательности предназначена для моделирования взаимодействия объектов системы во времени, а также обмена сообщениями между ними.

## 2.2 Диаграмма вариантов использования

Любые программные системы проектируются с учетом того, что в процессе своей работы они будут использоваться людьми.

Сущности, с которыми взаимодействует система в процессе своей работы, называются актерами. Графически актер изображается " человечком ".

Для каждого актера в диаграмме вариантов использования определены свои возможности, которые называются прецедентами. Графически прецедент изображается эллипсом с его описание внутри.

Прецедентами для пользователя являются редактирование информации в профиле: фамилия, имя, фото, создание мюсли, просмотр и редактирование ранее созданных мюсли, заказ мюсли. Данная диаграмма отображена на рисунке 2.1.

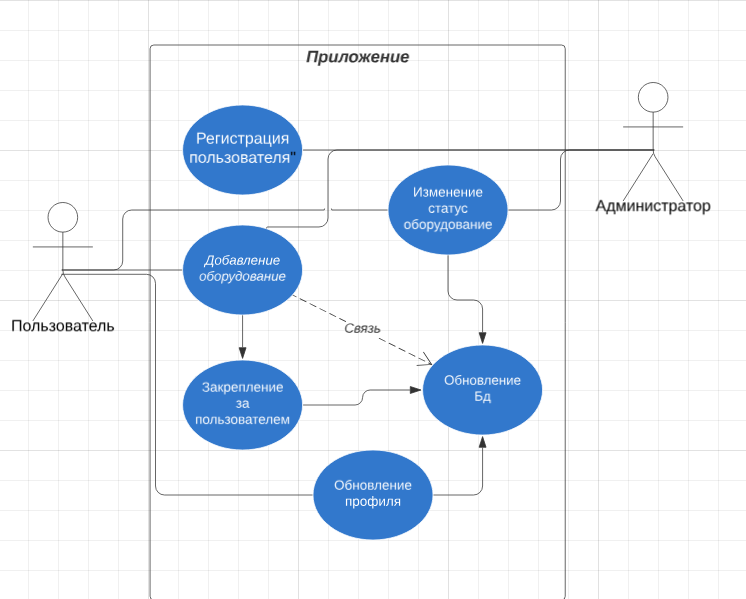


Рисунок 2.1 Use Case Diagram вариантов использования

Рисунок 2.1 Диаграмма USE CASE

2.3 ER-Диаграмма

Схема «сущность-связь» (также ERD или ER-диаграмма) — это разновидность блок-схемы, где показано, как разные «сущности» (люди, объекты, концепции и так далее) связаны между собой внутри системы. ER-диаграммы чаще всего применяются для проектирования и отладки реляционных баз данных в сфере образования, исследования и разработки программного обеспечения и информационных систем для бизнеса. ER-диаграммы (или ER-модели) полагаются на стандартный набор символов, включая прямоугольники, ромбы, овалы и соединительные линии, для отображения сущностей, их атрибутов и связей. Эти диаграммы устроены по тому же принципу, что и грамматические структуры: сущности выполняют роль существительных, а связи — глаголов (Рисунок 2.2).

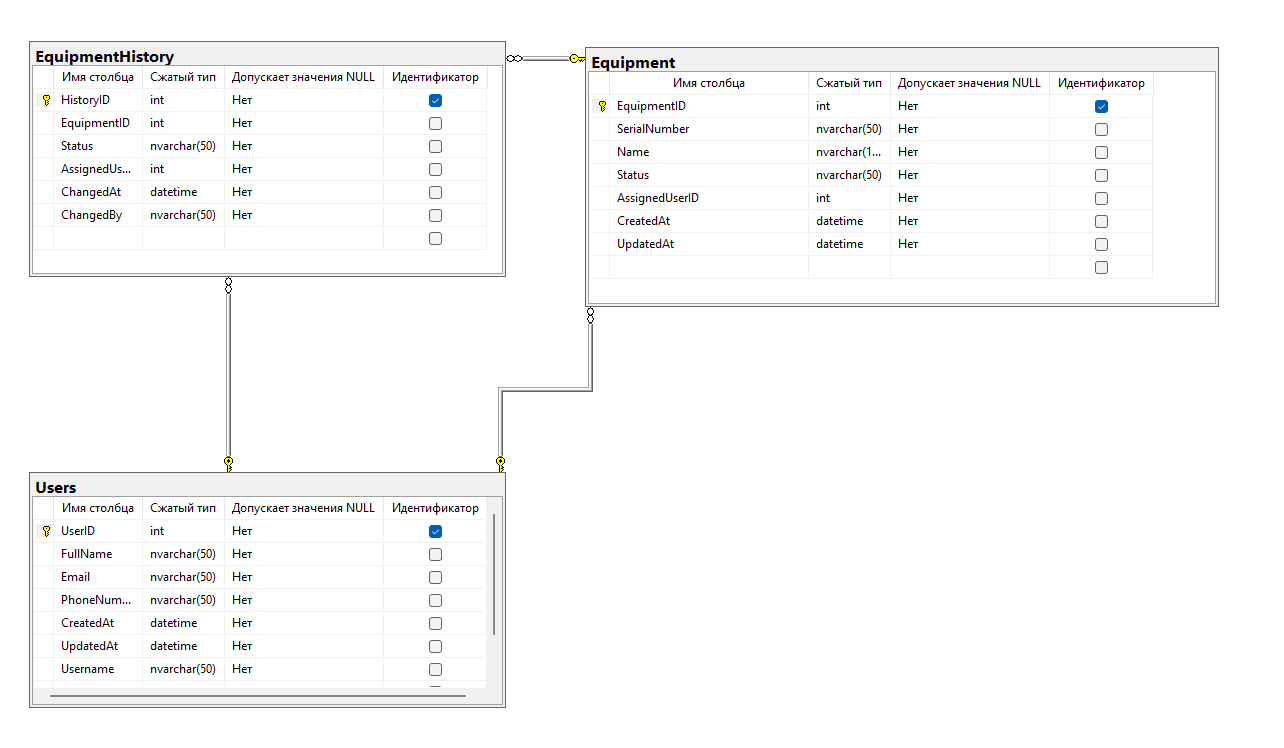


Рисунок 2.2 ER-диаграмма базы данных «Учет техники»

3 Реализация проекта

3.1 Проектирование проекта

Информационные системы предназначены для оперативного предоставления пользователю данных о внешнем мире посредством реализации вопросно-ответного взаимодействия. Это позволяет выделить определённый фрагмент внешнего мира, называемый предметной областью, для использования в системе. Данные о внешнем мире, представленные в информационной системе, имеют строго определённый формат. Такое ограничение способствует конкретизации семантики информации и задаёт её интерпретацию в пределах системы.

Совокупность данных, связей между ними, а также операций, выполняемых над ними, формирует информационную и функциональную модели предметной области. Эти модели описывают состояние предметной области с необходимой точностью и являются ключевым этапом разработки базы данных. При этом на этапе анализа требований модели создаются независимо от технологий, используемых для реализации.

Предметной областью текущего проекта является система «Учёт техники», предназначенная для управления данными о пользователях и оборудовании. Система решает задачу автоматизации взаимодействия между пользователями и оборудованием, предоставляя удобный графический интерфейс и функции управления.

# 3.2 Компоненты, использованные в программе

Для разработки приложения использованы следующие элементы интерфейса:

* ListBox: список, позволяющий пользователю выбирать один или несколько пунктов.
* TextBox: текстовое поле для ввода и редактирования текста. Оно поддерживает работу с длинными строками и защищённым текстом с использованием маски.
* Button: кнопка для взаимодействия с приложением. Основное событие — щелчок мышью, но она также обрабатывает нажатия клавиш и изменения параметров.
* Label: элемент для отображения текста на форме.
* ComboBox: выпадающий список для выбора одного элемента из коллекции.
* PasswordBox: текстовое поле для ввода пароля с подстановочными символами, скрывающими текст.

3.3 Этапы разработки

Этап 1: Создание концепта и прототипа программного продукта

На первом этапе была проведена аналитическая работа с существующими решениями для автоматизации учета оборудования. Эти решения послужили качественным примером как для проектирования пользовательского интерфейса, так и для разработки базового функционала. Основное внимание уделялось упрощению процесса взаимодействия с системой.

Этап 2: Создание рабочей модели без базы данных

Для упрощения разработки на втором этапе была создана тестовая версия приложения, не подключённая к базе данных. Это позволило сосредоточиться на разработке графического интерфейса и отладке базового функционала, не зависящего от внешних источников данных.

Этап 3: Проектирование и создание базы данных

На третьем этапе разработана база данных с использованием MS SQL Server Management Studio 2018. Основой для её проектирования стали атрибуты, определённые на этапе тестирования интерфейса. База данных включает таблицы, содержащие информацию о пользователях, оборудовании, истории действий, и другие важные элементы. Для визуализации структуры была создана диаграмма базы данных, отражающая взаимосвязи между таблицами.

Этап 4: Подключение базы данных к проекту

После создания базы данных она была интегрирована с программным продуктом. Это включало настройку подключения, разработку модели данных и реализацию операций с базой. Диаграмма базы данных наглядно отразила структуру и зависимости между таблицами, что упростило процесс подключения и дальнейшей работы с информацией.

### 3.4 Описание пользовательского интерфейса

Интерфейс информационной системы «Учет техники» разработан в соответствии с основными принципами проектирования пользовательских интерфейсов:

1. **Структурированность**: интерфейс приложения организован таким образом, что схожие по функционалу части системы связаны между собой. Это обеспечивает удобство навигации и логичность взаимодействия.
2. **Обратная связь**: приложение реализует возможность вернуться к предыдущим этапам, позволяя пользователю исправлять ошибки или повторно выполнять действия.
3. **Унификация**: многие элементы интерфейса повторяются в разных частях системы, чтобы пользователь быстрее освоился с приложением и мог интуитивно находить нужные функции.

Интерфейс состоит из следующих основных элементов:

1. **Окно входа в систему**:  
   Пользователь вводит логин и пароль в соответствующие поля и нажимает кнопку «Войти». В случае успешной авторизации открывается главное окно.
2. **Главное окно**:  
   После входа пользователь попадает в центральное окно приложения, которое выполняет роль основного меню. Здесь представлены кнопки для перехода к другим окнам системы:
   * **Добавление пользователя**;
   * **Редактирование профиля**;
   * **Управление техникой**;
   * **Просмотр техники**.  
     Нажатие на каждую из этих кнопок открывает соответствующее окно для выполнения выбранной задачи.

Интерфейс программы прост в использовании, интуитивно понятен и разработан с учётом потребностей конечного пользователя, что делает взаимодействие с системой удобным и эффективным.

# 3.5 Запросы и обращение к базе данных

Невозможно представить программный продукт без запросов или обращений к базе данных, так и в это программном продукте присутствуют данные виды работы с БД.

<connectionStrings>

<add name="AppDbContext"

connectionString="Server=exelllolr;Database=MachineAccounting;User Id=tarif;Password=tarif1;"

providerName="System.Data.SqlClient" />

<add name="MachineAccountingEntities" connectionString="metadata=res://\*/Models.DbModel.csdl|res://\*/Models.DbModel.ssdl|res://\*/Models.DbModel.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=exelllolr;initial catalog=MachineAccounting;integrated security=True;encrypt=False;MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />

</connectionStrings>

Запрос при авторизации пользователя

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

{

// Проверяем, что поля заполнены

if (string.IsNullOrEmpty(Username.Text) || string.IsNullOrEmpty(Password.Password))

{

MessageBox.Show("Пожалуйста, введите логин и пароль.");

return;

}

// Проверяем данные в базе

var user = \_context.Users

.FirstOrDefault(u => u.Email == Username.Text && u.Password == Password.Password);

if (user != null)

{

MessageBox.Show($"Добро пожаловать!");

// Открываем главное окно с передачей объекта пользователя

MainWindow mainWindow = new MainWindow(user);

mainWindow.Show();

this.Close();

}

else

{

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

}

}

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

{

// Переход в MainWindow без проверки пользователя

MainWindow mainWindow = new MainWindow(null); // Передаем null, так как пользователь не авторизован

mainWindow.Show();

this.Close();

}

} }

}

Добавление новой техники

Equipment newEquipment = new Equipment

{

Name = NameTextBox.Text,

Model = ModelTextBox.Text,

SerialNumber = SerialTextBox.Text,

PurchaseDate = DateTime.Parse(PurchaseDateTextBox.Text),

Status = StatusComboBox.SelectedItem.ToString()

};

\_context.Equipment.Add(newEquipment);

\_context.SaveChanges();

**Редактирование техники**

var equipment = \_context.Equipment.FirstOrDefault(e => e.EquipmentId == selectedEquipmentId);

if (equipment != null)

{

equipment.Name = NameTextBox.Text;

equipment.Model = ModelTextBox.Text;

equipment.SerialNumber = SerialTextBox.Text;

equipment.PurchaseDate = DateTime.Parse(PurchaseDateTextBox.Text);

equipment.Status = StatusComboBox.SelectedItem.ToString();

\_context.SaveChanges();

}

**Удаление техники**

var equipment = \_context.Equipment.FirstOrDefault(e => e.EquipmentId == selectedEquipmentId);

if (equipment != null)

{

\_context.Equipment.Remove(equipment);

\_context.SaveChanges();

}

### **Запросы для аналитики**

var equipmentStats = \_context.Equipment

.GroupBy(e => e.Status)

.Select(group => new { Status = group.Key, Count = group.Count() })

.ToList();

# 4. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

# 4.1 Работа с программой

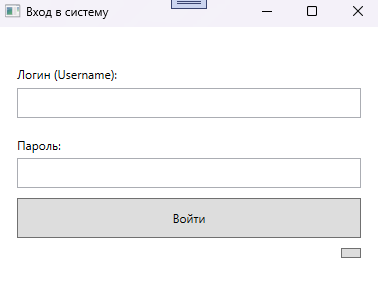


Рисунок 4.1 Меню авторизации

Это первое окно, с которого начинается работа пользователя с приложением. Окно предназначено для входа в систему под учетной записью (Рисунок 4.1).

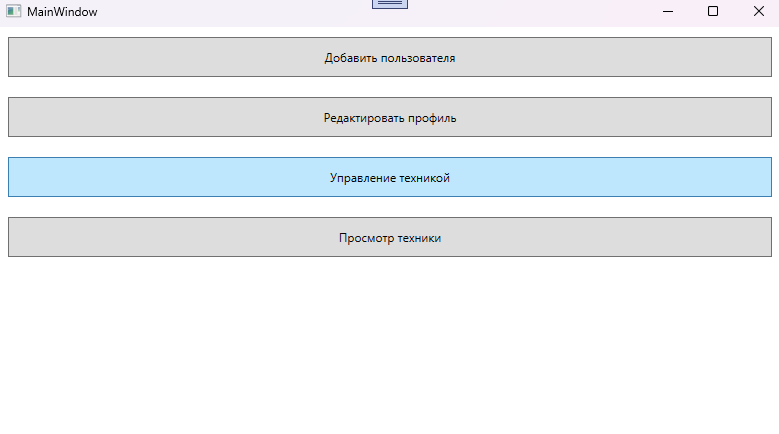


Рисунок 4.2 Главное окно

Основное окно приложения, служит как центр навигации. Содержит кнопки для перехода к функциям приложения (Рисунок 4.2).

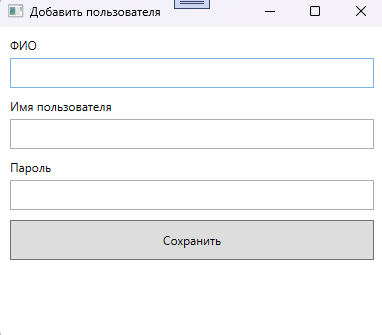


Рисунок 4.3 Окно регистрации

Используется для добавления новых пользователей в систему  
  Проверяет, что все поля заполнены корректно.

 Создаёт нового пользователя в базе данных.

 Показывает сообщение о успешной регистрации. (Рисунок 4.3)

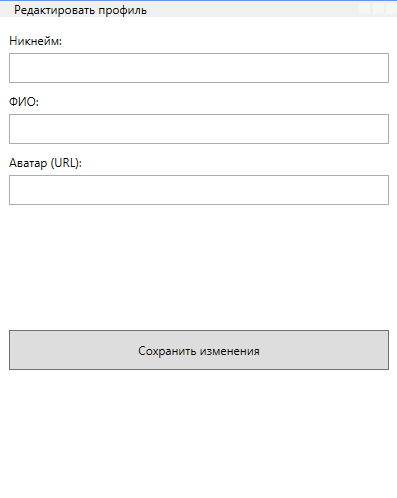


Рисунок 4.4 Окно редактирования профиля

Позволяет текущему пользователю редактировать свои данные

(Рисунок 4.4).

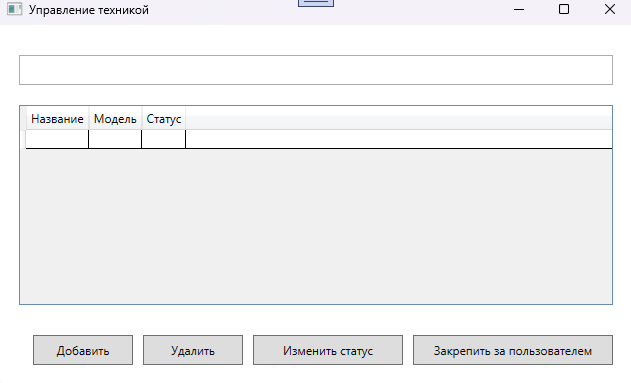


Рисунок 4.7 Окно управления техникой

Позволяет добавлять, редактировать и удалять записи о технике. (Рисунок 4.7).

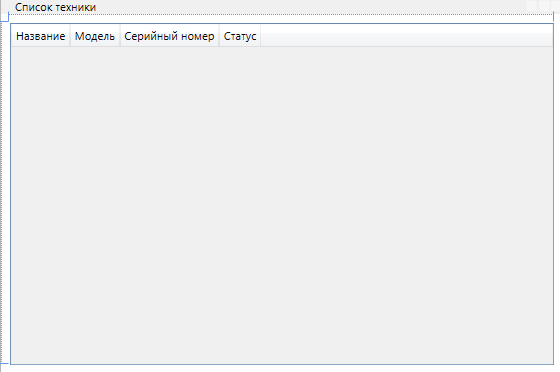


Рисунок 4.8 Окно просмотра техники

Отображает список всей техники, зарегистрированной в системе. (Рисунок 4.8).

# 5. ТЕСТИРОВАНИЕ

Основной задачей тестирования является получение информации о статусе готовности заявленной функциональности системы или приложения.

Тест-кейс — это проверка работоспособности программы или проекта. Тест-кейсы помогают провести проверку продукта без ознакомления со всей документацией. Написанный один раз, удобный в поддержке тест-кейс сэкономит много времени. Тестирование функционала проводится со стороны зарегистрированного и незарегистрированного пользователя.

Тест-кейсы разрабатывается для основных функций программ:

1. Регистрация пользователя (Таблица 5.1)

2. Удаление пользователя (Таблица 5.2)

3. Редактирование личной информации (Таблица 5.3)

**ID:** Test01  
**Приоритет:** medium-priority  
**Описание:** Пользователь авторизуется, вводя логин и пароль. После успешной авторизации открывается главное окно системы.

| **Входные значения** | **Шаги** | **Ожидаемый результат** | **Статус** |
| --- | --- | --- | --- |
| Логин: qwerty | 1) Перейти в окно авторизации. | Пользователь авторизован и попадает в главное окно | Успешно |
| Пароль: qwerty | 2) Ввести логин и пароль. |  |  |
|  | 3) Нажать кнопку «Войти». |  |  |

**ID:** Test02  
**Приоритет:** high-priority  
**Описание:** Администратор удаляет выбранного пользователя из системы.

| **Входные значения** | **Шаги** | **Ожидаемый результат** | **Статус** |
| --- | --- | --- | --- |
| Логин пользователя: test\_user | 1) Перейти в меню управления пользователями. | Пользователь успешно удален из системы | Успешно |
|  | 2) Выбрать пользователя «test\_user». |  |  |
|  | 3) Нажать кнопку «Удалить». |  |  |

**ID:** Test03  
**Приоритет:** medium-priority  
**Описание:** Пользователь редактирует свои личные данные, включая фамилию и имя.

| **Входные значения** | **Шаги** | **Ожидаемый результат** | **Статус** |
| --- | --- | --- | --- |
| Фамилия: Петров | 1) Перейти в меню профиля пользователя. | Данные успешно обновлены в системе | Успешно |
| Имя: Алексей | 2) Ввести новые значения для фамилии и имени. |  |  |
|  | 3) Нажать кнопку «Сохранить». |  |  |

**ID:** Test04  
**Приоритет:** high-priority  
**Описание:** Администратор добавляет новую единицу техники в базу данных.

| **Входные значения** | **Шаги** | **Ожидаемый результат** | **Статус** |
| --- | --- | --- | --- |
| Название товара: Ноутбук “Lenova” | 1) Перейти в меню добавления оборудования. | Новый оборудования успешно добавлен в базу данных | Успешно |
| Цена: 85000 рублей | 2) Ввести данные оборудования: название, цену, категорию. |  |  |
| Категория: Ноутбук | 3) Нажать кнопку «Добавить». |  |  |

**ID:** Test05  
**Приоритет:** low-priority  
**Описание:** Пользователь пытается войти с неверными данными, система выводит сообщение об ошибке.

| **Входные значения** | **Шаги** | **Ожидаемый результат** | **Статус** |
| --- | --- | --- | --- |
| Логин: неверный\_логин | 1) Перейти в окно авторизации. | Сообщение об ошибке: «Неверный логин или пароль». | Успешно |
| Пароль: неверный\_пароль | 2) Ввести некорректный логин и пароль. |  |  |
|  | 3) Нажать кнопку «Войти». |  |  |

**Заключение**

В ходе выполнения проекта была разработана автоматизированная информационная система “Machine Accounting”. В рамках работы была создана и подключена к проекту база данных, обеспечен ввод и вывод данных из таблиц базы, а также разработан программный продукт для учета и управления оборудованием. В процессе разработки были улучшены навыки проектирования и подключения баз данных через MS SQL Server, работы в Visual Studio 2019 и программирования на языке C#. Основные задачи и цели проекта были успешно выполнены.

Разработанная система Machine Accounting предназначена для учета оборудования, управления складскими запасами и обработки истории оборудования. Она позволяет эффективно анализировать данные, проводить мониторинг операций и принимать решения на основе полученной информации. Приложение оснащено интуитивно понятным интерфейсом, который соответствует основным требованиям к функционалу учета и управления.

Проект прошел все этапы разработки, включая проектирование архитектуры, написание кода, тестирование и отладку. Система продемонстрировала стабильную работу, а выявленные ошибки были устранены в процессе тестирования. Программный продукт полностью готов к эксплуатации.

В перспективе приложение “Machine Accounting” планируется интегрировать в существующие бизнес-процессы.

### СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19.102-77 Единая система программной документации. Стадии разработки
2. ГОСТ 19.201-78 Единая система программной документации. Техническое задание. Требования к содержанию и оформлению
3. ГОСТ 19.503-79 Единая система программной документации. Руководство системного программиста. Требования к содержанию и оформлению
4. ГОСТ 19.504-79 Единая система программной документации. Руководство программиста. Требования к содержанию и оформлению
5. ГОСТ 19.505-79 Единая система программной документации. Руководство оператора. Требования к содержанию и оформлению

2) Учебники, монографии, диссертации

1. Акопов, А. С.  Компьютерное моделирование: учебник и практикум для среднего профессионального образования / А. С. Акопов. — Москва : Издательство Юрайт, 2023. — 389 с. — (Профессиональное образование). — ISBN 978-5-534-10712-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/517999 (дата обращения: 11.05.2024).
2. Богатырев, В. А.  Надежность информационных систем : учебное пособие для среднего профессионального образования / В. А. Богатырев. — Москва : Издательство Юрайт, 2023. — 318 с. — (Профессиональное образование). — ISBN 978-5-534-15205-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/497246 (дата обращения: 11.05.2024).
3. Большаков, В. П.  Инженерная и компьютерная графика. Изделия с резьбовыми соединениями : учебное пособие для среднего профессионального образования / В. П. Большаков, А. В. Чагина. — 3-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023. — 152 с. — (Профессиональное образование). — ISBN 978-5-534-15593-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/516875 (дата обращения: 11.05.2023).
4. В. В. Троценко, Системы управления технологическими процессами и информационные технологии : учебное пособие для среднего профессионального образования / В. В. Троценко, В. К. Федоров, А. И. Забудский, В. В. Комендантов. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023. — 136 с. — (Профессиональное образование). — ISBN 978-5-534-09939-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: [https://urait.ru/bcode/493021](https://vk.com/away.php?utf=1&to=https%3A%2F%2Furait.ru%2Fbcode%2F493021) (дата обращения: 11.05.2024).
5. В. В. Троценко, Системы управления технологическими процессами и информационные технологии: учебное пособие для среднего профессионального образования / В. В. Троценко, В. К. Федоров, А. И. Забудский, В. В. Комендантов. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2023. — 136 с. — (Профессиональное образование). — ISBN 978-5-534-09939-3. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/493021 (дата обращения: 11.05.2024).
6. Гордеев, С. И.  Организация баз данных в 2 ч. Часть 1 : учебник для среднего профессионального образования / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023. — 310 с. — (Профессиональное образование). — ISBN 978-5-534-11626-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/518510 (дата обращения: 09.05.2024).
7. Гордеев, С. И.  Организация баз данных в 2 ч. Часть 2 : учебник для среднего профессионального образования / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023. — 513 с. — (Профессиональное образование). — ISBN 978-5-534-11625-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/518511 (дата обращения: 09.05.2024).
8. Древс, Ю. Г.  Имитационное моделирование : учебное пособие для среднего профессионального образования / Ю. Г. Древс, В. В. Золотарёв. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023. — 142 с. — (Профессиональное образование). — ISBN 978-5-534-11951-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/495329 (дата обращения: 11.05.2024).
9. Дьячков, В. П.  Аппаратные средства персонального компьютера : учебное пособие для среднего профессионального образования / В. П. Дьячков. — Москва : Издательство Юрайт, 2023. — 153 с. — (Профессиональное образование). — ISBN 978-5-534-14249-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/519869 (дата обращения: 11.05.2024).
10. Казанский, А. А.  Объектно-ориентированный анализ и программирование на Visual Basic 2013 : учебник для среднего профессионального образования / А. А. Казанский. — Москва : Издательство Юрайт, 2023. — 290 с. — (Профессиональное образование). — ISBN 978-5-534-03833-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/513399 (дата обращения: 09.05.2024).
11. Казанский, А. А.  Программирование на Visual C# : учебное пособие для среднего профессионального образования / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 192 с. — (Профессиональное образование). — ISBN 978-5-534-14130-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/513400 (дата обращения: 09.05.2024).
12. Казарин, О. В.  Основы информационной безопасности: надежность и безопасность программного обеспечения : учебное пособие для среднего профессионального образования / О. В. Казарин, И. Б. Шубинский. — Москва : Издательство Юрайт, 2023. — 342 с. — (Профессиональное образование). — ISBN 978-5-534-10671-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/518005 (дата обращения: 09.05.2024).
13. Казарин, О. В.  Программно-аппаратные средства защиты информации. Защита программного обеспечения : учебник и практикум для среднего профессионального образования / О. В. Казарин, А. С. Забабурин. — Москва : Издательство Юрайт, 2023. — 312 с. — (Профессиональное образование). — ISBN 978-5-534-13221-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/519364 (дата обращения: 11.05.2024).
14. Кудрявцева, И. А.  Программирование: комбинаторная логика : учебное пособие для среднего профессионального образования / И. А. Кудрявцева, М. В. Швецкий. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 524 с. — (Профессиональное образование). — ISBN 978-5-534-15128-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/517677 (дата обращения: 11.05.2024).
15. Кудрявцева, И. А.  Программирование: теория типов : учебное пособие для среднего профессионального образования / И. А. Кудрявцева, М. В. Швецкий. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 652 с. — (Профессиональное образование). — ISBN 978-5-534-15382-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/517829 (дата обращения: 11.05.2024).
16. Маркин, А. В.  Программирование на SQL : учебное пособие для среднего профессионального образования / А. В. Маркин. — Москва : Издательство Юрайт, 2022. — 435 с. — (Профессиональное образование). — ISBN 978-5-534-11093-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/495666 (дата обращения: 11.05.2023).
17. Маркин, А. В.  Системы графовых баз данных. Neo4j : учебное пособие для среднего профессионального образования / А. В. Маркин. — Москва : Издательство Юрайт, 2022. — 303 с. — (Высшее образование). — ISBN 978-5-534-13996-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/496805 (дата обращения: 11.05.2023).
18. Нестеров, С. А.  Базы данных : учебник и практикум для среднего профессионального образования / С. А. Нестеров. — Москва : Издательство Юрайт, 2022. — 230 с. — (Профессиональное образование). — ISBN 978-5-534-11629-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/495981 (дата обращения: 11.05.2024).
19. Новожилов, О. П.  Архитектура компьютерных систем в 2 ч. Часть 1 : учебное пособие для среднего профессионального образования / О. П. Новожилов. — Москва : Издательство Юрайт, 2023. — 276 с. — (Профессиональное образование). — ISBN 978-5-534-10299-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/517678 (дата обращения: 11.05.2024).
20. Новожилов, О. П.  Архитектура компьютерных систем в 2 ч. Часть 2 : учебное пособие для среднего профессионального образования / О. П. Новожилов. — Москва : Издательство Юрайт, 2023. — 246 с. — (Профессиональное образование). — ISBN 978-5-534-10301-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/517679 (дата обращения: 11.05.2024).
21. Станкевич, Л. А.  Интеллектуальные системы и технологии : учебник и практикум для среднего профессионального образования / Л. А. Станкевич. — Москва : Издательство Юрайт, 2023. — 397 с. — (Профессиональное образование). — ISBN 978-5-534-11659-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/495988 (дата обращения: 11.05.2024).
22. Стружкин, Н. П.  Базы данных: проектирование : учебник для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. — Москва : Издательство Юрайт, 2023. — 477 с. — (Профессиональное образование). — ISBN 978-5-534-11635-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/495973 (дата обращения: 11.05.2024).
23. Суворова, Г. М. Адаптивные информационные и коммуникационные технологии в управлении средой обитания : учебное пособие для среднего профессионального образования / Г. М. Суворова. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 210 с. — (Профессиональное образование). — ISBN 978-5-534-15192-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: [https://urait.ru/bcode/497222](https://vk.com/away.php?to=https%3A%2F%2Furait.ru%2Fbcode%2F497222&cc_key=) (дата обращения: 11.05.2024).
24. Толстобров, А. П.  Архитектура ЭВМ : учебное пособие для среднего профессионального образования / А. П. Толстобров. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023. — 154 с. — (Профессиональное образование). — ISBN 978-5-534-13398-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/518772 (дата обращения: 11.05.2024).
25. Черткова, Е. А.  Статистика. Автоматизация обработки информации : учебное пособие для среднего профессионального образования / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023. — 195 с. — (Профессиональное образование). — ISBN 978-5-9916-9342-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/513446 (дата обращения: 11.05.2024).

3) Интернет-ресурсы

1. [www.oracle.com](http://www.oracle.com) – Сайт компании Oracle *(*OracleCorporation) — американская корпорация, второй по размеру выручки разработчик программного обеспечения (после Microsoft).
2. dic.academic.ru – Сайт с научными статьями
3. ru.wikipedia.org – Всемирная web-энциклопедия
4. otzovik.com – Сайт с отзывами
5. [www.beboss.ru](http://www.beboss.ru) – Сайт с описанием бизнес-проектов

### ПРИЛОЖЕНИЕ А

Программное обеспечение **Machine Accounting** предназначено для использования в организациях с целью автоматизации взаимодействия с пользователями, повышения качества обслуживания и сохранения информации о технике и истории взаимодействий. Система помогает оптимизировать бизнес-процессы, анализировать их эффективность, упрощать управление складскими запасами, обрабатывать информацию.

### Основание для разработки

Разработка **Machine Accounting** основана на учебном плане для подготовки специалистов среднего звена по специальности **09.02.07 "Информационные системы и программирование"** с квалификацией **"Специалист по информационным системам"**.

Целью проекта является автоматизация и анализ бизнес-процессов организаций, активно взаимодействующих с пользователями. Программное обеспечение упрощает выполнение задач, распределённых между сотрудниками, и повышает эффективность их работы.

### Технические требования

#### 1. Требования к функциональным характеристикам

ПО должно выполнять следующие функции:

* Назначение прав доступа сотрудникам;
* Редактирование профиля пользователя и интерфейса рабочего стола;
* Управление и редактирование задач;
* Редактирование клиентской базы (пользователи и контактные лица);
* Ведение электронной переписки с пользователями;
* Анализ бизнес-процессов и генерация аналитических отчётов.

#### 2. Требования к надёжности

Надёжная работа программы обеспечивается следующими мерами:

* Использование бесперебойного питания технических средств;
* Применение лицензионного программного обеспечения;
* Выполнение рекомендаций Министерства труда и социальных ресурсов РФ (Постановление от 23 июля 1998 г.);
* Соответствие требованиям ГОСТ 51188-98 ("Защита информации. Испытания программных средств на наличие вирусов").

#### 3. Условия эксплуатации

* Для работы программы требуется не менее одного штатного сотрудника (конечного пользователя).

#### 4. Требования к оборудованию

* Программа должна поддерживать платформы Windows XP и выше.

#### 5. Требования к совместимости

* Необходимое наличие установленного **Microsoft .NET Framework**.

#### 6. Требования к транспортировке и хранению

* Документация поставляется в электронном и печатном формате.

#### 7. Специальные требования

* Отсутствие особых требований к эксплуатации.

### Стадии и этапы разработки

**Стадия 1: Разработка технического задания**

* Подготовка, согласование и утверждение технического задания.

**Стадия 2: Рабочее проектирование**

* Разработка программы;
* Подготовка программной документации;
* Проведение тестирования программы.

Программный продукт ориентирован на повышение эффективности взаимодействия с пользователями и автоматизацию основных бизнес-процессов.

### ПРИЛОЖЕНИE B

﻿using System.Linq;

using System.Windows;

using System.Windows.Navigation;

using Machine\_Accounting.Models;

using Machine\_Accounting.Widows;

namespace MachineAccounting.Views

{

public partial class LoginWindow : Window

{

private readonly MachineAccountingEntities \_context;

public LoginWindow()

{

InitializeComponent();

\_context = new MachineAccountingEntities(); // Инициализация контекста

}

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

{

// Проверяем, что поля заполнены

if (string.IsNullOrEmpty(Username.Text) || string.IsNullOrEmpty(Password.Password))

{

MessageBox.Show("Пожалуйста, введите логин и пароль.");

return;

}

// Проверяем данные в базе

var user = \_context.Users

.FirstOrDefault(u => u.Email == Username.Text && u.Password == Password.Password);

if (user != null)

{

MessageBox.Show($"Добро пожаловать!");

// Открываем главное окно с передачей объекта пользователя

MainWindow mainWindow = new MainWindow(user);

mainWindow.Show();

this.Close();

}

else

{

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

}

}

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

{

// Переход в MainWindow без проверки пользователя

MainWindow mainWindow = new MainWindow(null); // Передаем null, так как пользователь не авторизован

mainWindow.Show();

this.Close();

}

}

}

﻿using MachineAccounting.Views;

using System.Windows;

namespace Machine\_Accounting.Widows

{

public partial class MainWindow : Window

{

public MainWindow(Models.Users user)

{

InitializeComponent();

}

// Открытие окна для добавления пользователя

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

{

var addUserWindow = new AddUserWindow(); // Предполагается, что это окно уже создано

addUserWindow.ShowDialog(); // Открыть модально

}

// Открытие окна для редактирования профиля

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

{

var editProfileWindow = new EditProfileWindow(); // Предполагается, что это окно уже создано

editProfileWindow.ShowDialog();

}

// Открытие окна управления техникой

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

{

var manageEquipmentWindow = new ManageEquipmentWindow(); // Предполагается, что это окно уже создано

manageEquipmentWindow.ShowDialog();

}

// Открытие окна просмотра техники

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

{

var equipmentWindow = new EquipmentWindow(); // Предполагается, что это окно уже создано

equipmentWindow.ShowDialog();

}

}

}

﻿using System.Windows;

using MachineAccounting.Models;

namespace MachineAccounting.Views

{

public partial class AddEquipmentWindow : Window

{

private readonly AppDbContext \_context;

public AddEquipmentWindow()

{

InitializeComponent();

\_context = new AppDbContext();

}

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

{

var equipment = new Equipment

{

Name = NameTextBox.Text,

Model = ModelTextBox.Text,

SerialNumber = SerialNumberTextBox.Text,

Status = "Доступно"

};

\_context.Equipment.Add(equipment);

\_context.SaveChanges();

MessageBox.Show("Техника успешно добавлена!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

Close();

}

}

}

﻿using System.Windows;

using MachineAccounting.Models;

namespace MachineAccounting.Views

{

public partial class AddUserWindow : Window

{

private readonly AppDbContext \_context;

public AddUserWindow()

{

InitializeComponent();

\_context = new AppDbContext();

}

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

{

var user = new Users

{

FullName = FullNameTextBox.Text,

Username = UsernameTextBox.Text,

Password = PasswordBox.Password

};

\_context.Users.Add(user);

\_context.SaveChanges();

MessageBox.Show("Пользователь успешно добавлен!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

Close();

}

}

}

﻿using System.Linq;

using System.Windows;

using Machine\_Accounting.Models;

using MachineAccounting.Models;

namespace MachineAccounting.Views

{

public partial class AssignUserWindow : Window

{

private readonly MachineAccountingEntities \_context;

public AssignUserWindow(Machine\_Accounting.Models.Equipment selectedEquipment)

{

InitializeComponent();

// Инициализация контекста базы данных

\_context = new MachineAccountingEntities();

// Загрузка данных в ComboBox

LoadEquipmentData();

LoadUserData();

}

private void LoadEquipmentData()

{

// Загрузка списка доступной техники

var equipmentList = \_context.Equipment

.Where(e => e.Status == "Доступно") // Отображаем только доступную технику

.ToList();

EquipmentComboBox.ItemsSource = equipmentList;

EquipmentComboBox.DisplayMemberPath = "Name"; // Отображаем название техники

EquipmentComboBox.SelectedValuePath = "Id"; // Храним идентификатор

}

private void LoadUserData()

{

// Загрузка списка пользователей

var userList = \_context.Users.ToList();

UserComboBox.ItemsSource = userList;

UserComboBox.DisplayMemberPath = "FullName"; // Отображаем имя пользователя

UserComboBox.SelectedValuePath = "Id"; // Храним идентификатор

}

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

{

// Получение выбранных значений

var selectedEquipmentId = EquipmentComboBox.SelectedValue;

var selectedUserId = UserComboBox.SelectedValue;

if (selectedEquipmentId == null || selectedUserId == null)

{

MessageBox.Show("Выберите технику и пользователя.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

// Обновление статуса техники и закрепление за пользователем

var equipment = \_context.Equipment.Find(selectedEquipmentId);

var user = \_context.Users.Find(selectedUserId);

if (equipment != null && user != null)

{

equipment.Status = "Закреплено";

\_context.SaveChanges();

MessageBox.Show($"Техника \"{equipment.Name}\" успешно закреплена за пользователем \"{user.FullName}\".", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

Close();

}

else

{

MessageBox.Show("Ошибка при закреплении техники. Повторите попытку.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

// Закрытие окна

Close();

}

}

}

﻿using System.Windows;

using Microsoft.Win32;

using System.IO;

using Machine\_Accounting.Models;

namespace MachineAccounting.Views

{

public partial class EditProfileWindow : Window

{

private readonly Users \_currentUser;

private readonly MachineAccountingEntities \_context = new MachineAccountingEntities();

public EditProfileWindow(Users user)

{

InitializeComponent();

\_currentUser = user;

FullNameTextBox.Text = \_currentUser.FullName;

}

public EditProfileWindow()

{

}

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

{

// Сохранение изменений

\_currentUser.FullName = FullNameTextBox.Text;

\_context.Entry(\_currentUser).State = System.Data.Entity.EntityState.Modified;

\_context.SaveChanges();

MessageBox.Show("Профиль успешно обновлен!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

Close();

}

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

{

// Загрузка аватарки

OpenFileDialog openFileDialog = new OpenFileDialog

{

Filter = "Image Files|\*.jpg;\*.jpeg;\*.png"

};

if (openFileDialog.ShowDialog() == true)

{

byte[] avatar = File.ReadAllBytes(openFileDialog.FileName);

MessageBox.Show("Аватарка успешно изменена!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

}

}

﻿using System.Linq;

using System.Windows;

using MachineAccounting.Models;

namespace MachineAccounting.Views

{

public partial class EquipmentWindow : Window

{

private readonly AppDbContext \_context;

public EquipmentWindow()

{

InitializeComponent();

\_context = new AppDbContext();

LoadEquipment();

}

private void LoadEquipment()

{

EquipmentDataGrid.ItemsSource = \_context.Equipment.ToList();

}

}

}

﻿using System;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using Machine\_Accounting.Models;

namespace MachineAccounting.Views

{

public partial class ManageEquipmentWindow : Window

{

private readonly MachineAccountingEntities \_context;

public ManageEquipmentWindow()

{

InitializeComponent();

\_context = new MachineAccountingEntities();

LoadEquipment();

}

// Загрузка данных о технике в DataGrid

private void LoadEquipment()

{

EquipmentDataGrid.ItemsSource = \_context.Equipment.ToList();

}

// Добавление техники

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

{

var addEquipmentWindow = new AddEquipmentWindow();

if (addEquipmentWindow.ShowDialog() == true)

{

LoadEquipment(); // Обновить данные после добавления

}

}

// Удаление выбранной техники

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

{

if (EquipmentDataGrid.SelectedItem is Equipment selectedEquipment)

{

var result = MessageBox.Show($"Вы уверены, что хотите удалить технику '{selectedEquipment.Name}'?",

"Подтверждение удаления",

MessageBoxButton.YesNo,

MessageBoxImage.Question);

if (result == MessageBoxResult.Yes)

{

\_context.Equipment.Remove(selectedEquipment);

\_context.SaveChanges();

LoadEquipment(); // Обновить данные после удаления

MessageBox.Show("Техника успешно удалена.", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

else

{

MessageBox.Show("Выберите технику для удаления.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

// Изменение статуса техники

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

{

if (EquipmentDataGrid.SelectedItem is Equipment selectedEquipment)

{

var newStatus = selectedEquipment.Status == "Доступно" ? "Недоступно" : "Доступно";

selectedEquipment.Status = newStatus;

\_context.SaveChanges();

LoadEquipment(); // Обновить данные после изменения

MessageBox.Show($"Статус техники изменён на '{newStatus}'.", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

}

else

{

MessageBox.Show("Выберите технику для изменения статуса.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

// Закрепление техники за пользователем

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

{

if (EquipmentDataGrid.SelectedItem is Equipment selectedEquipment)

{

// Открываем окно выбора пользователя

var assignUserWindow = new AssignUserWindow(selectedEquipment);

if (assignUserWindow.ShowDialog() == true)

{

LoadEquipment(); // Обновить данные после закрепления

}

}

else

{

MessageBox.Show("Выберите технику для закрепления за пользователем.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

}

}