МИНИСТЕРСТВО ОБРАЗОВАНИЯ КИРОВСКОЙ ОБЛАСТИ

Кировское областное государственное профессиональное образовательное

бюджетное учреждение

«Слободской колледж педагогики и социальных отношений»

**ОТЧЕТ**

**по производственной практике**

**ПМ01. Разработка модулей программного обеспечения для компьютерных систем**

**Тема: «Разработка программного модуля для учета оборудования»**

Студент

Смагин Егор Максимович

Группа 21П-1

Специальность 09.02.07 Информационные системы и программирование

Руководитель практики от колледжа:

*Калинин Арсений Олегович*

подпись

УТВЕРЖДАЮ:

Директор

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 Наименование организации

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

подпись расшифровка

М. П.

2025 уч. год

**Содержание**

1. Сведения об организации (базе практики)
2. Анализ предметной области
3. Разработка API
4. Разработка библиотеки
5. Разработка настольного приложения
6. Разработка мобильного приложения
7. Тестирование
8. Рефакторинг кода
9. Заключение
10. Приложения
11. Сведения об организации (базе практики)

Производственную практику я проходил в КОГПОБУ «Слободской колледж педагогики и социальных отношений». Это Кировское областное государственное профессиональное образовательное бюджетное учреждение. Он расположен по адресу: Основной целью Учреждения является осуществление образовательной деятельности по образовательным программам среднего профессионального образования для подготовки квалифицированных рабочих, служащих и специалистов среднего звена в соответствии с потребностями личности, общества и государства, а также удовлетворение потребностей личности в углублении и расширении образования.

Ключевые элементы, которые обычно включаются в цель такого учреждения:

* + - 1. Подготовка кадров: Обучение студентов по программам СПО.
      2. Качество: Обеспечение высокого уровня образования.
      3. Востребованность: Соответствие выпускников требованиям рынка труда (регионального).
      4. Компетенции: Формирование профессиональных и общих навыков.
      5. Развитие личности: Воспитание и личностный рост студентов.



Рисунок 1 – Фотография учреждения

1. Анализ предметной области

Введение и Цель Системы

Целью данной системы является автоматизация процессов учета, отслеживания и управления жизненным циклом оборудования внутри организации. Система предназначена для замены ручного учета (например, в Excel-таблицах или на бумаге) или для улучшения существующей неполной системы, предоставляя централизованное хранилище данных, контроль доступа и возможности отчетности.

Основные задачи:

1.Ведение актуального реестра всего оборудования.

2.Отслеживание местонахождения каждой единицы оборудования.

3.Отслеживание ответственного пользователя (кому назначено оборудование).

4.Фиксация статуса оборудования (на складе, в использовании, в ремонте, списано).

5.Генерация отчетов по оборудованию.

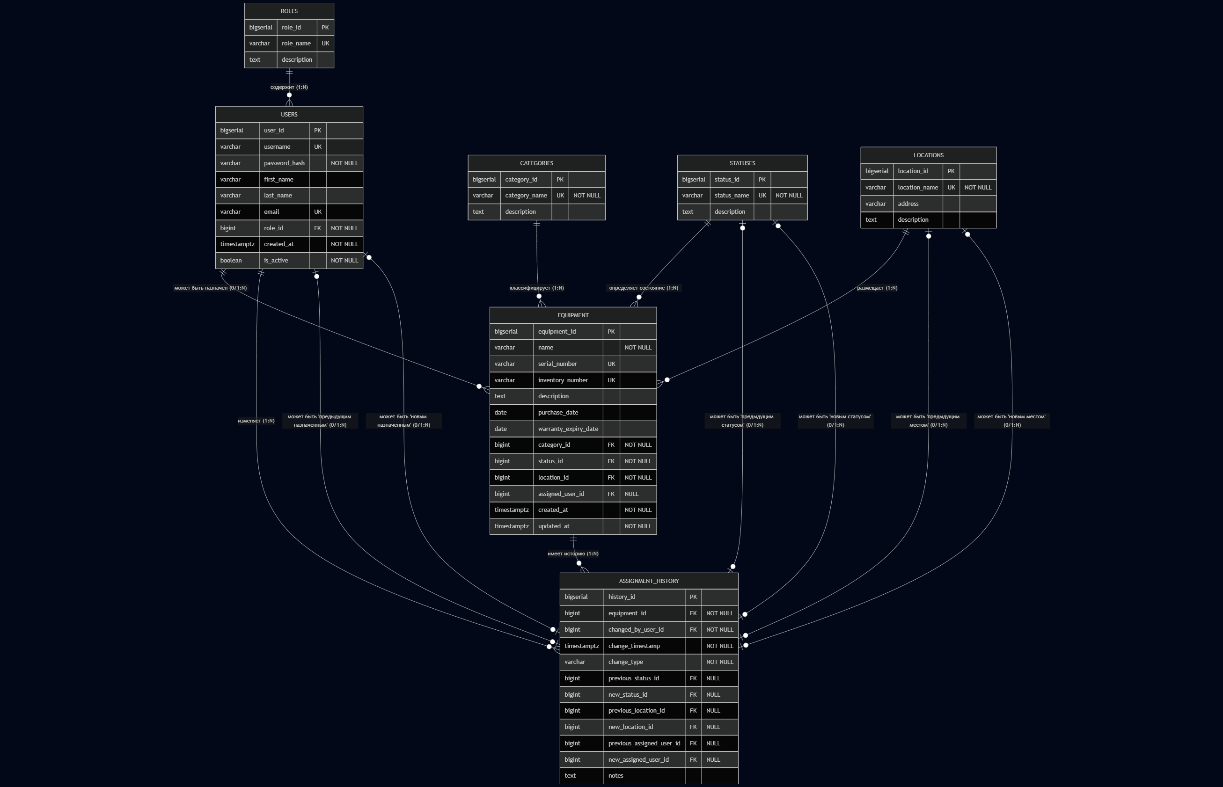


Рисунок – ERD диграмма

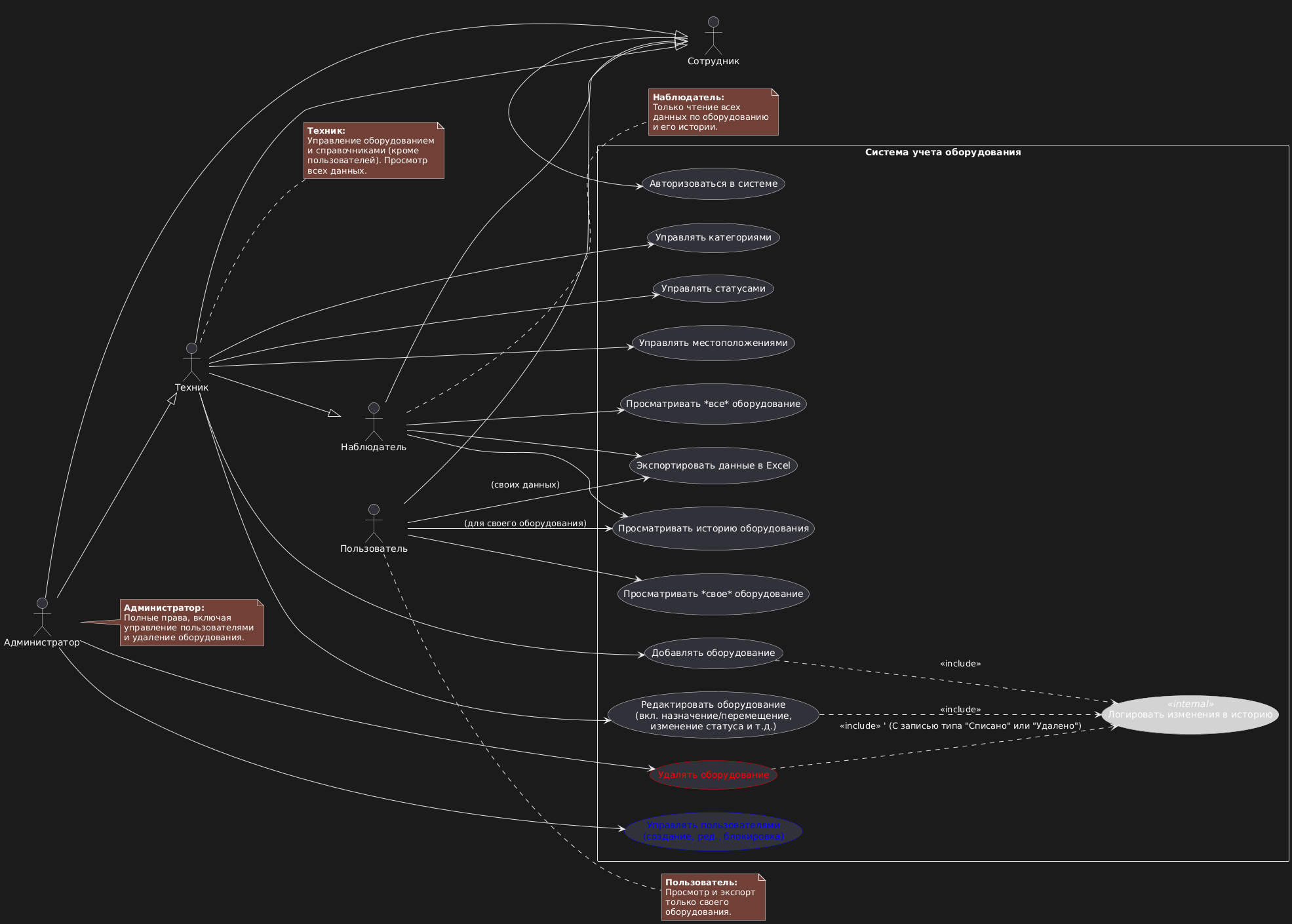


Рисунок – Диаграмма вариантов использования

1. Разработка API

Во время производственной практики был осуществлен полный цикл разработки серверной инфраструктуры на платформе Supabase, включающий проектирование и реализацию API для обеспечения взаимодействия клиентской части приложения с базой данных. В рамках этой работы были спроектированы и созданы необходимые таблицы для эффективного хранения и управления данными о пользователях системы с учетом различных ролей, подробной информации о оборудовании (Рисунок 2). Разработанная структура базы данных обеспечивает оптимальную производительность при выполнении запросов, поддерживает целостность данных и включает необходимые индексы для ускорения поиска информации.

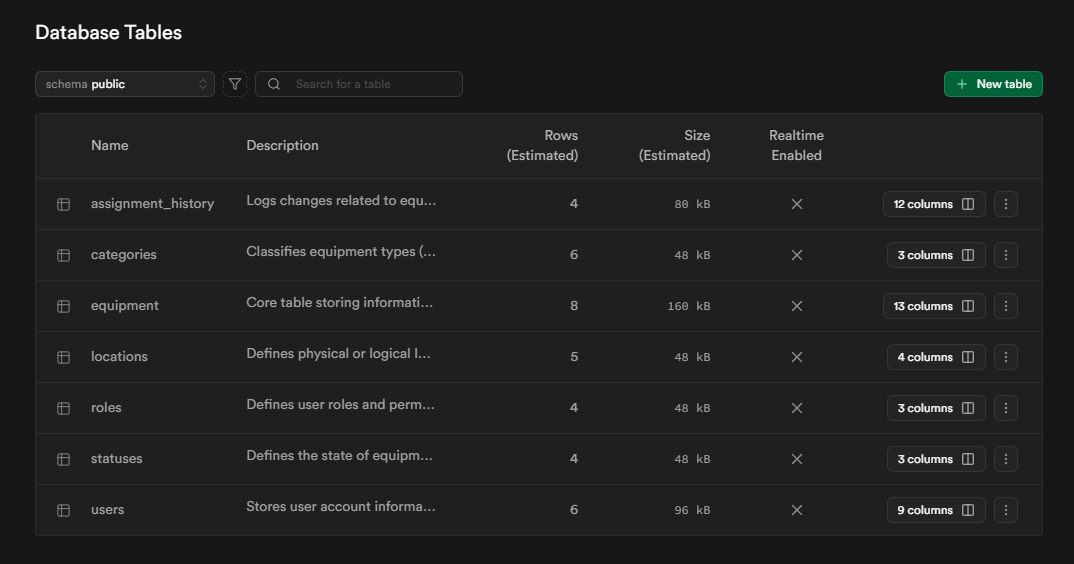


Рисунок 2 – База данных на supabase

1. Создание библиотеки

Создана .NET-библиотека, которая представляет собой утилиту для генерации отчета о распределении оборудования по местоположениям. Оно напрямую подключается к указанной базе данных Supabase. Основываясь на данных из предварительно созданного SQL-представления equipment\_count\_by\_location, приложение выводит в консоль отформатированный список всех зарегистрированных местоположений с указанием точного количества единиц оборудования, числящихся за каждым из них (Рисунок 3). Это позволяет быстро получить сводную картину без необходимости запуска основного графического приложения.

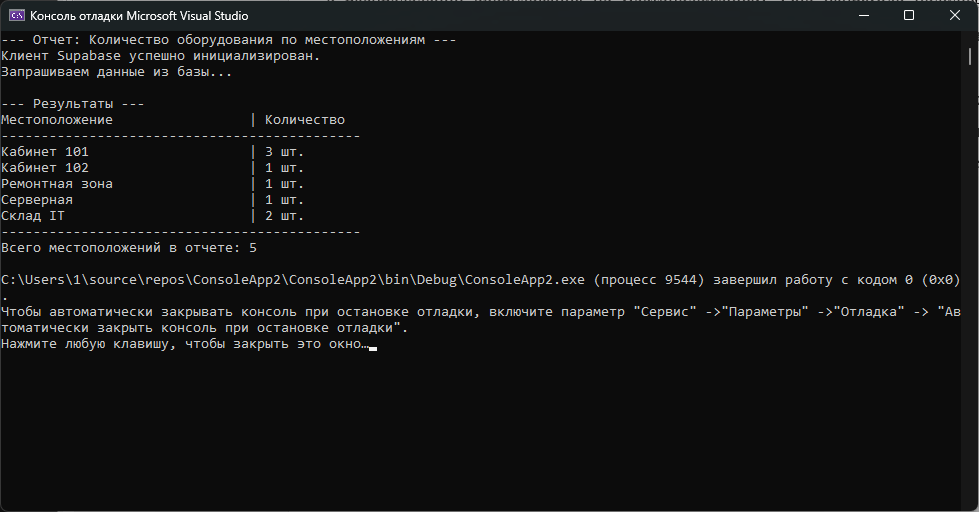


Рисунок 3 - Библиотека

1. Создание настольного приложения

Основной частью практической работы стала разработка настольного приложения на платформе Windows Forms с использованием языка C#. Приложение реализует основной пользовательский интерфейс для взаимодействия с системой учета оборудования. Ключевые реализованные функции:

Авторизация: Окно входа в систему, где пользователь вводит логин и пароль. Данные сверяются с записями в таблице users базы данных Supabase. (Хеширование паролей реализовано на базовом уровне или используется прямая строка для учебных целей).

Интерфейс пользователя: Главное окно отображает список оборудования в компоненте DataGridView с возможностью сортировки и базового поиска/фильтрации.

Ролевая модель: Функциональность приложения (доступность кнопок добавления, редактирования, удаления, доступа к справочникам) динамически изменяется в зависимости от роли авторизованного пользователя (Admin, Technician, User, Viewer), полученной из базы данных. Пользователи с ролью User видят только назначенное им оборудование.

Экспорт данных: Реализована функция экспорта текущего списка оборудования (отображенного в DataGridView) в файл формата Excel (.xlsx) с использованием библиотеки EPPlus.

Интерфейс приложения спроектирован с учетом простоты и удобства использования для целевых пользователей.

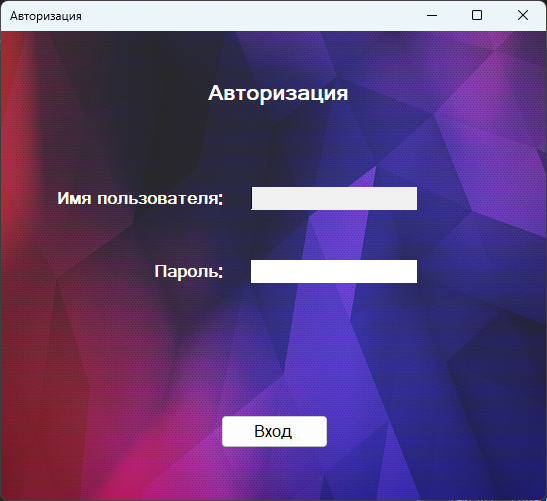


Рисунок 4 – Авторизация

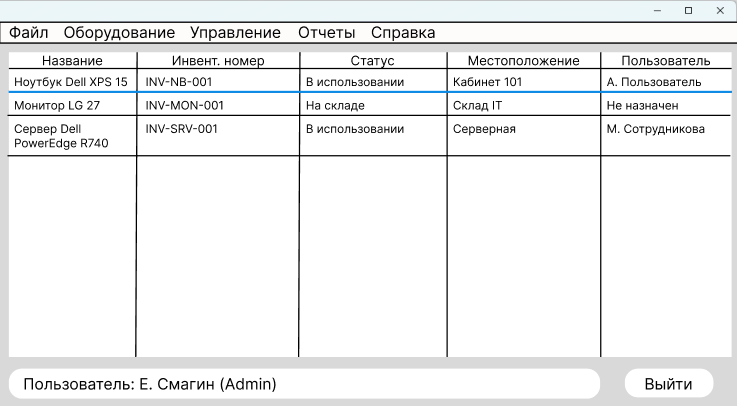


Рисунок 5 – Главное окно приложения

1. Создание мобильного приложения

Создано мобильное приложение на Xamarin.Forms для Android, в котором можно просматривать информацию о оборудовании (Рисунок 6).

Мобильное приложение синхронизировано с API и использует общую библиотеку логики.



Рисунок 6 – Мобильное приложение

1. Тестирование

В процессе разработки проводилось функциональное (ручное) тестирование приложения для обеспечения его корректной работы и надежности. Основные тестовые сценарии включали:

Тестирование авторизации: Проверка входа под учетными записями с разными ролями (Admin, Technician, User, Viewer) и проверка невозможности входа с неверными данными.

Тестирование управления данными: Проверка корректности добавления нового оборудования, сохранения изменений при редактировании, корректной работы выпадающих списков.

Тестирование ролевой модели: Проверка отображения данных и доступности функций (кнопки, пункты меню) в соответствии с правами каждой роли. Особое внимание уделялось проверке фильтрации данных для роли User.

Тестирование истории: Проверка создания записей в таблице assignment\_history после выполнения операций редактирования оборудования.

Тестирование экспорта: Проверка корректности создания Excel-файла и соответствия данных в нем отображаемым в приложении.

В ходе тестирования были выявлены и устранены мелкие ошибки, связанные с обработкой пользовательского ввода и отображением данных. Основной функционал приложения работает в соответствии с требованиями.

1. Рефакторинг кода

На завершающем этапе практики был проведен рефакторинг кода настольного приложения с целью улучшения его качества, читаемости и сопровождаемости:

Улучшение структуры: Код был разбит на более мелкие логические методы для повышения модульности.

Обработка исключений: Добавлены блоки try-catch для обработки потенциальных ошибок при взаимодействии с базой данных Supabase (например, сетевые ошибки, ошибки доступа).

Комментарии: Добавлены комментарии к ключевым блокам кода и сложным методам для пояснения их работы.

Именование: Проверена консистентность и ясность именования переменных, методов и элементов управления.

Оптимизация запросов: Хотя основная оптимизация достигается использованием представлений и индексов в БД, была проверена логика запросов на стороне клиента на предмет избыточности.

Проведенный рефакторинг позволил сделать код более понятным и подготовленным к возможному дальнейшему расширению функционала.

1. Заключение

В ходе производственной практики была успешно разработана система учета оборудования организации. Проект включает в себя спроектированную и реализованную базу данных на платформе Supabase, основное настольное приложение на Windows Forms (C#) для управления данными с учетом ролей пользователей, а также вспомогательное консольное приложение для генерации отчетов.

**ПРИЛОЖЕНИЯ**

Приложение 1

По ссылке, расположенной ниже, вы можете скачать и ознакомиться с программным продуктом. После перехода по ссылке откроется доступ к исполняемым файлам, исходному коду и документации, что позволит проверить программный продукт на практике. https://github.com/Koni1ak/proizvpm01.git

Приложение 2

Техническое Задание

На разработку программного обеспечения «Система Учета Оборудования Колледжа»

Версия: 1.0

Дата: 21.03.2025

1. Введение

1.1. Назначение документа: Настоящее Техническое Задание (ТЗ) определяет требования к разработке программного обеспечения (ПО) «Система Учета Оборудования Колледжа». ПО предназначено для автоматизации процессов учета, добавления, редактирования и отслеживания состояния оборудования в рамках образовательного учреждения.

1.2. Заказчик: КОГПОБУ СКПиСО

1.3. Исполнитель: Смагин Егор Максимович

1.4. Основание для разработки: Внутренняя потребность колледжа в систематизации учета IT-оборудования.

1.5. Краткое описание системы: Настольное приложение для платформы Windows, использующее графический интерфейс Windows Forms и подключающееся к облачной базе данных Supabase (PostgreSQL) для хранения и извлечения данных об оборудовании, его статусах, местоположениях.

2. Термины и определения

Система / ПО: Программное обеспечение «Система Учета Оборудования Колледжа».

Оборудование: Любая единица IT-техники (ноутбук, монитор, МФУ и т.д.), подлежащая учету.

Пользователь: Сотрудник отдела ИТ или другое ответственное лицо, использующее Систему для управления учетом оборудования. (В текущей версии не предусмотрено разделение ролей в интерфейсе).

Supabase: Облачная платформа "Backend-as-a-Service", предоставляющая базу данных PostgreSQL и API для взаимодействия с ней.

WinForms: Библиотека .NET для создания настольных приложений Windows с графическим интерфейсом.

История изменений: Журнал операций, связанных с конкретной единицей оборудования (создание, изменение статуса, местоположения, пользователя, удаление).

3. Назначение и цели создания системы

3.1. Назначение: Автоматизация и централизация учета IT-оборудования колледжа.

3.2. Цели:

Предоставление актуальной информации о наличии, состоянии и местонахождении оборудования.

Упрощение процесса инвентаризации.

Обеспечение возможности добавления, редактирования и удаления записей об оборудовании.

Ведение истории основных изменений по каждой единице оборудования.

Снижение риска потери оборудования и ошибок ручного учета.

4. Требования к системе

4.1. Функциональные требования:

4.1.1. Просмотр оборудования:

Система должна отображать список всего оборудования в табличном виде (DataGridView).

Таблица должна содержать как минимум следующие столбцы: ID, Название, Инвентарный номер, Серийный номер, Категория (название), Статус (название), Местоположение (название), Назначенный пользователь (ФИО или "Не назначен").

Должна быть реализована возможность обновления списка данных из базы данных по нажатию кнопки "Обновить".

4.1.2. Добавление оборудования:

Система должна предоставлять возможность добавления новой записи об оборудовании через отдельную форму (EquipmentForm).

Форма добавления должна позволять ввести/выбрать следующие данные: Название (обязательно), Серийный номер, Инвентарный номер, Описание, Дата покупки (опционально), Дата окончания гарантии (опционально), Категория (обязательно, выбор из справочника), Статус (обязательно, выбор из справочника), Местоположение (обязательно, выбор из справочника), Назначенный пользователь (опционально, выбор из справочника пользователей с возможностью выбрать "Не назначен").

При сохранении новой записи должна выполняться базовая валидация на заполнение обязательных полей.

4.1.3. Редактирование оборудования:

Система должна предоставлять возможность редактирования данных существующей единицы оборудования.

Доступ к редактированию должен осуществляться по выбору строки в таблице и нажатию кнопки "Редактировать" (или по двойному клику). Кнопка должна быть активна только при выборе ровно одной строки.

Редактирование должно происходить в той же форме (EquipmentForm), что и добавление, с предзаполненными текущими данными оборудования.

Должна быть предусмотрена возможность оставить заметку о причине изменения.

4.1.4. Удаление оборудования:

Система должна предоставлять возможность удаления записи об оборудовании.

Доступ к удалению должен осуществляться по выбору строки в таблице и нажатию кнопки "Удалить". Кнопка должна быть активна только при выборе ровно одной строки.

Перед удалением система должна запрашивать подтверждение у пользователя.

Каждая запись истории должна содержать ID оборудования, ID пользователя, выполнившего действие (в текущей реализации используется ID=1), временную метку, тип изменения и опциональные поля для старых/новых значений и заметок.

4.2. Нефункциональные требования:

4.2.1. Производительность:

Время загрузки основного списка оборудования при запуске или обновлении не должно превышать 5 секунд при типичном объеме данных (до 1000 записей) и стабильном интернет-соединении.

Открытие формы добавления/редактирования должно происходить не более чем за 3 секунды (включая загрузку справочников).

Операции сохранения и удаления должны выполняться не более чем за 3 секунды.

4.2.2. Надежность:

Система должна стабильно работать в целевой ОС Windows.

Должна быть реализована корректная обработка ошибок при взаимодействии с базой данных (например, отсутствие сети, ошибки Supabase API) с выводом информативных сообщений пользователю.

Должны использоваться механизмы предотвращения потери данных (например, подтверждение перед удалением).

4.2.3. Удобство использования (Usability):

Интерфейс должен быть интуитивно понятным, соответствовать стандартам Windows Desktop приложений.

Элементы управления должны быть логично сгруппированы.

Должна присутствовать обратная связь при выполнении длительных операций (например, курсор ожидания, статусная строка).

Сообщения об ошибках должны быть понятны пользователю.

4.2.4. Безопасность:

В текущей версии (1.0): Аутентификация пользователей не реализована. Доступ к базе данных осуществляется через публичный ключ Supabase (Anon Key). Все операции записи истории выполняются от имени предопределенного пользователя (ID=1).

Предполагается, что доступ к приложению имеют только доверенные сотрудники.

(Требование к будущим версиям): Реализовать систему аутентификации пользователей и разграничение прав доступа на основе ролей с использованием Row Level Security (RLS) в Supabase.

Приложение 3

1. Введение

Добро пожаловать в Руководство пользователя системы учета оборудования колледжа!

Это программное обеспечение предназначено для сотрудников отдела ИТ и других ответственных лиц колледжа для ведения централизованного учета компьютерного и другого IT-оборудования. Система позволяет просматривать, добавлять, редактировать и удалять информацию об оборудовании, а также отслеживать основные изменения, связанные с ним.

Данные хранятся в централизованной базе данных (Supabase), поэтому для работы программы требуется подключение к сети Интернет.

2. Системные требования

Операционная система: Windows 10 или Windows 11.

Программное обеспечение: Установленный .NET Runtime (обычно устанавливается вместе с приложением или операционной системой).

Сеть: Активное и стабильное подключение к сети Интернет.

3. Начало работы

Найдите исполняемый файл приложения (обычно с расширением .exe, например, EquipmentAccounting.exe) или ярлык на рабочем столе / в меню "Пуск".

Дважды щелкните по файлу или ярлыку для запуска приложения.

При первом запуске может потребоваться некоторое время для инициализации подключения к базе данных.

4. Обзор основного окна

После запуска откроется окно авторизации:

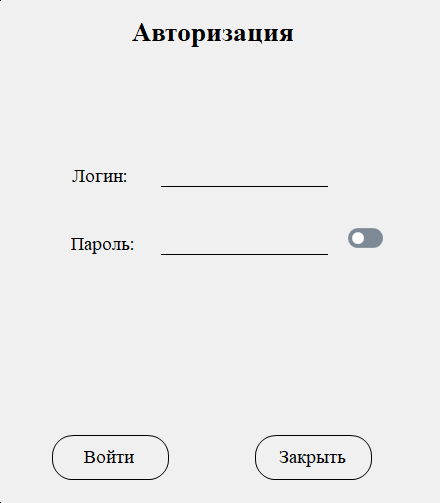


Рисунок – Окно авторизации

После успешной авторизации открывается главное окно:

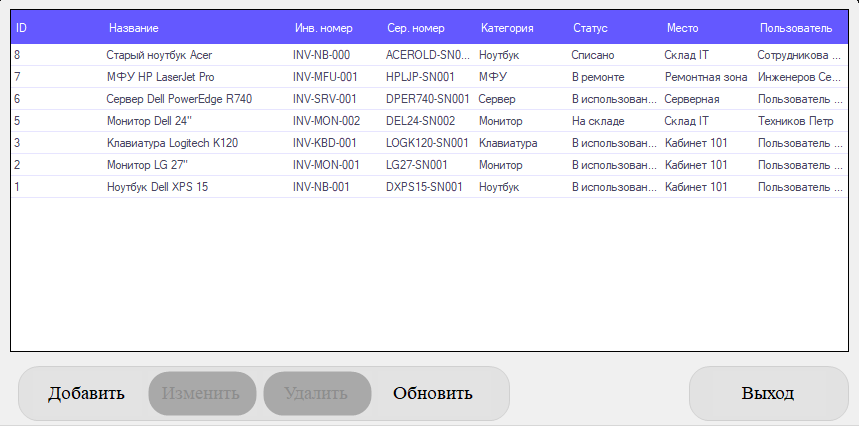


Рисунок – Главное окно

Таблица оборудования: Основную часть окна занимает таблица (список), в которой отображается информация об оборудовании. Основные столбцы:

ID: Уникальный идентификатор записи.

Название: Наименование оборудования.

Инв. номер: Инвентарный номер.

Сер. номер: Серийный номер.

Категория: Тип оборудования (например, "Ноутбук").

Статус: Текущее состояние (например, "В использовании").

Место: Физическое или логическое расположение.

Пользователь: Сотрудник, за которым закреплено оборудование, или "Не назначен".

Кнопки управления (под таблицей):

Добавить: Открывает форму для ввода данных о новом оборудовании.

Редактировать: Открывает форму для изменения данных выбранного в таблице оборудования (активна, только если выбрана ровно одна строка).

Удалить: Удаляет выбранную запись об оборудовании (активна, только если выбрана ровно одна строка). Требует подтверждения.

Обновить: Загружает актуальные данные из базы данных и обновляет таблицу.

Строка состояния (внизу окна): Отображает служебную информацию, такую как статус загрузки данных или количество отображаемых записей.

5. Основные операции

5.1. Просмотр списка оборудования:

При запуске приложения список загружается автоматически.

Для получения самых свежих данных нажмите кнопку "Обновить".

Вы можете прокручивать список с помощью полосы прокрутки.

Для сортировки данных щелкните по заголовку нужного столбца (повторный щелчок изменит направление сортировки). Примечание: Возможность сортировки зависит от настроек таблицы.

5.2. Добавление нового оборудования:

Нажмите кнопку "Добавить". Откроется форма "Добавить новое оборудование".

Заполните поля на форме:

Название\*: Обязательное поле. Введите понятное наименование (например, "Ноутбук Dell Latitude 5420").

Инвентарный номер: Введите уникальный инвентарный номер (если есть).

Серийный номер: Введите уникальный серийный номер (если есть).

Категория\*: Обязательное поле. Выберите тип оборудования из выпадающего списка.

Статус\*: Обязательное поле. Выберите текущий статус из выпадающего списка.

Местоположение\*: Обязательное поле. Выберите местонахождение из выпадающего списка.

Пользователь: Выберите сотрудника из выпадающего списка, за которым будет закреплено оборудование. Оставьте "Не назначен", если оборудование на складе или не закреплено.

Дата покупки: Установите галочку и выберите дату из календаря, если она известна. Снимите галочку, если дата неизвестна.

Гарантия до: Установите галочку и выберите дату окончания гарантии из календаря, если она известна. Снимите галочку, если дата неизвестна.

Описание: Введите дополнительную информацию или характеристики.

После заполнения полей нажмите кнопку "Сохранить".

Если все обязательные поля заполнены корректно, форма закроется, и новая запись появится в общей таблице (может потребоваться обновление списка).

5.3. Редактирование данных оборудования:

В основной таблице выберите ровно одну строку с оборудованием, которое вы хотите отредактировать.

Кнопка "Редактировать" станет активной. Нажмите ее (или дважды щелкните по выбранной строке).

Откроется форма "Редактировать: [Название оборудования]" с текущими данными.

Внесите необходимые изменения в поля формы.

Заметка (причина изм.): Если вы вносите существенные изменения (например, меняете статус, пользователя или местоположение), рекомендуется кратко описать причину в поле "Заметка". Эта информация сохранится в истории.

Нажмите кнопку "Сохранить".

Форма закроется, и изменения отобразятся в общей таблице (может потребоваться обновление списка).

5.4. Удаление оборудования:

В основной таблице выберите ровно одну строку с оборудованием, которое вы хотите удалить.

Кнопка "Удалить" станет активной. Нажмите ее.

Появится диалоговое окно с запросом подтверждения удаления. Внимательно проверьте, то ли оборудование вы выбрали.

Для подтверждения удаления нажмите кнопку "Да". Для отмены – "Нет".

При подтверждении запись будет удалена из системы, и информация об этом будет сохранена в истории. Запись исчезнет из таблицы. Внимание! Операция удаления необратима.

6. Возможные проблемы и решения

Ошибка "Не удалось инициализировать подключение к базе данных" при запуске:

Проверьте ваше подключение к сети Интернет.

Убедитесь, что нет проблем с доступом к сервису Supabase (возможно, временные технические работы).

Обратитесь к системному администратору или разработчику.

Ошибка "Ошибка загрузки данных" / Список пуст, хотя оборудование есть:

Нажмите кнопку "Обновить".

Проверьте ваше подключение к сети Интернет.

Если ошибка повторяется, обратитесь к системному администратору или разработчику, возможно, проблема в базе данных или API.

Кнопки "Редактировать" или "Удалить" неактивны (серые):

Убедитесь, что вы выбрали ровно одну строку в таблице. Эти кнопки активируются только при выборе одной записи.

Убедитесь, что данные в данный момент не загружаются (курсор не в виде песочных часов, в строке состояния нет сообщения о загрузке).

Приложение работает медленно:

Возможно, медленное интернет-соединение.

Большое количество записей в базе данных может замедлять загрузку.