

Краевое государственное автономное
профессиональное образовательное учреждение
«Пермский авиационный техникум им. А.Д. Швецова»

ДНЕВНИК
О ПРОХОЖДЕНИИ УЧЕБНОЙ ПРАКТИКИ

специальность 09.02.07 Информационные системы и программирование

Студент Шишигина Вера Антоновна

Группа ИСВ-22-3-22 Специальность 09.02.07 Информационные системы и
программирование

Квалификация Веб-разработчик

Место практики Краевое государственное автономное профессиональное
образовательное учреждение «Пермский авиационный техникум им. А. Д.
Швецова»

(Наименование организации)

Начало практики 05.05.2025

Окончание
практики _____

Руководитель практики от техникума Суслوнова М.Л. Ощепков А. О.

Дата	Содержание работы	Подпись
05.05.25	<p>Бизнес-процессы. Анализ предметной области подразумевает изучение всех аспектов деятельности организации, связанных с управлением оборудованием и расходными материалами. SADT— это методология структурного анализа и проектирования, используемая для моделирования бизнес-процессов. IDEF0 — это одна из нотаций в рамках методологии IDEF, предназначенная для функционального моделирования процессов. Декомпозиция до уровня отдельных процедур означает разбиение крупных процессов на более мелкие, детализированные шаги. Техническое задание должно содержать четкие требования к системе, включая функциональные и нефункциональные требования, описание архитектуры, интерфейсов, безопасности и других аспектов.</p>	
06.05.25	<p>Диаграмма вариантов использования. Спецификация системы. Диаграмма вариантов использования (Use Case) в UML отображает взаимодействие между акторами (пользователями) и системой. В данном случае акторы — это администратор, преподаватель и сотрудник.</p>	
07.05.25	<p>Диаграмма классов - используется для моделирования данных в базе данных. Сущности в ERD соответствуют классам в диаграмме классов, а атрибуты сущностей — атрибутам классов. Отношения между сущностями (один-ко-многим, многие-ко-многим) также должны отражаться в диаграмме классов через ассоциации. Нужно убедиться, что каждое поле в ERD имеет соответствующий атрибут в классе и наоборот. Соответствие диаграммы классов и ERD - она показывает, как объект изменяет свои состояния в ответ на события. Например, объект "Оборудование" может иметь состояния: "Доступно", "На ремонте", "Списано". Переходы между состояниями могут быть вызваны событиями, такими как "Начать ремонт", "Завершить ремонт", "Списать". Диаграмма состояний объектов. Также стоит обратить внимание на отношения. Например, если в ERD есть отношение между "Оборудование" и "Инвентаризация" как "один-ко-многим", то в диаграмме классов это может быть представлено как ассоциация с кратностью.</p>	

12.05.25	Диаграммы деятельности на каждый вариант использования системы (прецедент).	
13.05.25	Диаграмма последовательности на каждый вариант использования системы (прецедент).	
14.05.25	Диаграмма кооперации взаимодействующих между собой объектов информационной системы. Руководство пользователя	

Краевое государственное автономное
профессиональное образовательное учреждение
«Пермский авиационный техникум им. А.Д. Швецова»

ОТЧЕТ
О ПРОХОЖДЕНИИ УЧЕБНОЙ ПРАКТИКИ
СПЕЦИАЛЬНОСТЬ 09.02.07 Информационные системы и программирование

Студент Шишигина Вера Антоновна

Группа ИСВ-22-3-22 Специальность 09.02.07 Информационные системы и
программирование

Квалификация Веб-разработчик

Место практики Краевое государственное автономное профессиональное
образовательное учреждение «Пермский авиационный техникум им. А. Д.
Швецова»

(Наименование организации)

Начало практики 05.05.2025

Окончание
практики _____

Руководитель практики от техникума Суслوнова М.Л. Ощепков А. О.

Содержание

Занятие 1	6
Занятие 2	13
Занятие 3.....	16
Занятие 4.....	17
Занятие 5.....	23
Занятие 6.....	24
РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ.....	26
Занятие 1.....	26

					АТПР.09.02.07.22.07									
Изм	Лист	№ документа	Подпись	Дата	Учебная практика				Литера			Лист	Листов	
Разраб.		М.М. Махмутова							У		5	41		
Проверил								ИСВ-22-3					
Н. контр.														
Утв.														

Занятие 1

Предметная область: Управление оборудованием и расходными материалами в учебном заведении.

Задание 1. Определите состав бизнес-процессов для своей предметной области, подлежащих автоматизации, и кратко их опишите. Составить организационную диаграмму:

```
graph TD; A[Главный исполнительный директор] --> B[Финансовый директор]; A --> C[Технический директор]; A --> D[Директор по маркетингу]; A --> E[Директор по логистике]; A --> F[Директор по инновациям]; A --> G[Директор по качеству]; A --> H[Отдел кадров]; H --> I[Управление производством]; H --> J[Управление ключевыми клиентами]; H --> K[Оптимизация процессов]; H --> L[Управление данными]; H --> M[Управление закупками]; H --> N[Автоматизация отчетности];
```

Управление оборудованием:

1. Учет оборудования (добавление, удаление, редактирование).
2. Перемещение оборудования между аудиториями.
3. Назначение ответственных пользователей.
4. Учет статуса оборудования (на ремонте, используется, сломано и т.д.).
5. Генерация отчетов по оборудованию.

Управление аудиториями:

1. Учет аудиторий (добавление, удаление, редактирование).
2. Назначение ответственных пользователей за аудитории.

Управление расходными материалами:

1. Учет расходных материалов (добавление, удаление, редактирование).
2. Учёт, привязка к оборудованию.

3. Учет ответственных пользователей за расходные материалы.

Инвентаризация:

1. Планирование, проведение, фиксация результатов.
2. Комментарии к состоянию оборудования
3. Генерация отчетов по инвентаризации.

Управление пользователями:

1. Учет пользователей (добавление, удаление, редактирование).
2. Привязка к оборудованию и аудиториям.

Управление сетевыми настройками:

1. Учет сетевых настроек оборудования.
2. Проверка сетевых настроек.

Генерация документов:

Создание актов приема-передачи оборудования и расходных материалов.

Задание 2. Постройте контекстную диаграмму предметной области с подробными пояснениями

Ошибки в примере диаграммы и ее декомпозиции

1. Не указаны механизмы, которые используются для выполнения процесса.
2. Входные и выходные данные не четко разделены.
3. Отсутствует описание управляющих действий

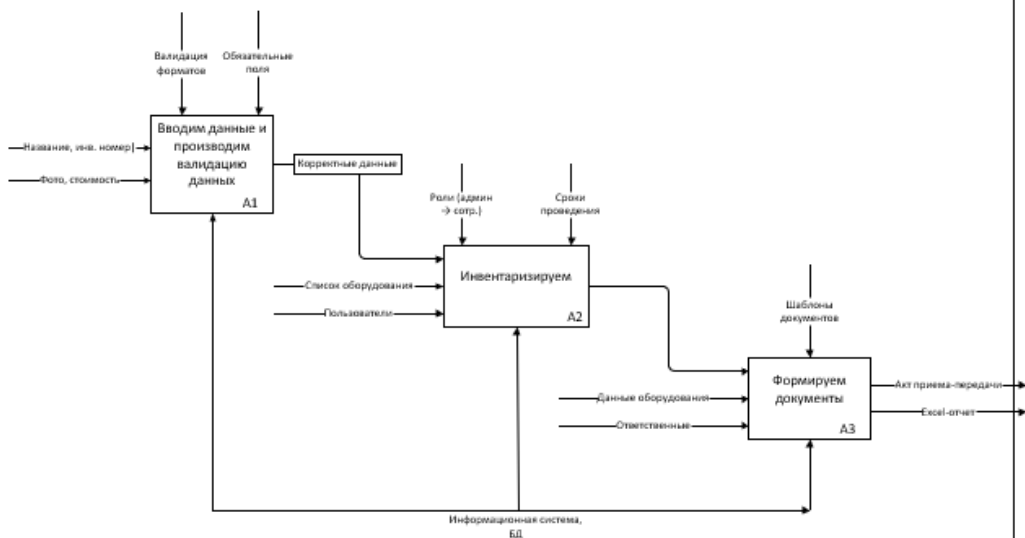
Задание 3. Декомпозируйте контекстную диаграмму предметной области на один уровень. Определите блоки, требующие дальнейшей декомпозиции.

Цель: провести анализ данных об оборудовании и расходных материалах учебного заведения, содержащихся в базе.

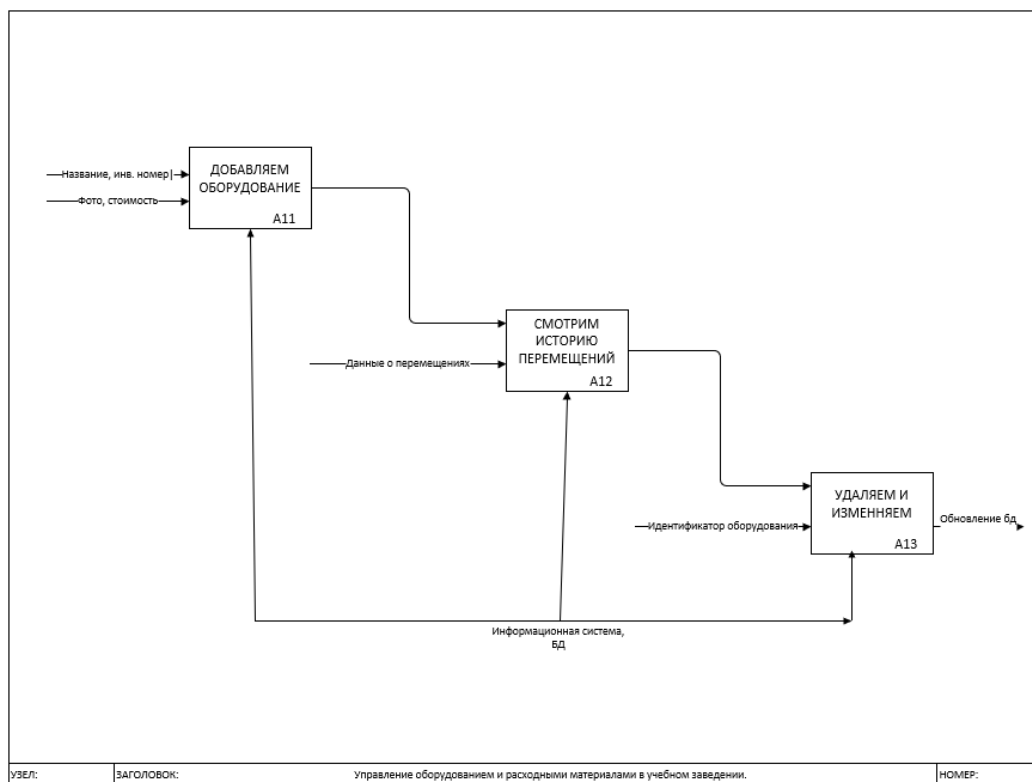
Точка зрения: Администратор информационной системы управления оборудованием учебного заведения.



УЗЕЛ: ЗАГЛОВОК: Управление оборудованием и расходными материалами в учебном заведении. НОМЕР:



УЗЕЛ: ЗАГЛОВОК: Управление оборудованием и расходными материалами в учебном заведении. НОМЕР:



1. Назначение системы

Разработка программного обеспечения для автоматизации учета, контроля и управления оборудованием, аудиториями, расходными материалами и пользователями в учебном заведении.

2. Функциональные требования

2.1 Основные модули

Управление оборудованием

Добавление, редактирование, удаление данных.

Привязка к аудиториям и ответственным.

История перемещений и изменений.

Валидация: инвентарный номер (только цифры), стоимость (числовой формат).

Учёт аудиторий

Наименование, ответственные, временные пользователи.

Поиск и сортировка.

Управление пользователями

Регистрация, роли (администратор, преподаватель, сотрудник).

Привязка к оборудованию и аудиториям.

Инвентаризация

Планирование, проведение, фиксация результатов.

Комментарии к состоянию оборудования.

Расходные материалы

Учёт, привязка к оборудованию.

Характеристики (цвет, тип, объём).

Сетевые настройки

Управление IP-адресами, проверка уникальности.

Документооборот

Генерация актов приёма-передачи (примеры в Приложениях 1-3 ТЗ).

2.2 Дополнительные функции

Импорт/экспорт данных (Excel, CSV).

Резервное копирование и восстановление.

Поиск и фильтрация по всем модулям.

3. Технические требования

3.1 Архитектура

Тип: Модульное веб-приложение.

Backend: C# (ASP.NET Core), Python/Django (на выбор).

Frontend: HTML5, CSS3, JavaScript (React/Vue.js).

База данных: MySQL или MS SQL Server.

API: RESTful для интеграции с внешними сервисами.

3.2 Безопасность

Аутентификация: логин/пароль, хеширование (SHA-256).

Авторизация: ролевая модель (JWT-токены).

Валидация данных на стороне сервера и клиента.

3.3 Интерфейсы

Пользовательский интерфейс:

Цветовая схема: серый (#9EA1A2), красный (#E41613), синий (#0060AC).

Адаптивный дизайн для ПК и планшетов.

Внешние интеграции:

Экспорт отчётов в PDF/Excel.

Загрузка изображений оборудования (JPG, PNG).

3.4 Ограничения

Поддержка только современных браузеров (Chrome, Firefox, Edge).

Максимальный размер загружаемых изображений: 5 МБ.

4. Нефункциональные требования

4.1 Производительность

Время отклика: ≤ 2 сек для стандартных операций.

Поддержка 100+ одновременных пользователей.

4.2 Надёжность

Резервное копирование данных раз в сутки.

Логирование ошибок (хранение логов 30 дней).

4.3 Документация

Руководство пользователя (HTML/PDF).

Техническая документация для разработчиков (Swagger, комментарии в коде).

5. Примеры использования

Сценарий 1: Преподаватель отмечает оборудование в инвентаризации.

Сценарий 2: Администратор генерирует акт приёма-передачи ноутбука.

Приложения:

Макеты интерфейсов.

Примеры SQL-запросов.

Диаграммы (Use Case, ERD).

Контрольные вопросы:

1. Укажите, сколько уровней имеет иерархическая структура для вашей предметной области.

Иерархическая структура имеет 3 уровня: общий уровень
детализации процессов.

2. Укажите обозначения для структурных единиц и для исполнителей

Структурные единицы: блоки (A1, A2, A3, A4).

Исполнители: пользователи системы (администраторы, преподаватели, сотрудники).

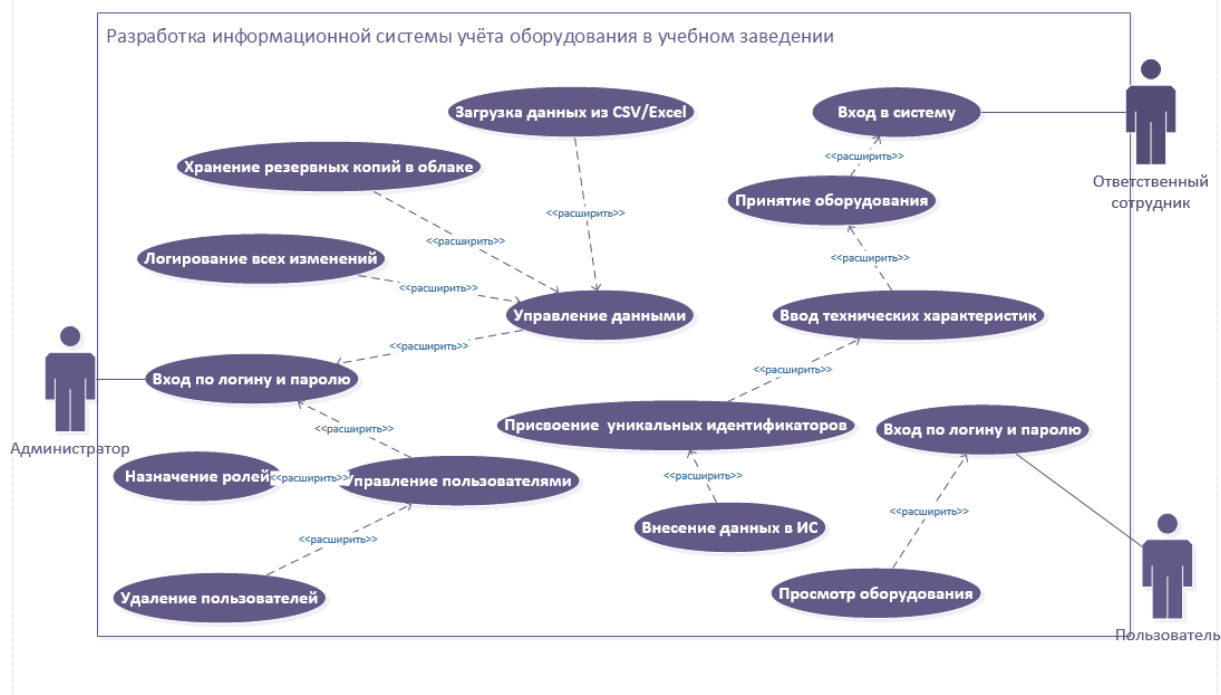
3. Укажите, сколько уровней декомпозиции потребовалось для полного понимания процессов.

Потребовалось 1 уровня декомпозиции: общий уровень и уровень детализации процессов.

Вывод: В ходе выполнения задания была проведена декомпозиция процесса управления оборудованием и расходными материалами в учебном заведении с использованием нотации IDEF0. Основной процесс был разделен на четыре ключевых подпроцесса: учет оборудования, учет расходных материалов, инвентаризация и генерация документов. Каждый подпроцесс был детализирован с учетом входов, управляющих действий, механизмов и выходов.

Занятие 2

Диаграмма прецедентов



Спецификация информационной системы управления оборудованием учебного заведения

1. Общее описание системы

Название: информационная система учета оборудования

Назначение:

1. Учет, управление и контроль оборудования, аудиторий, расходных материалов и пользователей в учебном заведении.
2. Автоматизация документооборота (акты приема-передачи, инвентаризация).
3. Обеспечение прозрачности перемещения оборудования и ответственных лиц.

Целевая аудитория:

Администраторы – полный доступ ко всем функциям.

Ответственный сотрудник – просмотр, участие в инвентаризации.

2. Функциональные модули

4. Ограничения и требования

Категория	Требования
Браузеры	Chrome, Firefox
Разрешения экрана	1920x1080 (Desktop)

5. Этапы тестирования

1. **Unit-тесты** – проверка изолированного функционала (API, валидация).
2. **Интеграция** – тестирование связей между модулями.
3. **Интерфейс** – корректность отображения и логики UI.
4. **Нагрузка** – имитация работы системы при 100+ пользователей.

6. Дополнительные требования:

Документирование кода: Реализация XML-описаний для всех элементов системы с целью стандартизации и упрощения поддержки.

Управление версиями: Применение Git (Gogs) с ветками, соответствующими этапам жизненного цикла разработки.

Мониторинг событий: Интеграция email-оповещений для уведомления о критических операциях в режиме реального времени.

Вывод:

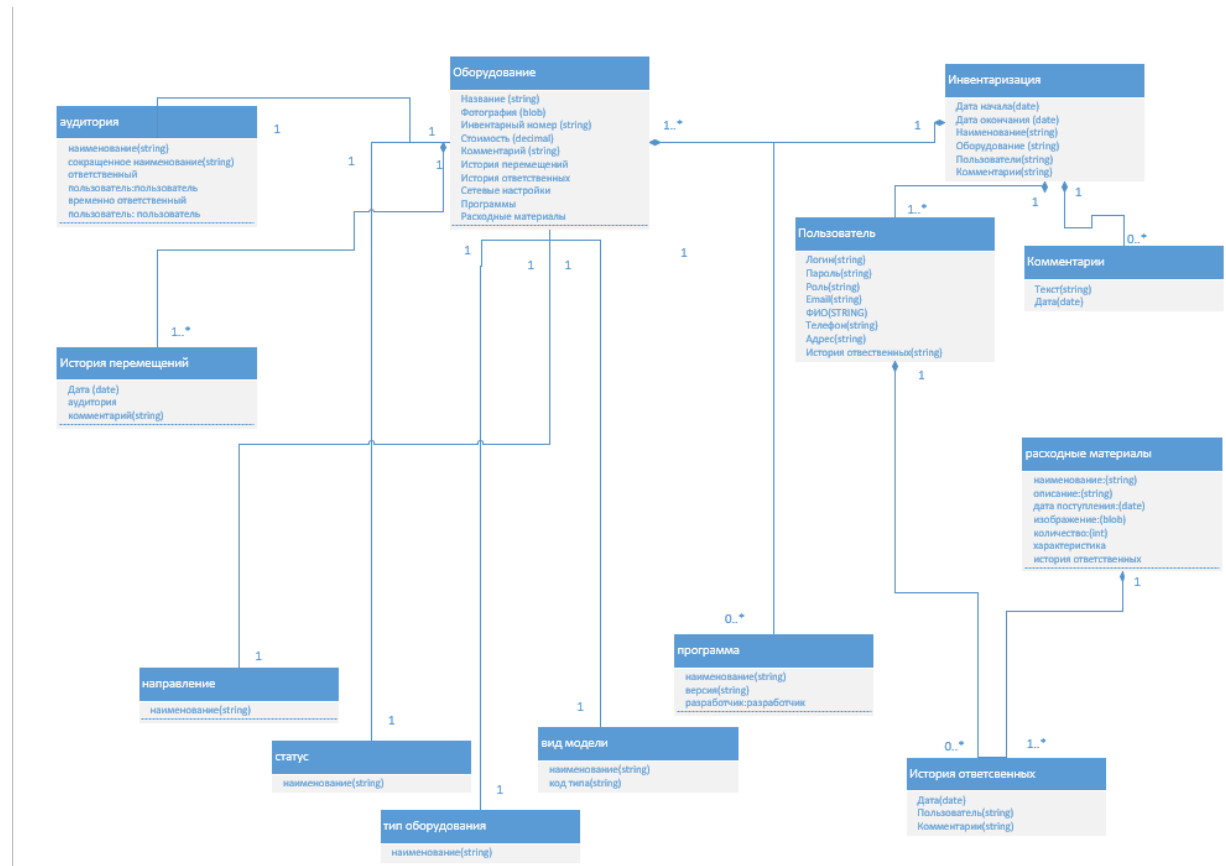
Спецификация детализирует функционал, архитектуру и требования к системе. Для реализации необходимо:

1. Согласовать дизайн интерфейса (макеты).
2. Подготовить тестовые данные (оборудование, пользователи).

Занятие 3.

Диаграмма классов. Соответствие диаграммы классов и ERD.

Диаграмма состояний объектов.



ERD

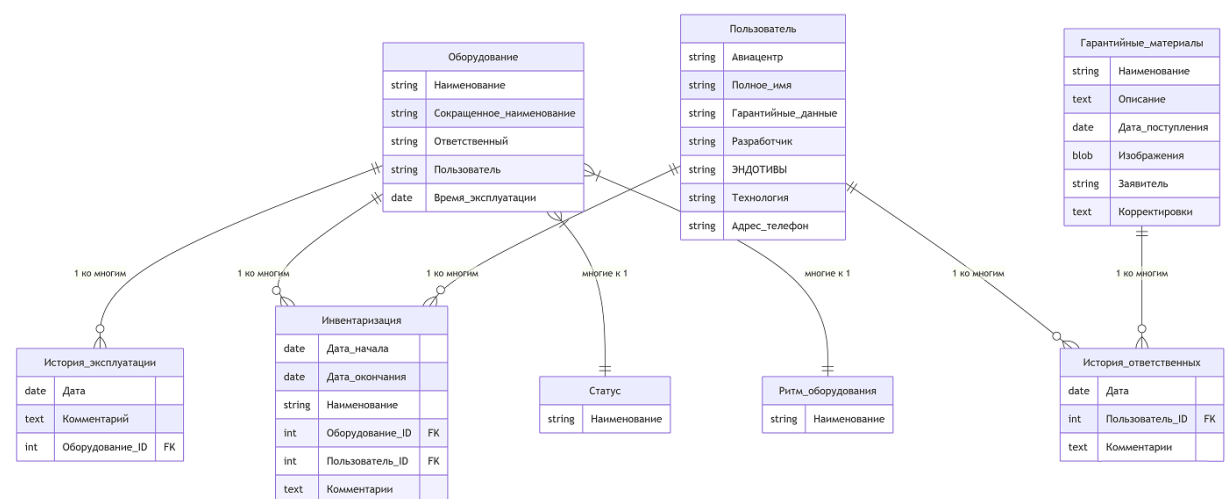
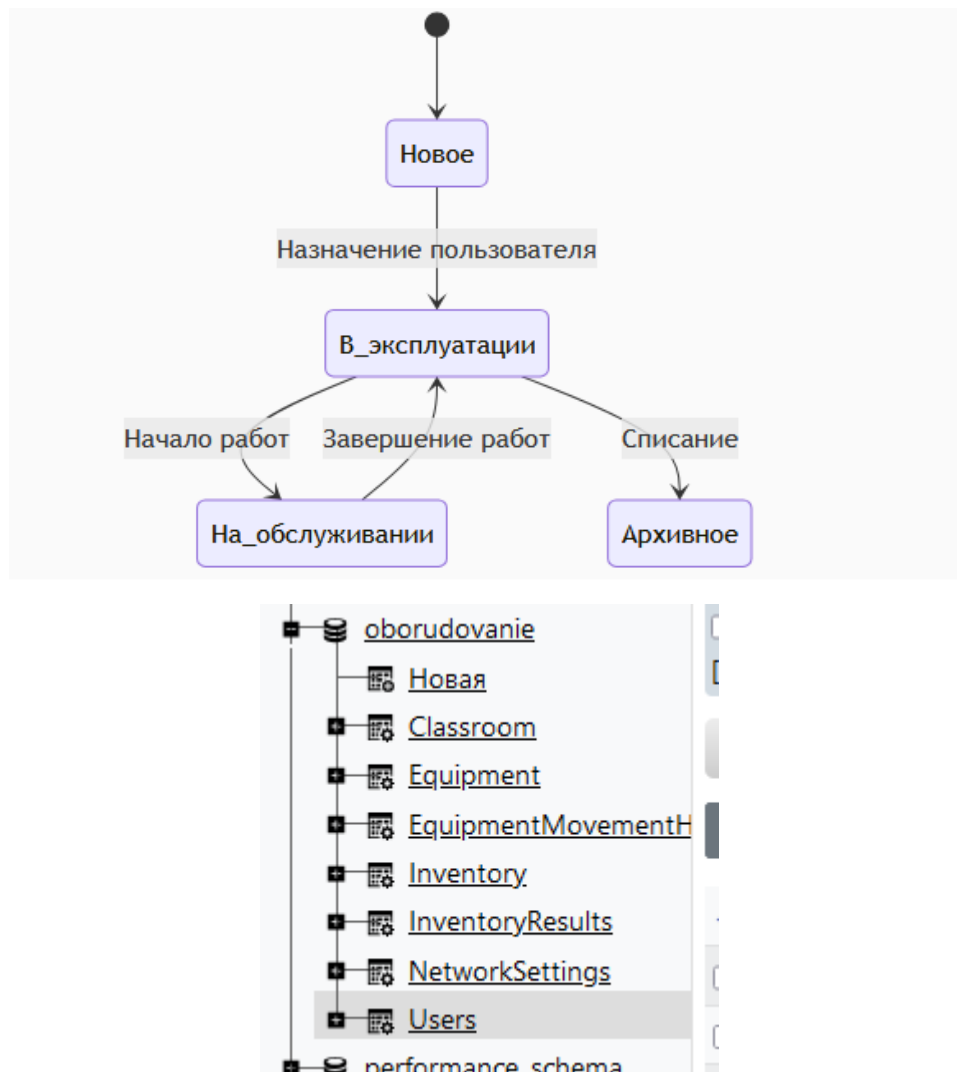


Диаграмма состояний объектов



Занятие 4.

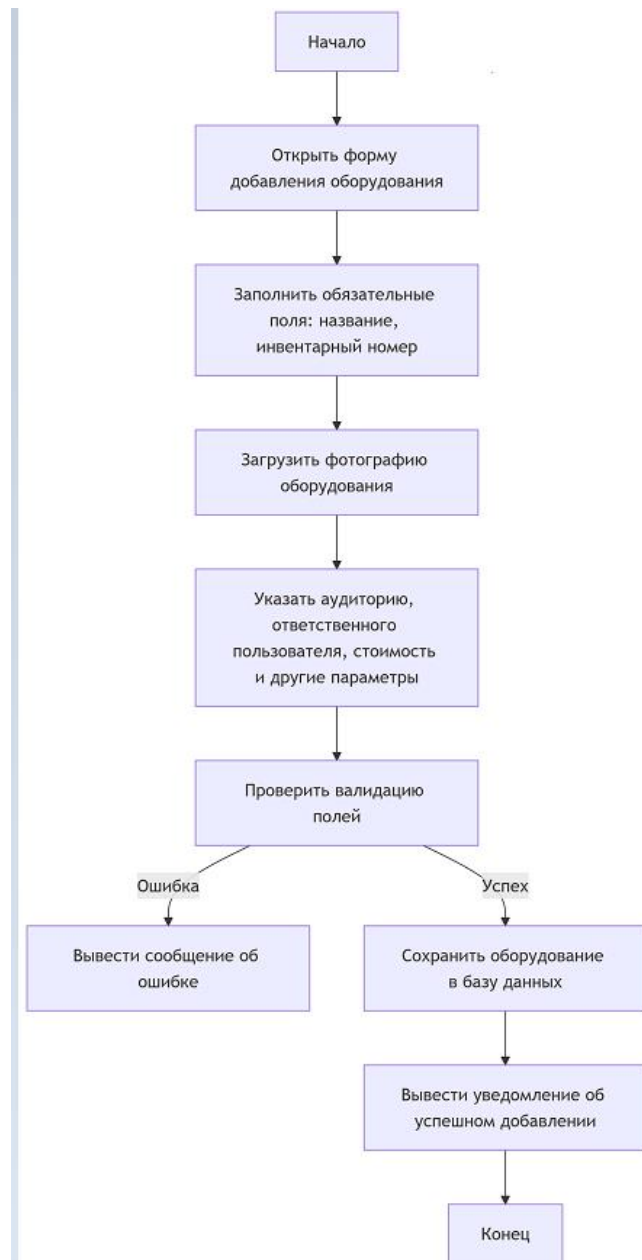
Ниже представлены диаграммы деятельности для ключевых прецедентов системы управления оборудованием учебного заведения. Диаграммы визуализируют последовательность действий пользователей и системы.

1. Добавление оборудования

Актеры: Администратор

Цель: добавить новое оборудование в систему.

Диаграмма деятельности:

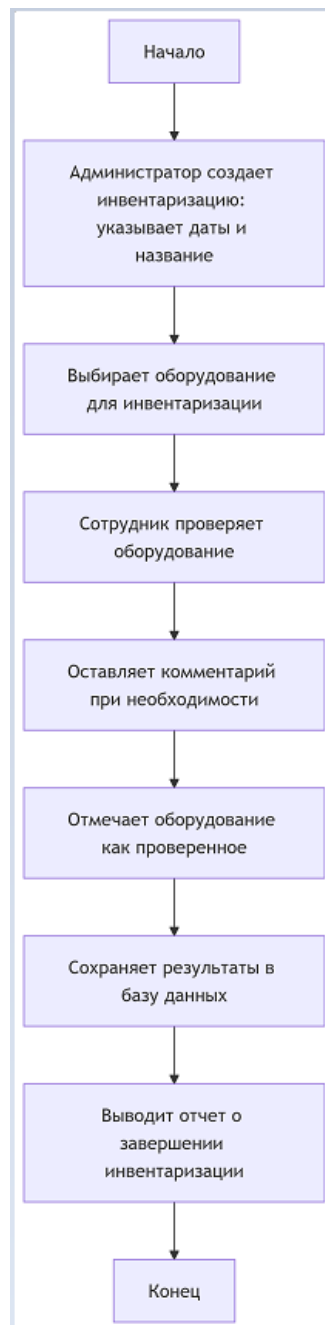


2. Проведение инвентаризации

Актеры: Администратор, Сотрудник

Цель: провести инвентаризацию оборудования.

Диаграмма деятельности:



3. Передача оборудования на временное пользование

Актеры: Администратор, Сотрудник

Цель: оформить передачу оборудования сотруднику.

Диаграмма деятельности:

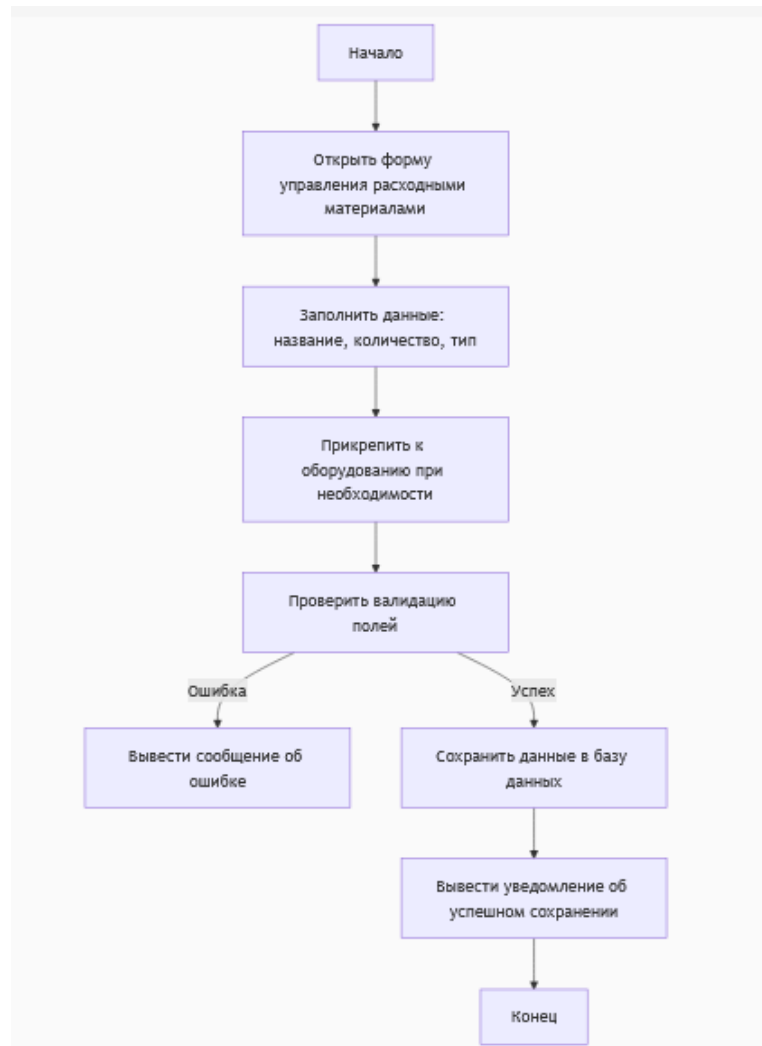


4. Управление расходными материалами

Актеры: Администратор

Цель: добавить или изменить расходные материалы.

Диаграмма деятельности:



5. Проверка сетевых настроек оборудования

Актеры: Администратор

Цель: проверить уникальность IP-адресов и доступность устройств.

Диаграмма деятельности:

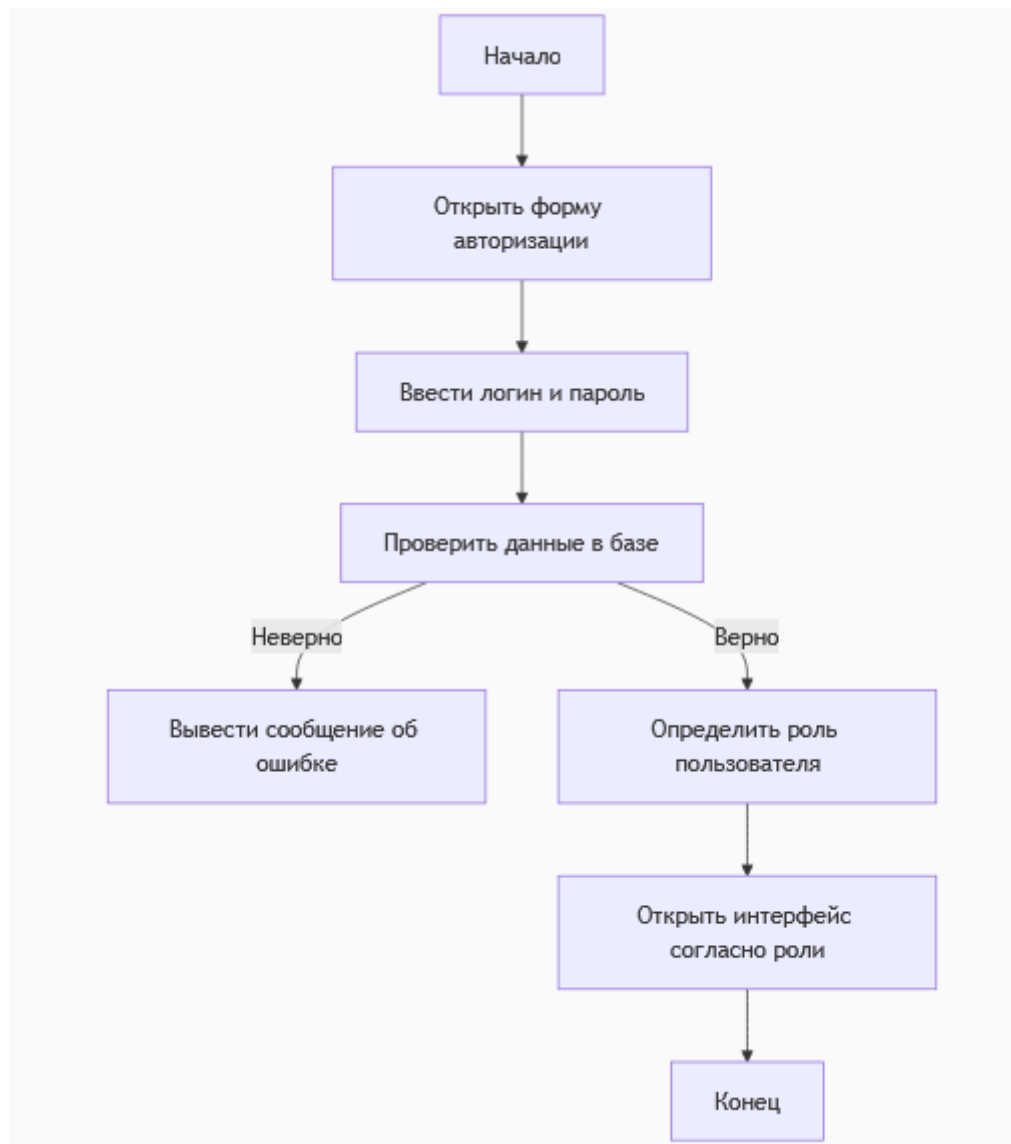


6. Авторизация пользователя

Актеры: Пользователь

Цель: войти в систему.

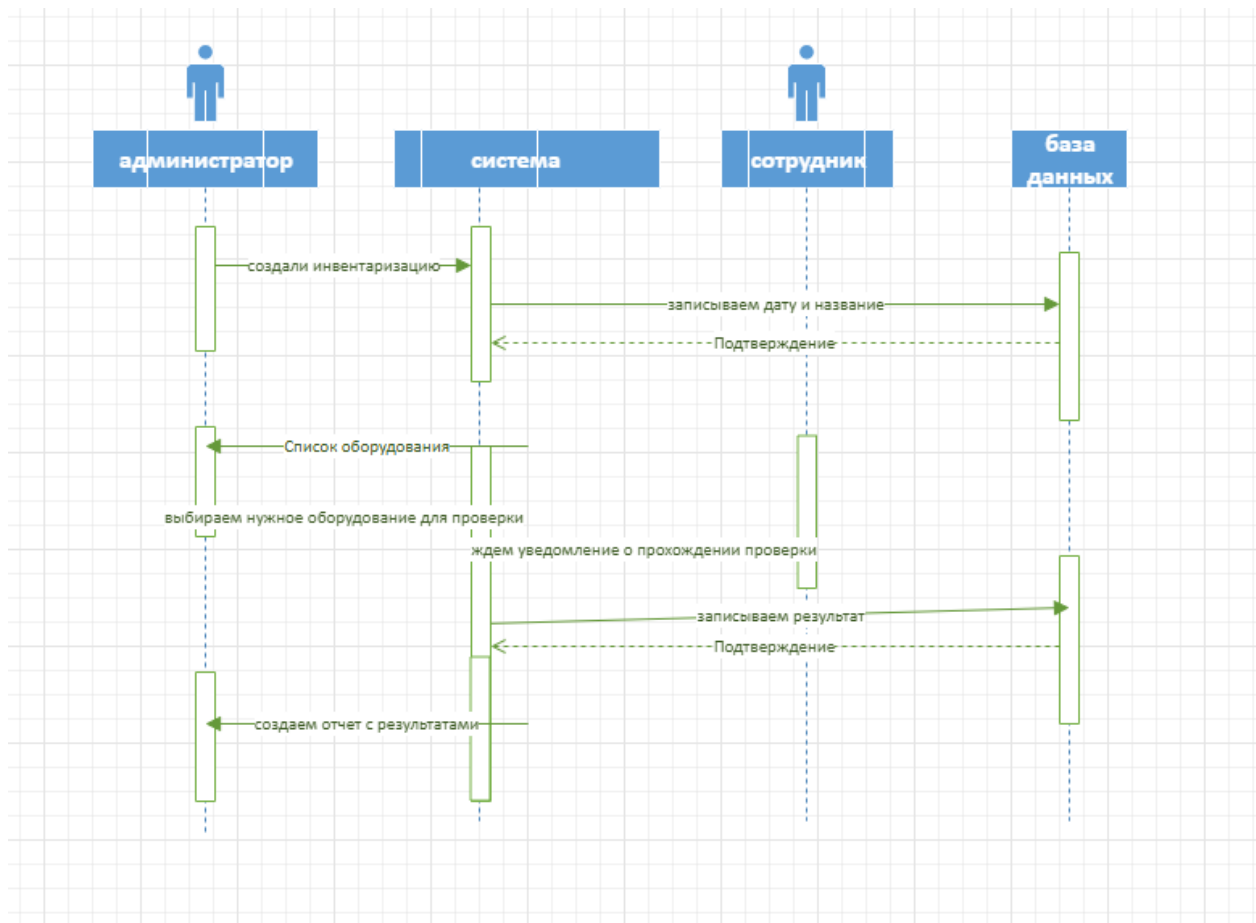
Диаграмма деятельности:



Занятие 5.

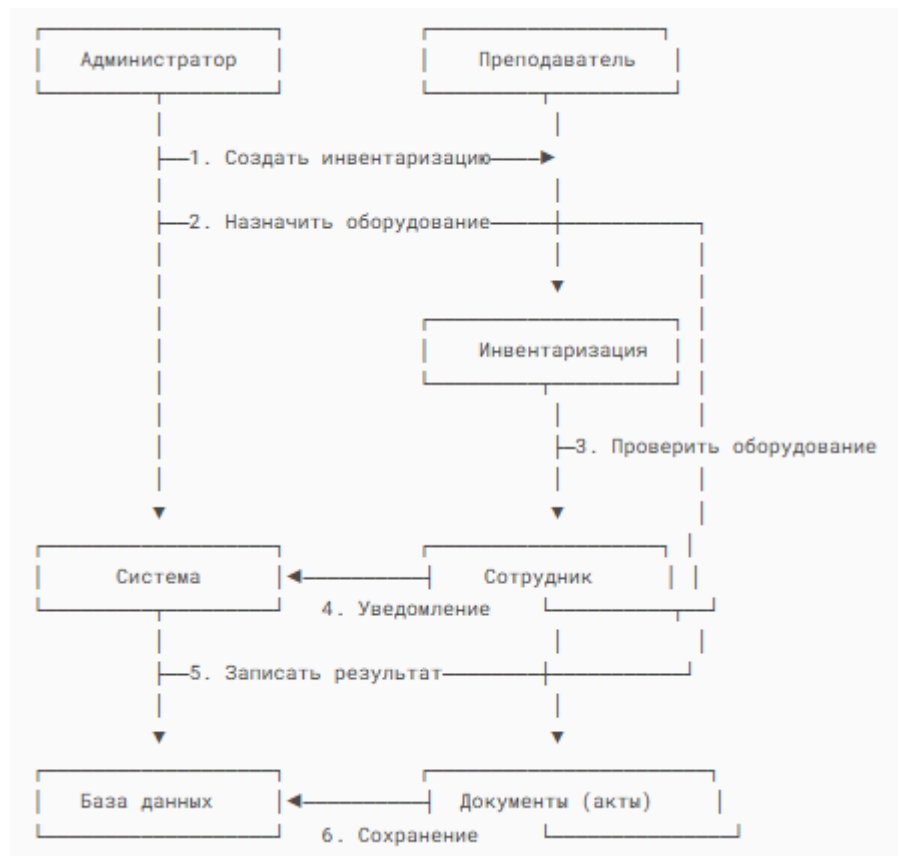
Диаграмма последовательности на каждый вариант использования системы (прецедент).

Ниже представлена диаграмма последовательности на каждый вариант использования системы.



Занятие 6.

Диаграмма кооперации взаимодействующих между собой объектов информационной системы. Руководство пользователя



Взаимодействия с другими объектами:

Оборудование

Добавление/изменение (проверка валидации)

Привязка к аудитории и пользователю

Связь с расходными материалами

Аудитория

Назначение ответственных

История перемещения оборудования

Сетевые настройки

Настройка IP-адресов (валидация)

Проверка уникальности адресов

Программы

Учет версий и разработчиков

Фильтрация по ПО

Расходные материалы

Привязка к оборудованию

Учет характеристик (цвет, тип и т.д.)

Руководство пользователя (краткая версия):

Авторизация:

Введите логин, пароль (скрыт символами *).

Управление оборудованием:

Добавление: укажите название, инвентарный номер, фото (jpg/png).

Поиск: используйте фильтры по названию или типу.

Инвентаризация:

Администратор: создает задание, выбирает оборудование.

Сотрудник: отмечает статус оборудования, оставляет комментарии.

Сетевые настройки:

Введите IP-адрес в формате XXX.XXX.XXX.XXX (0-255).

Система проверит уникальность адреса.

Документы:

Автоматическая генерация актов приема-передачи (см. Приложения).

Ошибки:

При удалении связанных данных система выдаст предупреждение.

Все ошибки сохраняются в лог (доступны разработчикам).

РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ

Занятие 1.

1 Проектирование базы данных

- Определение сущностей и их атрибутов
- Проектирование схемы (таблицы, связи, индексы)
- Нормализация структуры (1NF, 2NF, 3NF)
- Выбор типа БД (SQL: PostgreSQL, MySQL / NoSQL: MongoDB)

Сущности и их атрибуты

Основные сущности системы:

Оборудование - основная сущность системы

Аудитории - места размещения оборудования

Направления - цели приобретения оборудования

Статусы - текущее состояние оборудования

Типы оборудования - категории оборудования

Модели оборудования - конкретные модели

Программы - ПО, установленное на оборудовании

Разработчики ПО - производители программ

Инвентаризации - учетные мероприятия

Пользователи - сотрудники учебного заведения

Сетевые настройки - параметры сетевого оборудования

Расходные материалы - сопутствующие материалы

Типы расходных материалов - категории расходников

Характеристики расходных материалов - специфические параметры

Все таблицы приведены к 3NF, использовали **MySQL**

Создаем бд и заполняем таблицами.

Часть таблиц создавала Мунира, оставшиеся таблицы создавала Вера.

```
✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,3983 сек.)

CREATE TABLE Users ( id INT AUTO_INCREMENT PRIMARY KEY, username VARCHAR(50) NOT NULL, password VARCHAR(255) NOT NULL, role ENUM('admin', 'teacher', 'employee') NOT NULL, email VARCHAR(100), last_name VARCHAR(50) NOT NULL, first_name VARCHAR(50), middle_name VARCHAR(50), phone VARCHAR(20), address TEXT, created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP, updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, UNIQUE(username) );
[ Построение редактирование ] [ Изменить ] [ Создать PHP-код ]

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,5935 сек.)

CREATE TABLE Rooms ( id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NULL, short_name VARCHAR(20), responsible_user_id INT, temp_responsible_user_id INT, created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP, updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, FOREIGN KEY (responsible_user_id) REFERENCES Users(id), FOREIGN KEY (temp_responsible_user_id) REFERENCES Users(id) );
[ Построение редактирование ] [ Изменить ] [ Создать PHP-код ]
```

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0.6131 сек.)

```
CREATE TABLE Equipment ( id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NULL, photo LONGBLOB, inventory_number VARCHAR(50) NOT NULL, room_id INT, current_user_id INT, temp_user_id INT, cost DECIMAL(10,2), direction_id INT, status_id INT, model_id INT, comment TEXT, created_at TIMESTAMPT DEFAULT CURRENT_TIMESTAMP, updated_at TIMESTAMPT DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, FOREIGN KEY (room_id) REFERENCES Rooms(id), FOREIGN KEY (current_user_id) REFERENCES Users(id), FOREIGN KEY (temp_user_id) REFERENCES Users(id), UNIQUE (inventory_number) );
```

[Построение редактирование](#) [Изменить](#) [Создать PHP-код](#)

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0.5649 сек.)

```
CREATE TABLE Consumables ( id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NULL, description TEXT, receipt_date DATE, image LONGBLOB, quantity INT, responsible_user_id INT, temp_responsible_user_id INT, consumable_type INT, created_at TIMESTAMPT DEFAULT CURRENT_TIMESTAMP, updated_at TIMESTAMPT DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, FOREIGN KEY (responsible_user_id) REFERENCES Users(id), FOREIGN KEY (temp_responsible_user_id) REFERENCES Users(id) );
```

[Построение редактирование](#) [Изменить](#) [Создать PHP-код](#)

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0.5854 сек.)

```
CREATE TABLE EquipmentHistory ( id INT AUTO_INCREMENT PRIMARY KEY, equipment_id INT NOT NULL, room_id INT, user_id INT, date_from DATE NOT NULL, date_to DATE, comment TEXT, created_at TIMESTAMPT DEFAULT CURRENT_TIMESTAMP, FOREIGN KEY (equipment_id) REFERENCES Equipment(id), FOREIGN KEY (room_id) REFERENCES Rooms(id), FOREIGN KEY (user_id) REFERENCES Users(id) );
```

[Построение редактирование](#) [Изменить](#) [Создать PHP-код](#)

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0.4419 сек.)

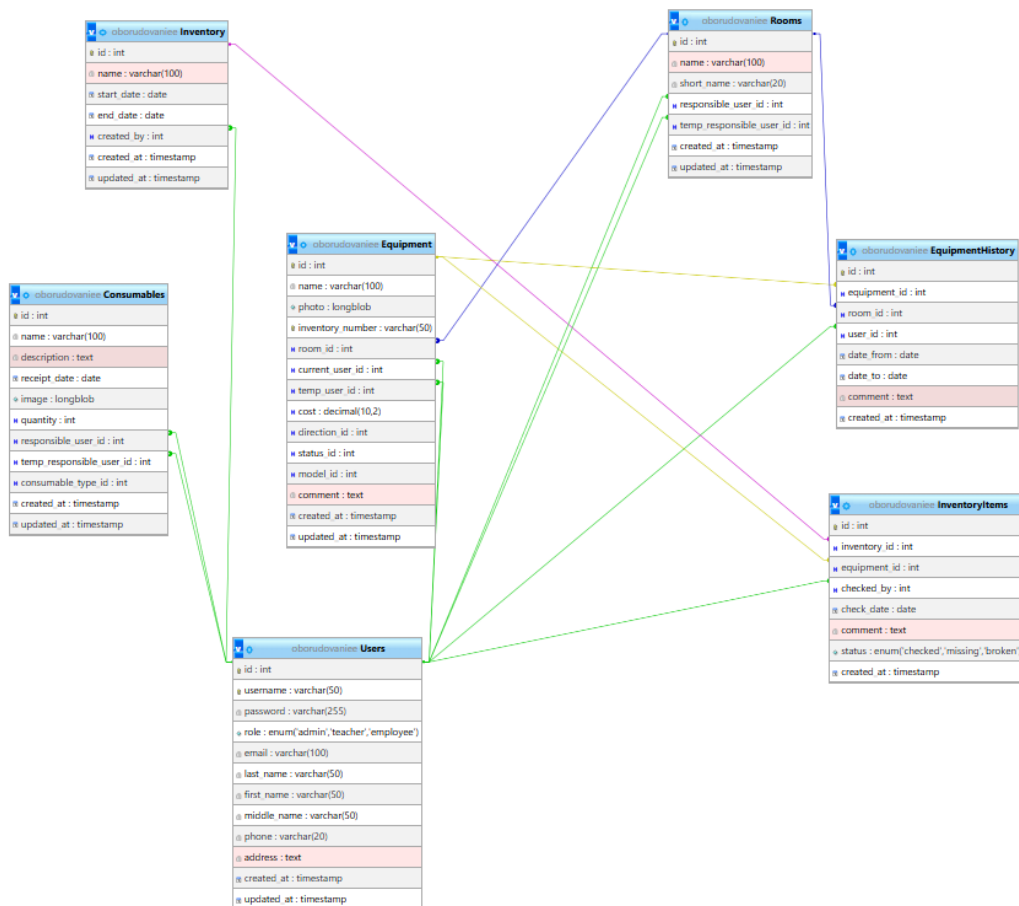
```
CREATE TABLE Inventory ( id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NULL, start_date DATE NOT NULL, end_date DATE, created_by INT NOT NULL, created_at TIMESTAMPT DEFAULT CURRENT_TIMESTAMP, updated_at TIMESTAMPT DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, FOREIGN KEY (created_by) REFERENCES Users(id) );
```

[Построение редактирование](#) [Изменить](#) [Создать PHP-код](#)

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0.6396 сек.)

```
CREATE TABLE InventoryItems ( id INT AUTO_INCREMENT PRIMARY KEY, inventory_id INT NOT NULL, equipment_id INT NOT NULL, checked_by INT, check_date DATE, comment TEXT, status ENUM('checked', 'missing', 'broken') DEFAULT 'checked', created_at TIMESTAMPT DEFAULT CURRENT_TIMESTAMP, updated_at TIMESTAMPT DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP, FOREIGN KEY (inventory_id) REFERENCES Inventory(id), FOREIGN KEY (equipment_id) REFERENCES Equipment(id), FOREIGN KEY (checked_by) REFERENCES Users(id) );
```

[Построение редактирование](#) [Изменить](#) [Создать PHP-код](#)



Настройка индексов для ускорения запросов

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0603 сек.)

```
CREATE INDEX idx_equipment_room ON Equipment(room_id);
```

[[Построчное редактирование](#)] [[Изменить](#)] [[Создать PHP-код](#)]

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,1184 сек.)

```
CREATE INDEX idx_equipment_user ON Equipment(current_user_id);
```

[[Построчное редактирование](#)] [[Изменить](#)] [[Создать PHP-код](#)]

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0211 сек.)

```
CREATE INDEX idx_equipment_status ON Equipment(status_id);
```

[[Построчное редактирование](#)] [[Изменить](#)] [[Создать PHP-код](#)]

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0557 сек.)

```
CREATE INDEX idx_equipment_model ON Equipment(model_id);
```

[[Построчное редактирование](#)] [[Изменить](#)] [[Создать PHP-код](#)]

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,2555 сек.)

```
CREATE INDEX idx_room_responsible ON Rooms(responsible_user_id);
```

[[Построчное редактирование](#)] [[Изменить](#)] [[Создать PHP-код](#)]

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0562 сек.)

```
CREATE INDEX idx_history_equipment ON EquipmentHistory(equipment_id);
```

[[Построчное редактирование](#)] [[Изменить](#)] [[Создать PHP-код](#)]

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0107 сек.)

```
CREATE INDEX idx_history_dates ON EquipmentHistory(date_from, date_to);
```

[[Построчное редактирование](#)] [[Изменить](#)] [[Создать PHP-код](#)]

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,1289 сек.)

```
CREATE INDEX idx_inventory_items ON InventoryItems(inventory_id, equipment_id);
```

[[Построчное редактирование](#)] [[Изменить](#)] [[Создать PHP-код](#)]

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0110 сек.)

```
CREATE INDEX idx_inventory_check_date ON InventoryItems(check_date);
```

[[Построчное редактирование](#)] [[Изменить](#)] [[Создать PHP-код](#)]

Анализ производительности запросов (EXPLAIN в SQL)

Делала Вера

```
EXPLAIN SELECT e.name, r.name AS room_name, u.last_name FROM Equipment e JOIN Rooms r ON e.room_id = r.id JOIN Users u ON e.current_user_id = u.id WHERE e.status_id = 1;
```

[[Построчное редактирование](#)] [[Изменить](#)] [[Убрать анализ SQL](#)] [[Результат анализа mariadb.org](#)] [[Создать PHP-код](#)]

Extra options

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	e	NULL	ref	idx_equipment_room,idx_equipment_user,idx_equipmen...	idx_equipment_status	5	const	1	100.00	Using where
1	SIMPLE	r	NULL	eq_ref	PRIMARY	PRIMARY	4	oborudovanie.e.room_id	1	100.00	NULL
1	SIMPLE	u	NULL	eq_ref	PRIMARY	PRIMARY	4	oborudovanie.e.current_user_id	1	100.00	NULL

Проверка пользовательских ограничений

Делала Мунира

-- Для поиска по названию (частичное совпадение)

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,1316 сек.)

```
CREATE INDEX idx_equipment_name ON Equipment(name);
```

[[Построчное редактирование](#)] [[Изменить](#)] [[Создать PHP-код](#)]

-- Комбинированный индекс для частых совместных фильтров

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,1189 сек.)

```
CREATE INDEX idx_equipment_room_status ON Equipment(room_id, status_id);
```

[[Построчное редактирование](#)] [[Изменить](#)] [[Создать PHP-код](#)]

-- Для поиска по названию

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0218 сек.)

```
CREATE INDEX idx_rooms_name ON Rooms(name);
```

[[Построчное редактирование](#)] [[Изменить](#)] [[Создать PHP-код](#)]

-- Для фильтрации по ответственному пользователю

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,1188 сек.)

```
CREATE INDEX idx_rooms_responsible ON Rooms(responsible_user_id);
```

[[Построчное редактирование](#)] [[Изменить](#)] [[Создать PHP-код](#)]

-- Для сортировки по названию

✓ MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0939 сек.)

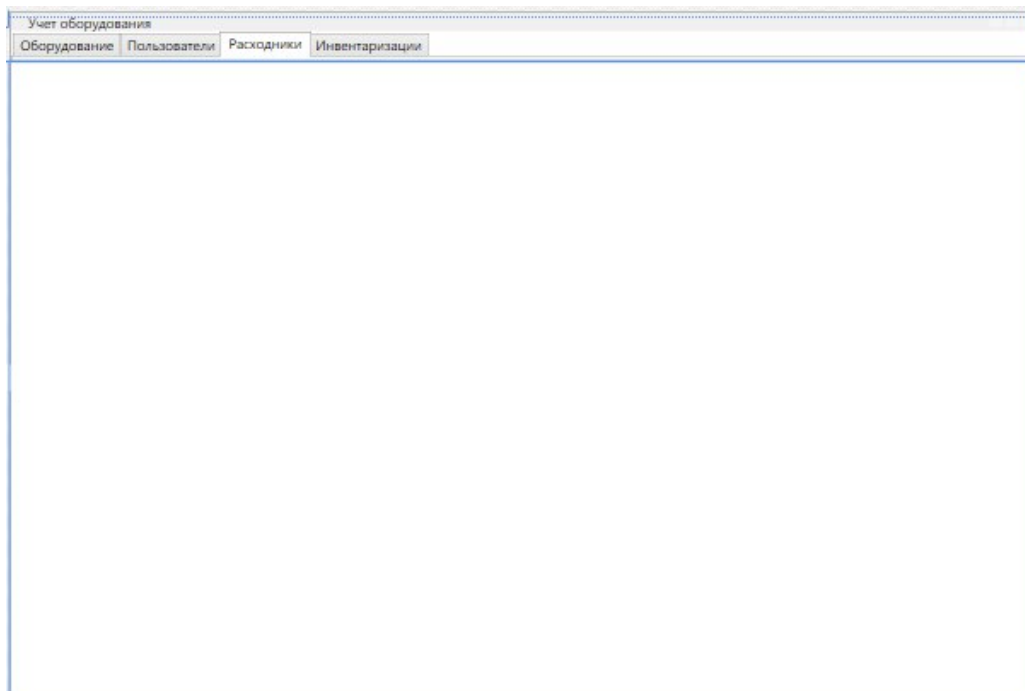
```
CREATE INDEX idx_rooms_name_sort ON Rooms(name ASC);
```

[[Построчное редактирование](#)] [[Изменить](#)] [[Создать PHP-код](#)]

Занятие 2.

На этом занятии были сделаны интерфейсы страничек, с помощью раннее созданного бд.

Создана главная страничка с помощью которой мы переходим на другие.



```
<Window x:Class="EquipmentManagement.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="Учет оборудования" Height="600" Width="900"
        WindowStartupLocation="CenterScreen">

    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*/>
        </Grid.RowDefinitions>

        <!-- Меню -->
        <TabControl Grid.Row="0">
            <TabItem Header="Оборудование" x:Name="EquipmentTab"/>
            <TabItem Header="Пользователи" x:Name="UsersTab"/>
            <TabItem Header="Расходники" x:Name="ConsumablesTab"/>
            <TabItem Header="Инвентаризации" x:Name="InventoryTab"/>
        </TabControl>

        <!-- Контент -->
        <Frame Grid.Row="1" x:Name="MainFrame" NavigationUIVisibility="Hidden"/>
    </Grid>
</Window>
```

Далее создана инвентаризация и история инвентаризации, где можно посмотреть дату начала и окончания, помещение и пользователя.

История оборудования

(%)

✓

<

История перемещения оборудования

Выбор даты

15

Выбор даты

15

Применить фильтр

Сбросить

ID	Оборудование	Помещение	Пользователь	Дата начала	Дата окончания
1	Ноутбук Dell XPS	Кабинет 101	Иванов И.И.	10.01.2023	15.05.2023
2	Проектор Epson	Актный зал	Петров П.П.	01.03.2023	

Загружено записей: 2

16.05.2025 11:23

16.05.2025 11:23

```

<Grid Margin="10">
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*/"/>
        <RowDefinition Height="Auto"/>
    </Grid.RowDefinitions>

    <!-- Заголовок и фильтры -->
    <StackPanel Grid.Row="0" Orientation="Vertical" Margin="0,0,0,10">
        <TextBlock Text="История перемещения оборудования"
            FontSize="16" FontWeight="Bold" Margin="0,0,0,10"/>

        <StackPanel Orientation="Horizontal">
            <DatePicker x:Name="DateFromPicker" Width="120" Margin="0,0,10,0"/>
            <DatePicker x:Name="DateToPicker" Width="120" Margin="0,0,10,0"/>
            <Button Content="Применить фильтр" Width="120" Click="ApplyFilter_Click"/>
            <Button Content="Сбросить" Width="80" Margin="10,0,0,0" Click="ResetFilter_Click"/>
        </StackPanel>
    </StackPanel>

    <!-- Таблица истории -->
    <DataGrid Grid.Row="1" x:Name="HistoryDataGrid" AutoGenerateColumns="False"
        CanUserAddRows="False" IsReadOnly="True" SelectionMode="Single">
        <DataGrid.Columns>
            <DataGridTextColumn Header="ID" Binding="{Binding Id}" Width="Auto"/>
            <DataGridTextColumn Header="Оборудование" Binding="{Binding EquipmentName}" Width="*/"/>
            <DataGridTextColumn Header="Помещение" Binding="{Binding RoomName}" Width="*/"/>
            <DataGridTextColumn Header="Пользователь" Binding="{Binding UserName}" Width="*/"/>
            <DataGridTextColumn Header="Дата начала" Binding="{Binding DateFrom, StringFormat=dd.MM.yyyy}" Width="120"/>
            <DataGridTextColumn Header="Дата окончания" Binding="{Binding DateTo, StringFormat=dd.MM.yyyy}" Width="120"/>
        </DataGrid.Columns>
    </DataGrid>

    <!-- Статус бар -->
    <StatusBar Grid.Row="2">
        <StatusBarItem>
            <TextBlock x:Name="StatusText" Text="Горюхо"/>
        </StatusBarItem>
        <StatusBarItem HorizontalAlignment="Right">
            <TextBlock Text="{Binding CurrentDateTime}" />
        </StatusBarItem>
    </StatusBar>
</Grid>
</Window>

```

```

private void LoadHistoryData()
{
    try
    {
        // Здесь должна быть логика загрузки данных из БД
        // Вместо этого используем тестовые данные
        HistoryItems = new ObservableCollection<EquipmentHistory>
        {
            new EquipmentHistory
            {
                Id = 1,
                EquipmentName = "Ноутбук Dell XPS",
                RoomName = "Кабинет 101",
                UserName = "Иванов И.И.",
                DateFrom = new DateTime(2023, 1, 10),
                DateTo = new DateTime(2023, 5, 15),
                Comment = "Основной пользователь"
            },
            new EquipmentHistory
            {
                Id = 2,
                EquipmentName = "Проектор Epson",
                RoomName = "Актовый зал",
                UserName = "Петров П.П.",
                DateFrom = new DateTime(2023, 3, 1),
                DateTo = null,
                Comment = "Для проведения мероприятий"
            }
        };

        HistoryDataGrid.ItemsSource = HistoryItems;
        StatusText.Text = $"Загружено записей: {HistoryItems.Count}";
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка загрузки данных: {ex.Message}", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
        StatusText.Text = "Ошибка загрузки данных";
    }
}

Ссылка 1
private void ApplyFilter_Click(object sender, RoutedEventArgs e)
{
    // Логика фильтрации по дате
    if (DateFromPicker.SelectedDate != null || DateToPicker.SelectedDate != null)
    {
        StatusText.Text = "Фильтр применен";
    }
}

}

catch (Exception ex)
{
    MessageBox.Show($"Ошибка загрузки данных: {ex.Message}", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
    StatusText.Text = "Ошибка загрузки данных";
}

}

Ссылка 1
private void ApplyFilter_Click(object sender, RoutedEventArgs e)
{
    // Логика фильтрации по дате
    if (DateFromPicker.SelectedDate != null || DateToPicker.SelectedDate != null)
    {
        StatusText.Text = "Фильтр применен";
    }
}

}

Ссылка 1
private void ResetFilter_Click(object sender, RoutedEventArgs e)
{
    DateFromPicker.SelectedDate = null;
    DateToPicker.SelectedDate = null;
    StatusText.Text = "Фильтр сброшен";
}

}

```

Была создана управление инвентаризационными позициями, где можно применять фильтры, выбирать дату, смотреть статусы.

Управление инвентаризационными позициями

Инвентаризационные позиции

Все статусы

Выбор даты

15

Применить фильтры

Сбросить

ID	ID инвентаризации	ID оборудования	Проверено (ID)	Дата проверки	Статус	Комментарий
1	101	5001	1001	15.05.2023	checked	В отличном состоянии
2	101	5002	1001	15.05.2023	missing	Не найдено на месте
3	101	5003	1001	16.05.2023	broken	Требуется ремонт

Загружено позиций: 3

16.05.2025 11:31

```
<Grid.RowDefinitions>
<RowDefinition Height="Auto"/>
<RowDefinition Height="*" />
<RowDefinition Height="Auto"/>
</Grid.RowDefinitions>

<!-- Панель заголовка и фильтров -->
<StackPanel Grid.Row="0" Orientation="Vertical" Margin="0,0,0,15">
<TextBlock Text="Инвентаризационные позиции"
FontSize="18" FontWeight="Bold" Margin="0,0,0,10"/>

<StackPanel Orientation="Horizontal">
<ComboBox x:Name="StatusFilter" Width="150" Margin="0,0,10,0"
DisplayMemberPath="Value" SelectedValuePath="Key"
SelectedValue="{Binding StatusFilterValue}"/>

<DatePicker x:Name="DateFilter" Width="150" Margin="0,0,10,0"/>

<Button Content="Применить фильтры" Width="120" Padding="5"
Click="ApplyFilters_Click"/>

<Button Content="Сбросить" Width="80" Margin="10,0,0,0" Padding="5"
Click="ResetFilters_Click"/>
</StackPanel>
</StackPanel>

<!-- Таблица данных -->
<DataGrid Grid.Row="1" x:Name="ItemsDataGrid"
AutoGenerateColumns="False" IsReadOnly="True"
SelectionMode="Single" SelectionUnit="FullRow">
<DataGrid.Columns>
<DataGridTextColumn Header="ID" Binding="{Binding Id}" Width="Auto"/>
<DataGridTextColumn Header="ID инвентаризации" Binding="{Binding InventoryId}" Width="120"/>
<DataGridTextColumn Header="ID оборудования" Binding="{Binding EquipmentId}" Width="120"/>
<DataGridTextColumn Header="Проверено (ID)" Binding="{Binding CheckedById}" Width="100"/>
<DataGridTextColumn Header="Дата проверки" Binding="{Binding CheckDate, StringFormat=dd.MM.yyyy}" Width="120"/>
<DataGridTextColumn Header="Статус" Binding="{Binding Status}" Width="100"/>
<DataGridTextColumn Header="Комментарий" Binding="{Binding Comment}" Width="*"/>
</DataGrid.Columns>
</DataGrid>

<!-- Панель статуса -->
<StatusBar Grid.Row="2">
<StatusBarItem>
<TextBlock x:Name="StatusText" Text="Горючо"/>
</StatusBarItem>
<StatusBarItem HorizontalAlignment="Right">
<TextBlock Text="{Binding CurrentDateTime}"/>
</StatusBarItem>
</StatusBar>
```

```

Ссылка 0
public MainWindow()
{
    InitializeComponent();
    DataContext = this;
    InitializeFilters();
    LoadInventoryItems();
}

Ссылка 1
private void InitializeFilters()
{
    // Инициализация фильтров статуса
    var statusItems = new[]
    {
        new { Key = "all", Value = "Все статусы" },
        new { Key = "checked", Value = "Проверено" },
        new { Key = "missing", Value = "Отсутствует" },
        new { Key = "broken", Value = "Сломано" }
    };

    StatusFilter.ItemsSource = statusItems;
    StatusFilter.SelectedIndex = 0;
}

Ссылка 1
private void LoadInventoryItems()
{
    try
    {
        // Здесь должна быть логика загрузки данных из БД
        // Вместо этого используем тестовые данные
        InventoryItems = new ObservableCollection<InventoryItem>
        {
            new InventoryItem
            {
                Id = 1,
                InventoryId = 101,
                EquipmentId = 5001,
                CheckedById = 1001,
                CheckDate = new DateTime(2023, 5, 15),
                Status = "checked",
                Comment = "В отличном состоянии"
            },
            new InventoryItem
            {
                Id = 2,
                InventoryId = 101,
                EquipmentId = 5002
            }
        };
    }
}

```

```

    try
    {
        // Здесь должна быть логика загрузки данных из БД
        // Вместо этого используем тестовые данные
        InventoryItems = new ObservableCollection<InventoryItem>
        {
            new InventoryItem
            {
                Id = 1,
                InventoryId = 101,
                EquipmentId = 5001,
                CheckedById = 1001,
                CheckDate = new DateTime(2023, 5, 15),
                Status = "checked",
                Comment = "В отличном состоянии"
            },
            new InventoryItem
            {
                Id = 2,
                InventoryId = 101,
                EquipmentId = 5002,
                CheckedById = 1001,
                CheckDate = new DateTime(2023, 5, 15),
                Status = "missing",
                Comment = "Не найдено на месте"
            },
            new InventoryItem
            {
                Id = 3,
                InventoryId = 101,
                EquipmentId = 5003,
                CheckedById = 1001,
                CheckDate = new DateTime(2023, 5, 16),
                Status = "broken",
                Comment = "Требуется ремонт"
            }
        };

        ItemsDataGrid.ItemsSource = InventoryItems;
        StatusText.Text = $"Загружено позиций: {InventoryItems.Count}";
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка загрузки данных: {ex.Message}", "Ошибка",
            MessageBoxButton.OK, MessageBoxImage.Error);
        StatusText.Text = "Ошибка загрузки данных";
    }
}

Ссылка 1
private void ApplyFilters_Click(object sender, RoutedEventArgs e)

```

```

};

ItemsDataGrid.ItemsSource = InventoryItems;
StatusText.Text = $"Загружено позиций: {InventoryItems.Count}";
}
catch (Exception ex)
{
    MessageBox.Show($"Ошибка загрузки данных: {ex.Message}", "Ошибка",
        MessageBoxButton.OK, MessageBoxImage.Error);
    StatusText.Text = "Ошибка загрузки данных";
}
}

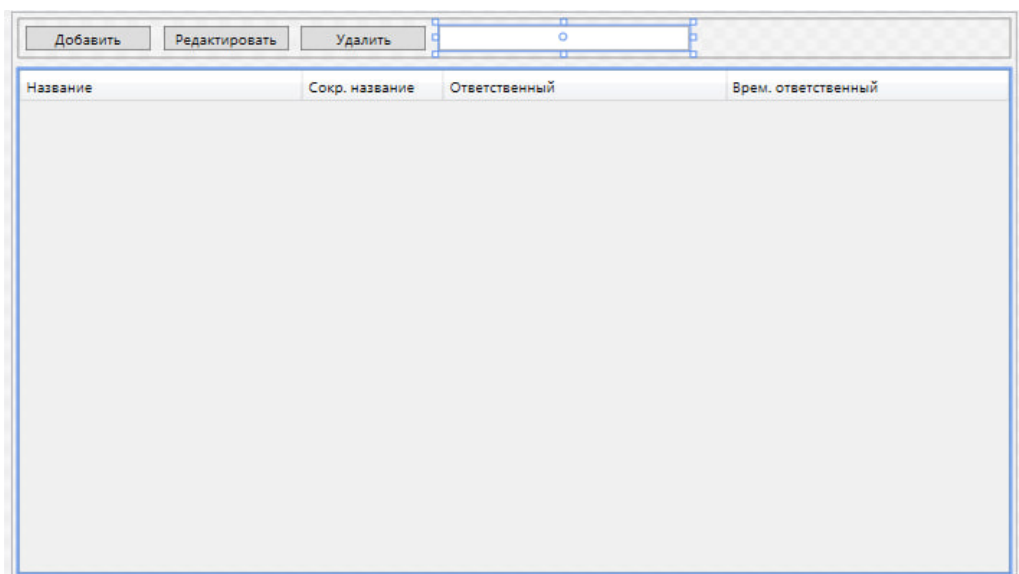
Ссылка 1
private void ApplyFilters_Click(object sender, RoutedEventArgs e)
{
    // Логика применения фильтров
    string statusFilter = (StatusFilter.SelectedValue as string) ?? "all";
    DateTime? dateFilter = DateFilter.SelectedDate;

    // Здесь должна быть логика фильтрации данных
    StatusText.Text = "Фильтры применены";
}

Ссылка 1
private void ResetFilters_Click(object sender, RoutedEventArgs e)
{
    StatusFilter.SelectedIndex = 0;
    DateFilter.SelectedDate = null;
    StatusText.Text = "Фильтры сброшены";
}
}

```

Реализовала окно инвентаризационных позиций, сделала фильтры по дате, чтобы было удобнее искать.




```

26
27
28 <!-- Список аудиторий -->
29 <DataGrid Grid.Row="1" x:Name="RoomsGrid" Margin="5"
30     AutoGenerateColumns="False" IsReadOnly="True"
31     SelectionMode="Single">
32     <DataGrid.Columns>
33         <DataGridTextColumn Header="Название" Binding="{Binding Name}" Width="2*"/>
34         <DataGridTextColumn Header="Сокр. название" Binding="{Binding ShortName}" Width="1*"/>
35         <DataGridTextColumn Header="Ответственный"
36             Binding="{Binding ResponsibleUser.LastName}" Width="2*"/>
37         <DataGridTextColumn Header="Врем. ответственный"
38             Binding="{Binding TempResponsibleUser.LastName}" Width="2*"/>
39     </DataGrid.Columns>
40 </DataGrid>
41 </Grid>
42 </Page>

```

RoomEditWindow

Название:

Сокращение:

Ответственный:

Временный ответ.:

```

8      Title="RoomEditWindow" Height="450" Width="800">
9      <Grid Margin="10">
10      <Grid.RowDefinitions>
11      <RowDefinition Height="Auto"/>
12      <RowDefinition Height="Auto"/>
13      <RowDefinition Height="Auto"/>
14      <RowDefinition Height="Auto"/>
15      <RowDefinition Height="Auto"/>
16      <RowDefinition Height="40"/>
17      </Grid.RowDefinitions>
18
19      <Grid.ColumnDefinitions>
20      <ColumnDefinition Width="150"/>
21      <ColumnDefinition Width="*"/>
22      </Grid.ColumnDefinitions>
23
24      <Label Grid.Row="0" Grid.Column="0" Content="Название:"/>
25      <TextBox Grid.Row="0" Grid.Column="1" x:Name="NameBox" Margin="5"/>
26
27      <Label Grid.Row="1" Grid.Column="0" Content="Сокращение:"/>
28      <TextBox Grid.Row="1" Grid.Column="1" x:Name="ShortNameBox" Margin="5"/>
29
30      <Label Grid.Row="2" Grid.Column="0" Content="Ответственный:"/>
31      <ComboBox Grid.Row="2" Grid.Column="1" x:Name="ResponsibleUserCombo"
32      DisplayMemberPath="LastName" Margin="5"/>
33
34      <Label Grid.Row="3" Grid.Column="0" Content="Временный ответ.:"/>
35      <ComboBox Grid.Row="3" Grid.Column="1" x:Name="TempResponsibleUserCombo"
36      DisplayMemberPath="LastName" Margin="5"/>
37
38      <!-- Кнопки -->
39      <StackPanel Grid.Row="5" Grid.ColumnSpan="2" Orientation="Horizontal" HorizontalAlignment="Right">
40      <Button Content="Сохранить" Width="80" Margin="5" Click="Save_Click"/>
41      <Button Content="Отмена" Width="80" Margin="5" Click="Cancel_Click"/>
42      </StackPanel>
43      </Grid>
44      </Window>
45

```

Вэра

Управление пользователями

3

Список пользователей

Найти

Добавить

ID	Логин	ФИО	Роль	Email	Телефон	Действия
1	admin1		admin	ivanov@example.com	+7 (123) 456-7890	<div>Изменить</div> <div>Удалить</div>
2	teacher1		teacher	petrova@example.com	+7 (987) 654-3210	<div>Изменить</div> <div>Удалить</div>
3	employee1		employee	sidorov@example.com	+7 (555) 123-4567	<div>Изменить</div> <div>Удалить</div>

Загружено пользователей: 3

16.05.2025 11:39

```

<Window x:Class="UserManagementSystem.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="Управление пользователями"
        Width="1000" Height="700"
        MinWidth="800" MinHeight="600"
        WindowStartupLocation="CenterScreen">

    <Grid Margin="10">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*" />
            <RowDefinition Height="Auto"/>
        </Grid.RowDefinitions>

        <!-- Заголовок и панель управления -->
        <StackPanel Grid.Row="0" Orientation="Vertical" Margin="0,0,0,15">
            <TextBlock Text="Список пользователей"
                FontSize="18" FontWeight="Bold" Margin="0,0,0,10"/>

            <StackPanel Orientation="Horizontal">
                <TextBox x:Name="SearchBox" Width="250" Margin="0,0,10,0"
                    Text="{Binding SearchText, UpdateSourceTrigger=PropertyChanged}"
                    />

                <ComboBox x:Name="RoleFilter" Width="150" Margin="0,0,10,0"
                    ItemsSource="{Binding Roles}"
                    SelectedItem="{Binding SelectedRole}" />

                <Button Content="Найти" Width="80" Command="{Binding SearchCommand}" />
                <Button Content="Добавить" Width="80" Margin="10,0,0,0"
                    Command="{Binding AddUserCommand}" />
            </StackPanel>
        </StackPanel>

        <!-- Таблица пользователей -->
        <DataGrid Grid.Row="1" x:Name="UsersDataGrid"
            ItemsSource="{Binding Users}"
            AutoGenerateColumns="False" IsReadOnly="True"
            SelectionMode="Single" SelectionUnit="FullRow">
            <DataGrid.Columns>
                <DataGridTextColumn Header="ID" Binding="{Binding Id}" Width="Auto"/>
                <DataGridTextColumn Header="Логин" Binding="{Binding Username}" Width="120"/>
                <DataGridTextColumn Header="ФИО"
                    Binding="{Binding LastName, StringFormat='{0} {1} {2}',
                        ConverterParameter='FirstName,MiddleName'}"
                    Width="200"/>
                <DataGridTextColumn Header="Поль" Binding="{Binding Role}" Width="100"/>
                <DataGridTextColumn Header="Email" Binding="{Binding Email}" Width="200"/>
                <DataGridTextColumn Header="Телефон" Binding="{Binding Phone}" Width="120"/>
            </DataGrid.Columns>
        </DataGrid>

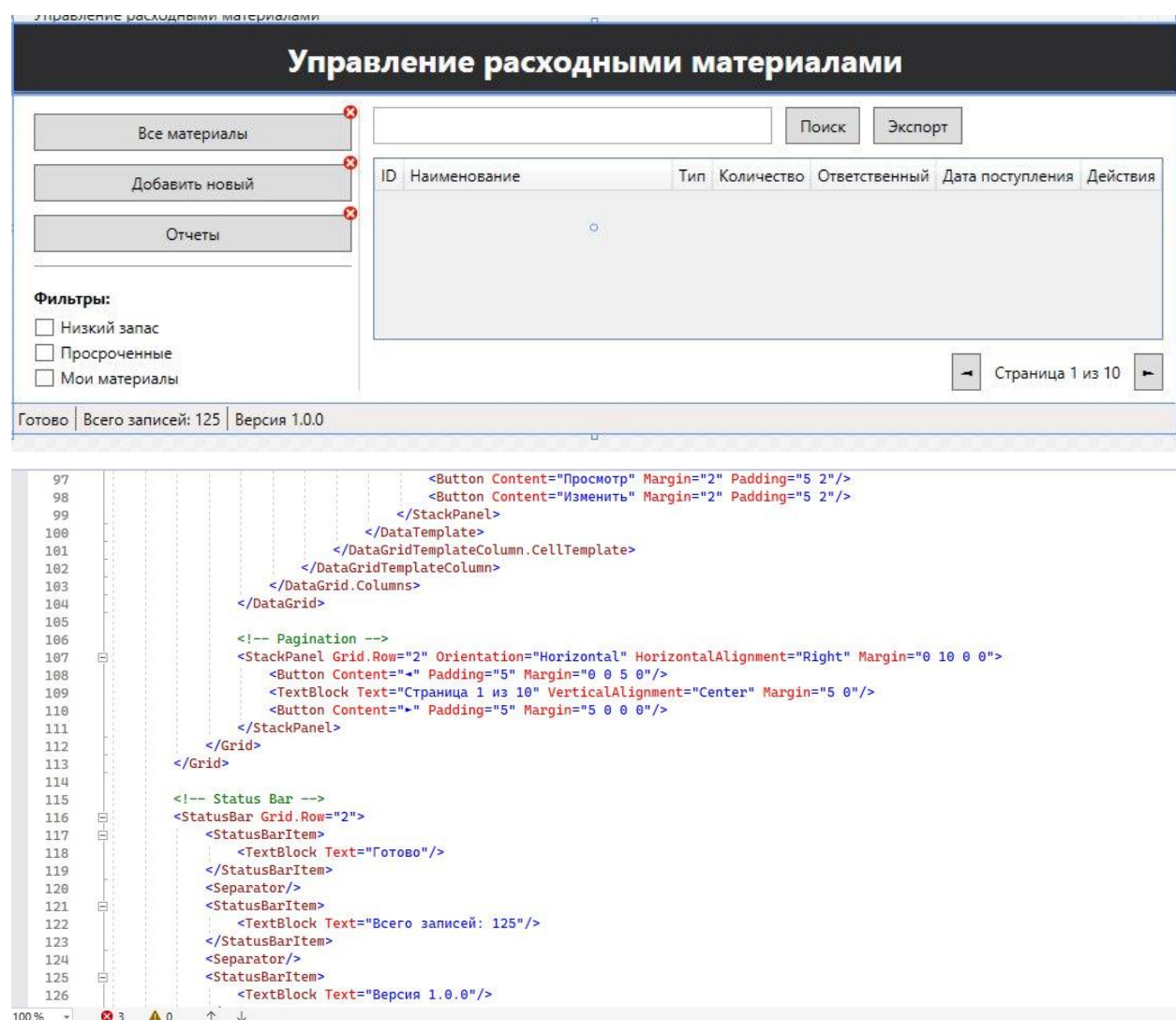
        <!-- Таблица пользователей -->
        <DataGrid Grid.Row="1" x:Name="UsersDataGrid"
            ItemsSource="{Binding Users}"
            AutoGenerateColumns="False" IsReadOnly="True"
            SelectionMode="Single" SelectionUnit="FullRow">
            <DataGrid.Columns>
                <DataGridTextColumn Header="ID" Binding="{Binding Id}" Width="Auto"/>
                <DataGridTextColumn Header="Логин" Binding="{Binding Username}" Width="120"/>
                <DataGridTextColumn Header="ФИО"
                    Binding="{Binding LastName, StringFormat='{0} {1} {2}',
                        ConverterParameter='FirstName,MiddleName'}"
                    Width="200"/>
                <DataGridTextColumn Header="Поль" Binding="{Binding Role}" Width="100"/>
                <DataGridTextColumn Header="Email" Binding="{Binding Email}" Width="200"/>
                <DataGridTextColumn Header="Телефон" Binding="{Binding Phone}" Width="120"/>
                <DataGridTemplateColumn Header="Действия" Width="Auto">
                    <DataGridTemplateColumn.CellTemplate>
                        <DataTemplate>
                            <StackPanel Orientation="Horizontal">
                                <Button Content="Изменить" Width="80" Margin="0,0,5,0"
                                    Command="{Binding DataContext.EditUserCommand,
                                        RelativeSource={RelativeSource AncestorType=DataGrid}}"
                                    CommandParameter="{Binding}" />
                                <Button Content="Удалить" Width="80"
                                    Command="{Binding DataContext.DeleteUserCommand,
                                        RelativeSource={RelativeSource AncestorType=DataGrid}}"
                                    CommandParameter="{Binding}" />
                            </StackPanel>
                        </DataTemplate>
                    </DataGridTemplateColumn.CellTemplate>
                </DataGridTemplateColumn>
            </DataGrid.Columns>
        </DataGrid>

        <!-- Статус бар -->
        <StatusBar Grid.Row="2">
            <StatusBarItem>
                <TextBlock Text="{Binding StatusMessage}" />
            </StatusBarItem>
            <StatusBarItem HorizontalAlignment="Right">
                <TextBlock Text="{Binding CurrentDateTime}" />
            </StatusBarItem>
        </StatusBar>
    </Grid>
</Window>

```

Также реализовала окно со списком пользователей/учителей, где выводятся все учителя их логины. Управление расходными материалами

сделала более понятный интерфейс, где можно увидеть что есть, что заканчивается и т.п.



Вывод: был реализован интерфейс всех страничек, протестированы все интерфейсы.