

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления
Кафедра Интеллектуальных информационных технологий

ОТЧЁТ

по лабораторной работе №2
по дисциплине «Проектирование Баз Знаний»
вариант 18

Выполнил:

Т. М. Робилко

Студент группы
221701

Проверил:

М. Г. Соколович

Минск 2024

1 Постановка задачи

Вариант 18 «Учет отказа оборудования»:

Предприятие, имеющее в своей структуре иерархически подчиняющиеся подразделения.

Необходимо вести учет отказа оборудования на каком-либо участке предприятия. Производственные участки на предприятии имеют номер, название (гальванический участок, участок переработки сырья и т.п.). На каждом участке может работать разное оборудование: газовое, сварочное, электрическое. Оборудование имеет номер и название (манометр, газосварочный аппарат и т.п.). Любое оборудование проходит технический осмотр, за год их м.б. несколько. Фиксируется дата осмотра и результат – годен к работе, передать в ремонт, списать и т.п., а также причину нерабочего состояния оборудования (механическая поломка, электропроводка и т.п. – причины отказа). Каждый осмотр проводит какой-либо сотрудник технического отдела, необходимо хранить о нем следующие данные: табельный номер, ФИО, занимаемую на время осмотра должность. Оборудование может отказать в работе и между техническими осмотрами, в таком случае также фиксируется дата, причина, ФИО сотрудника, проводшего соответствующий осмотр на момент отказа.

Необходимо реализовать выполнения следующих функций:

- Добавление/редактирование/удаление информации о производственных участках.
- Добавление/редактирование/удаление информации об оборудовании.
- Добавление/редактирование/удаление информации о техническом осмотре оборудования.
- Добавление/редактирование/удаление информации о сотруднике.
- Просмотр списка отказавшего оборудования на участках предприятия: дата, список (название оборудования, тип оборудования, название участка, причина отказа, дата отказа).
- Просмотр истории технических осмотров оборудования с заданным инвентарным номером: дата формирования отчета, инвентарный номер, название, тип оборудования, результат осмотра.
- Просмотр имён и должностей сотрудников на заданную дату: дата, список (ФИО, должность).

2 Концептуальное проектирование

2.1 BPMN диаграммы бизнес-процессов

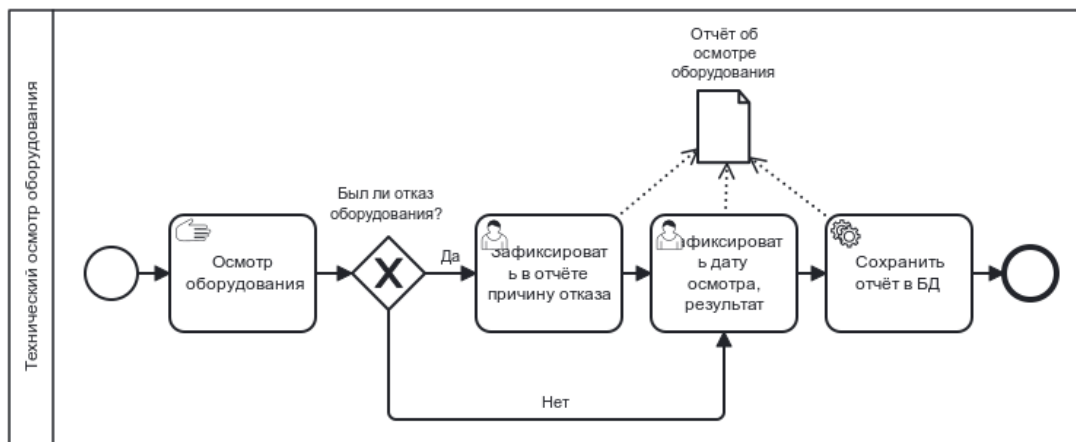


Рис. 1: BPMN-диаграмма процесса технического осмотра оборудования

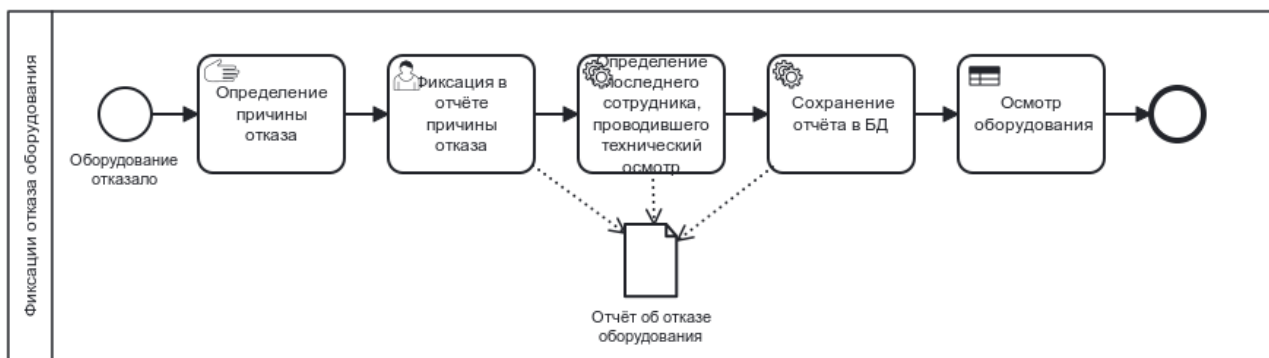


Рис. 2: BPMN-диаграмма процесса фиксации отказа оборудования

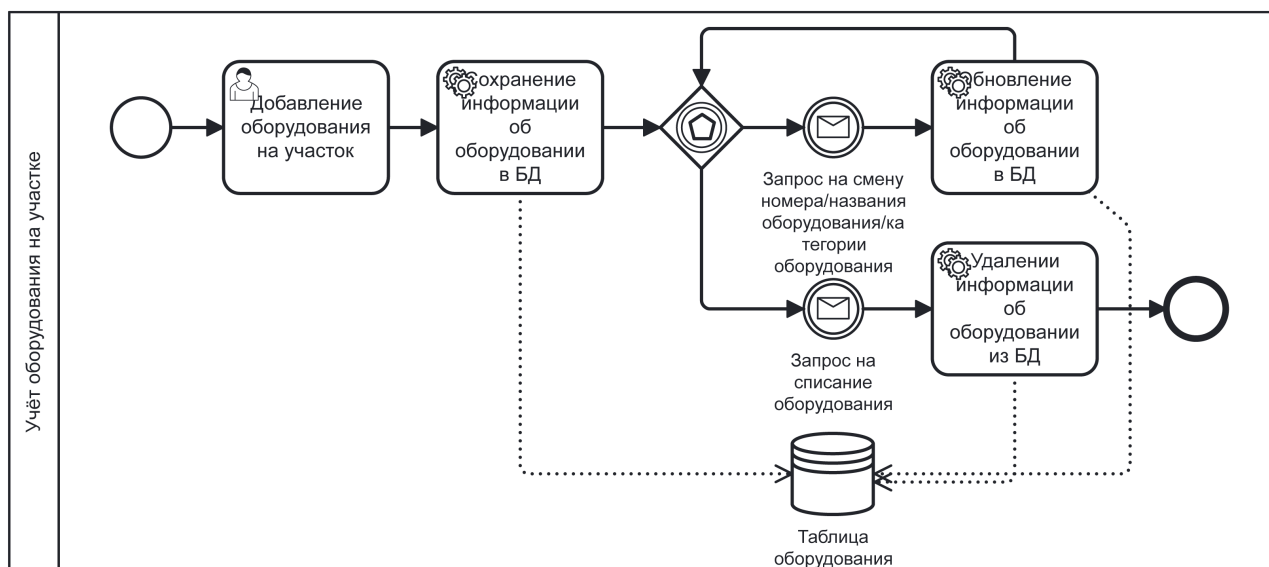


Рис. 3: BPMN-диаграмма процесса учета оборудования на участке

3 Логическое проектирование

В рамках логического проектирования были продуманы сущности системы согласно текстовому описанию, построена ER-диаграмма для их графического отображения.

3.1 ER диаграмма



Рис. 4: ER-диаграмма проектируемой системы

4 Физическое проектирование

В рамках физического проектирования была разработана схема базы данных для хранения вышеописанных сущностей, а также их взаимосвязей.

4.1 Схема базы данных

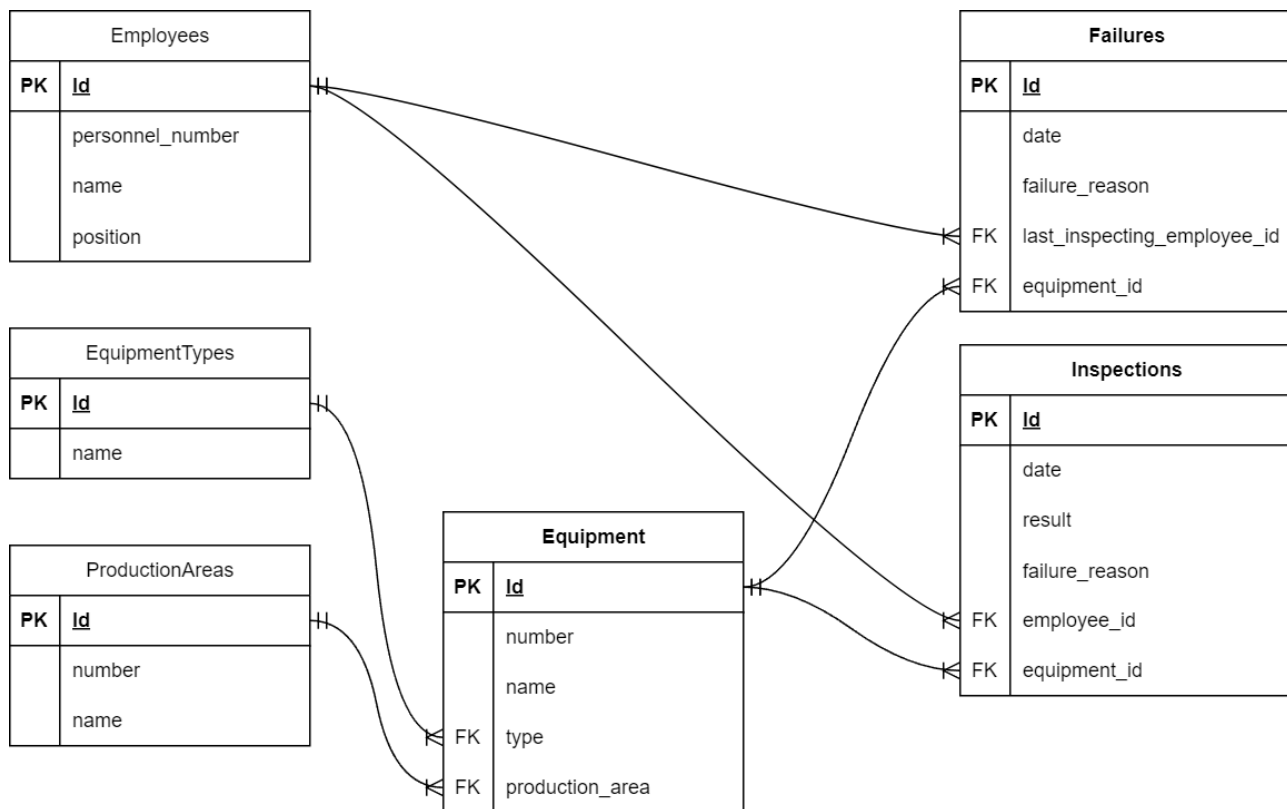


Рис. 5: Схема базы данных проектируемой системы

5 Основные запросы, функции и процедуры системы

5.1 Процедура инициализации базы данных

Для инициализации базы данных и создания таблиц в соответствии с разработанными схемами была представлена ниже реализована процедура.

```
CREATE PROCEDURE CreateTableIfNotExists()
BEGIN
    CREATE TABLE IF NOT EXISTS Employees (
        Id INT AUTO_INCREMENT PRIMARY KEY,
        personnel_number NVARCHAR(50) NOT NULL,
        name NVARCHAR(255) NOT NULL,
        position NVARCHAR(255) NOT NULL
    );

    CREATE TABLE IF NOT EXISTS ProductionAreas (
        Id INT AUTO_INCREMENT PRIMARY KEY,
        number NVARCHAR(50) NOT NULL,
        name NVARCHAR(255) NOT NULL
    );

    CREATE TABLE IF NOT EXISTS EquipmentTypes (
        Id INT AUTO_INCREMENT PRIMARY KEY,
        name NVARCHAR(255) NOT NULL
    );

    CREATE TABLE IF NOT EXISTS Equipment (
        Id INT AUTO_INCREMENT PRIMARY KEY,
        number NVARCHAR(50) NOT NULL,
        name NVARCHAR(255) NOT NULL,
        type INT NOT NULL, -- FK to EquipmentTypes
        production_area INT NOT NULL, -- FK to ProductionAreas
        FOREIGN KEY (type) REFERENCES EquipmentTypes(Id) ON DELETE CASCADE,
        FOREIGN KEY (production_area) REFERENCES ProductionAreas(Id)
        ON DELETE CASCADE
    );

    CREATE TABLE IF NOT EXISTS Failures (
        Id INT AUTO_INCREMENT PRIMARY KEY,
        date DATETIME NOT NULL,
        failure_reason NVARCHAR(255),
        last_inspecting_employee_id INT, -- FK to Employees
        equipment_id INT NOT NULL, -- FK to Equipment
        FOREIGN KEY (last_inspecting_employee_id) REFERENCES Employees(Id)
        ON DELETE SET NULL,
        FOREIGN KEY (equipment_id) REFERENCES Equipment(Id) ON DELETE CASCADE
    );

    CREATE TABLE IF NOT EXISTS Inspections (
        Id INT AUTO_INCREMENT PRIMARY KEY,
```

```

    date DATETIME NOT NULL,
    result NVARCHAR(50) NOT NULL,
    failure_reason NVARCHAR(255),
    employee_id INT NOT NULL, -- FK to Employees
    equipment_id INT NOT NULL, -- FK to Equipment
    FOREIGN KEY (employee_id) REFERENCES Employees(Id) ON DELETE CASCADE,
    FOREIGN KEY (equipment_id) REFERENCES Equipment(Id) ON DELETE CASCADE
);
END

```

Также для достижения однородности данных и выполнения бизнес-требований были созданы три триггера:

1. Триггер для вставки нового отчёта об техническом осмотре, который автоматически присваивает текущее значение времени записи
2. Триггер для недопущения ручного обновления времени записи о техническом осмотре при обновлении записи
3. Триггер для автоматической установки сотрудника, который проводил технический осмотр, при добавлении новой записи о поломке оборудования

Код триггеров приведён ниже.

```
DELIMITER $$
```

```

DROP TRIGGER IF EXISTS set_inspection_datetime $$
CREATE TRIGGER set_inspection_datetime
BEFORE INSERT ON Inspections
FOR EACH ROW
BEGIN
    SET NEW.date = NOW();
END
$$

```

```

DROP TRIGGER IF EXISTS set_inspection_datetime_update $$
CREATE TRIGGER set_inspection_datetime_update
BEFORE UPDATE ON Inspections
FOR EACH ROW
BEGIN
    SET NEW.date = OLD.date;
END
$$

```

```

DROP TRIGGER IF EXISTS set_last_inspecting_employee_for_failure $$
CREATE TRIGGER set_last_inspecting_employee_for_failure
BEFORE Insert ON Failures
FOR EACH ROW
BEGIN
    SET NEW.last_inspecting_employee_id = (
        select employee_id from Inspections
        where equipment_id = NEW.equipment_id
    )

```

```

    order by Inspections.date desc
    limit 1
);
END
$$

```

```
DELIMITER ;
```

Основной функционал реализован с помощью Entity Framework Core, который позволяет манипулировать данными на уровне кода приложения без прямого написания SQL-запросов. Однако, запросы, которые могут быть использованы для работы с БД без использования графического интерфейса, приведены ниже.

5.2 Добавление/редактирование/удаление информации о производственных участках

5.2.1 Добавление информации о производственных участках

```

INSERT INTO ProductionAreas (number, name)
VALUES ('P123', 'Assembly Line 1');

```

5.2.2 Редактирование информации о производственных участках

```

UPDATE ProductionAreas
SET name = 'Updated Area Name'
WHERE number = 'P123';

```

5.2.3 Удаление информации о производственных участках

```

DELETE FROM ProductionAreas
WHERE number = 'P123';

```

5.3 Добавление/редактирование/удаление информации об оборудовании

5.3.1 Добавление информации об оборудовании

```

INSERT INTO Equipment (number, name, type, production_area)
VALUES ('E567', 'Drill Press', 1, 2);

```

5.3.2 Редактирование информации об оборудовании

```

UPDATE Equipment
SET name = 'Updated Drill Press', type = 2, production_area = 3
WHERE number = 'E567';

```

5.3.3 Удаление информации об оборудовании

```

DELETE FROM Equipment
WHERE number = 'E567';

```


5.4 Добавление/редактирование/удаление информации о техническом осмотре оборудования

5.4.1 Добавление информации о техническом осмотре оборудования

```
INSERT INTO Inspections
(date, result, failure_reason, employee_id, equipment_id)
VALUES (NOW(), 'Pass', NULL, 3, 4);
```

5.4.2 Редактирование информации о техническом осмотре оборудования

```
UPDATE Inspections
SET result = 'Fail', failure_reason = 'Faulty wiring'
WHERE Id = 5;
```

5.4.3 Удаление информации о техническом осмотре оборудования

```
DELETE FROM Inspections
WHERE Id = 5;
```

5.5 Добавление/редактирование/удаление информации о сотруднике

5.5.1 Добавление информации о сотруднике

```
INSERT INTO Employees (personnel_number, name, position)
VALUES ('EMP001', 'John Doe', 'Technician');
```

5.5.2 Редактирование информации о сотруднике

```
UPDATE Employees
SET name = 'Jane Doe', position = 'Senior Technician'
WHERE personnel_number = 'EMP001';
```

5.5.3 Удаление информации о сотруднике

```
DELETE FROM Employees
WHERE personnel_number = 'EMP001';
```

5.6 Просмотр отказавшего оборудования на участках предприятия

```
SELECT F.date AS failure_date, E.name AS equipment_name,
ET.name AS equipment_type, PA.name AS production_area, F.failure_reason
FROM Failures F
JOIN Equipment E ON F.equipment_id = E.Id
JOIN EquipmentTypes ET ON E.type = ET.Id
JOIN ProductionAreas PA ON E.production_area = PA.Id;
```

5.7 Просмотр истории технических осмотров оборудования с заданным инвентарным номером

```
SELECT NOW() AS report_date, E.number AS equipment_number,  
E.name AS equipment_name, ET.name AS equipment_type,  
I.date AS inspection_date, I.result AS inspection_result  
FROM Inspections I  
JOIN Equipment E ON I.equipment_id = E.Id  
JOIN EquipmentTypes ET ON E.type = ET.Id  
WHERE E.number = 'E567';
```

5.8 Просмотр фамилии и должности сотрудников на текущую дату

```
SELECT E.name AS employee_name, E.position AS employee_position FROM Employees E
```

6 Интерфейс приложения

Интерфейс разработанной системы позволяет использовать весь функционал, описанный выше.

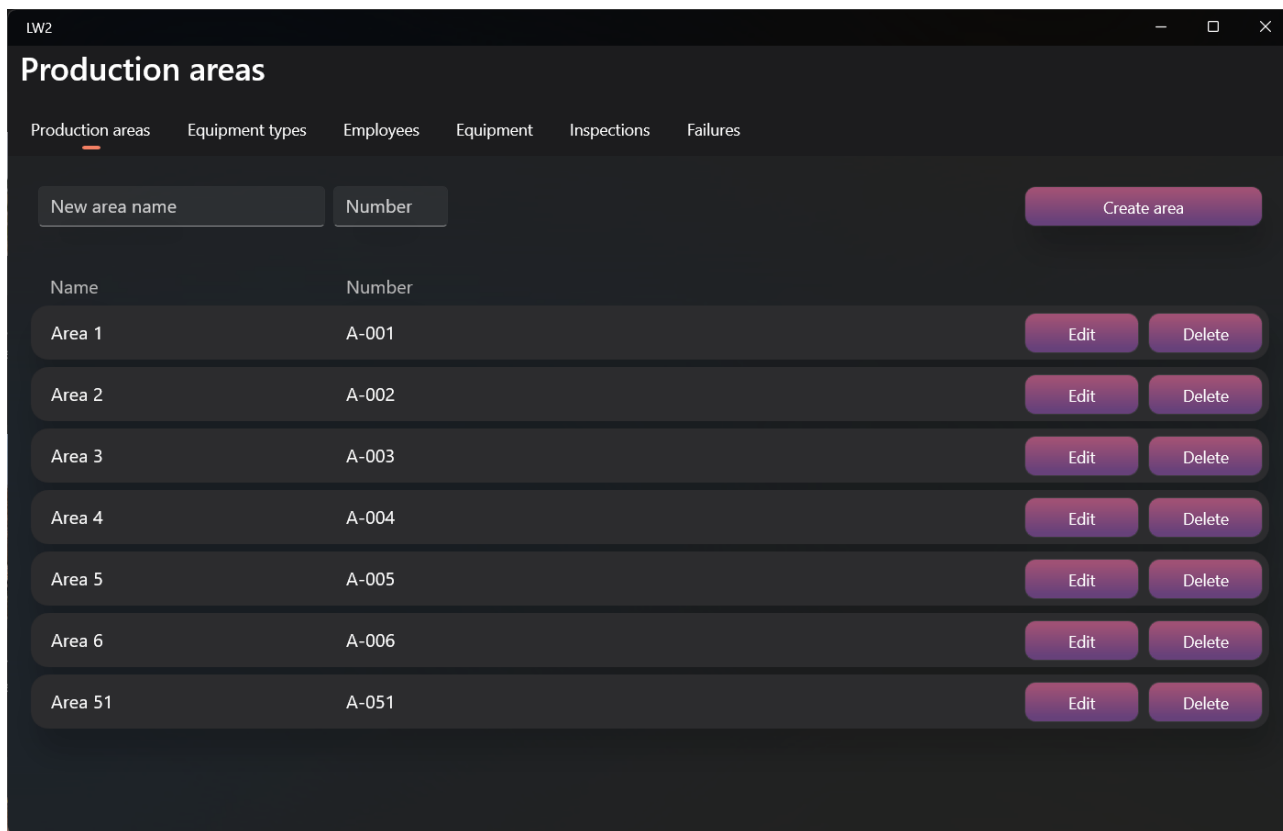


Рис. 6: Окно управления производственными участками

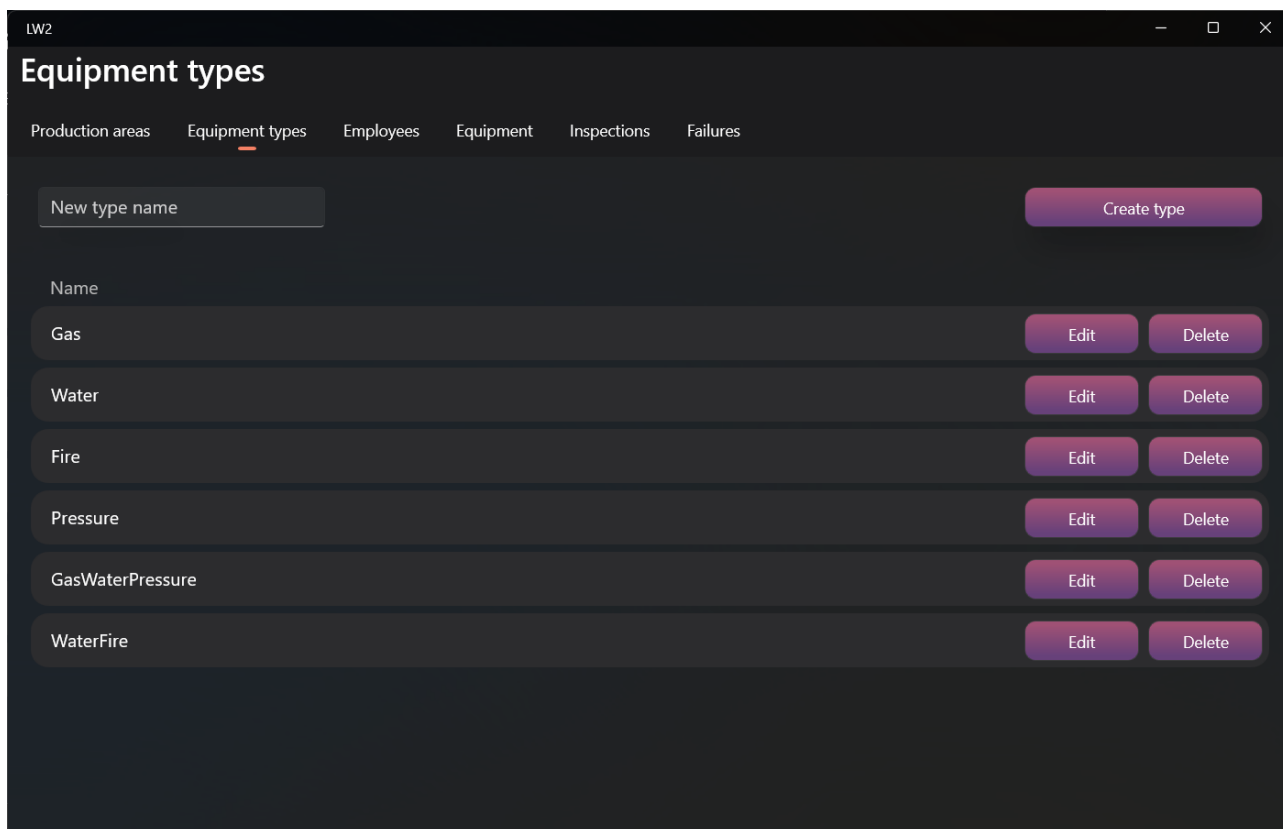


Рис. 7: Окно управления типами оборудования

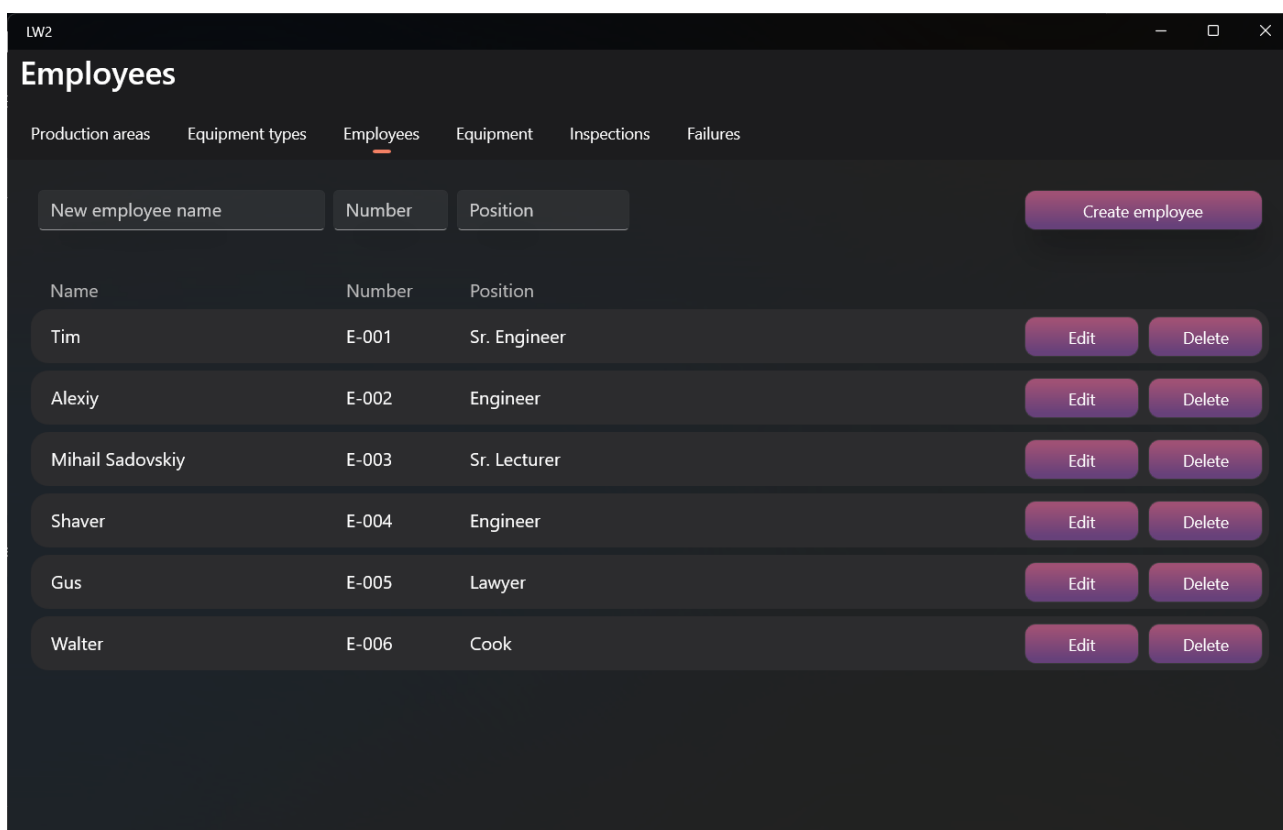


Рис. 8: Окно управления работниками

LW2

Equipment

Production areas

Equipment types

Employees

Equipment

Inspections

Failures

New name

number

Pressure

Area 6

Create equipment

Name	Number	Type	Area		
Stove	100001	Fire	Area 1	Edit	Delete
Oven	100002	Fire	Area 2	Edit	Delete
Press	100003	Pressure	Area 2	Edit	Delete
Roller	100004	GasWaterPressure	Area 51	Edit	Delete

Рис. 9: Окно управления оборудованием

LW2

Inspections

Production areas

Equipment types

Employees

Equipment

Inspections

Failures

New result

Press

Alexiy

*Failure reason

Add inspection

Date and time	Result	Equipment	Employee	*Failure reason		
06.10.2024 09:00	Ok	Stove	Tim		Edit	Delete
06.10.2024 09:00	Ok	Oven	Tim		Edit	Delete
06.10.2024 09:00	Ok	Press	Alexiy		Edit	Delete
06.10.2024 09:00	Fail	Roller	Alexiy	Idk lol	Edit	Delete
06.10.2024 09:00	Ok	Stove	Mihail Sadovskiy		Edit	Delete
06.10.2024 09:00	Fail	Oven	Mihail Sadovskiy	IDC totally	Edit	Delete

Рис. 10: Окно управления осмотрами

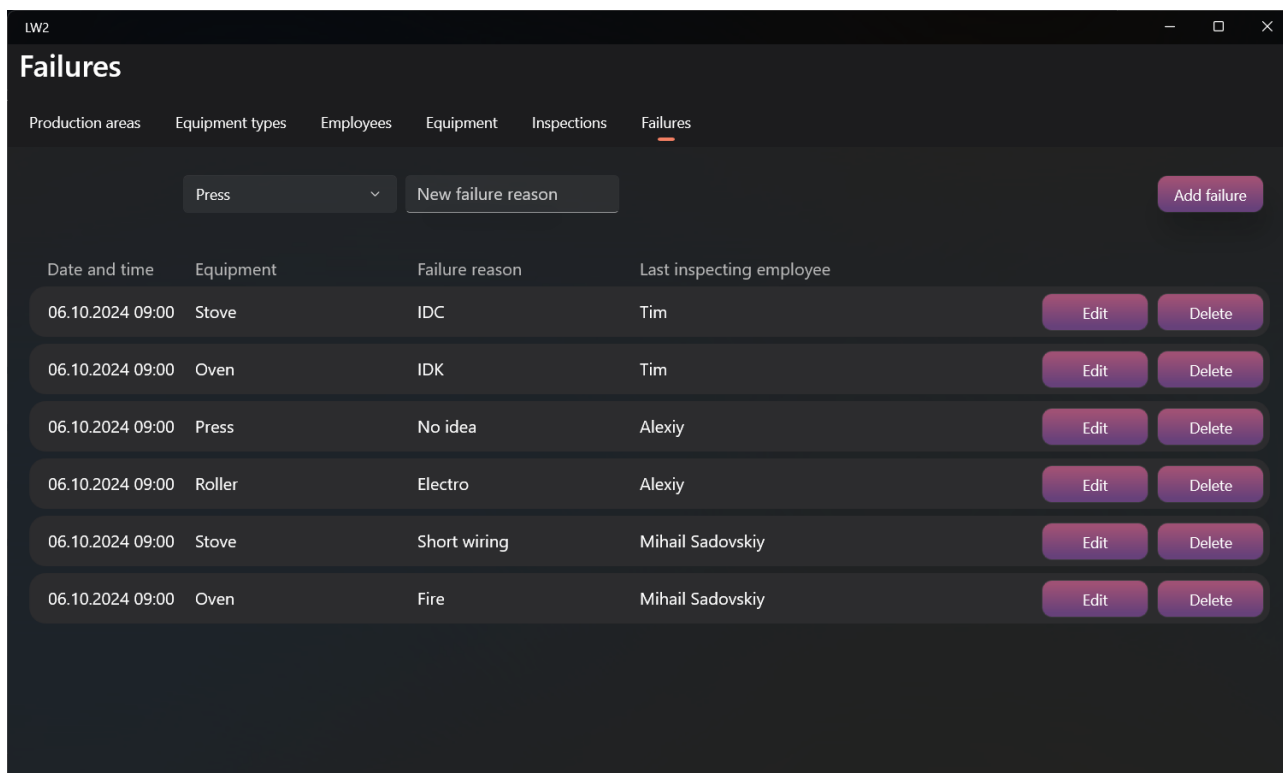


Рис. 11: Окно управления поломками

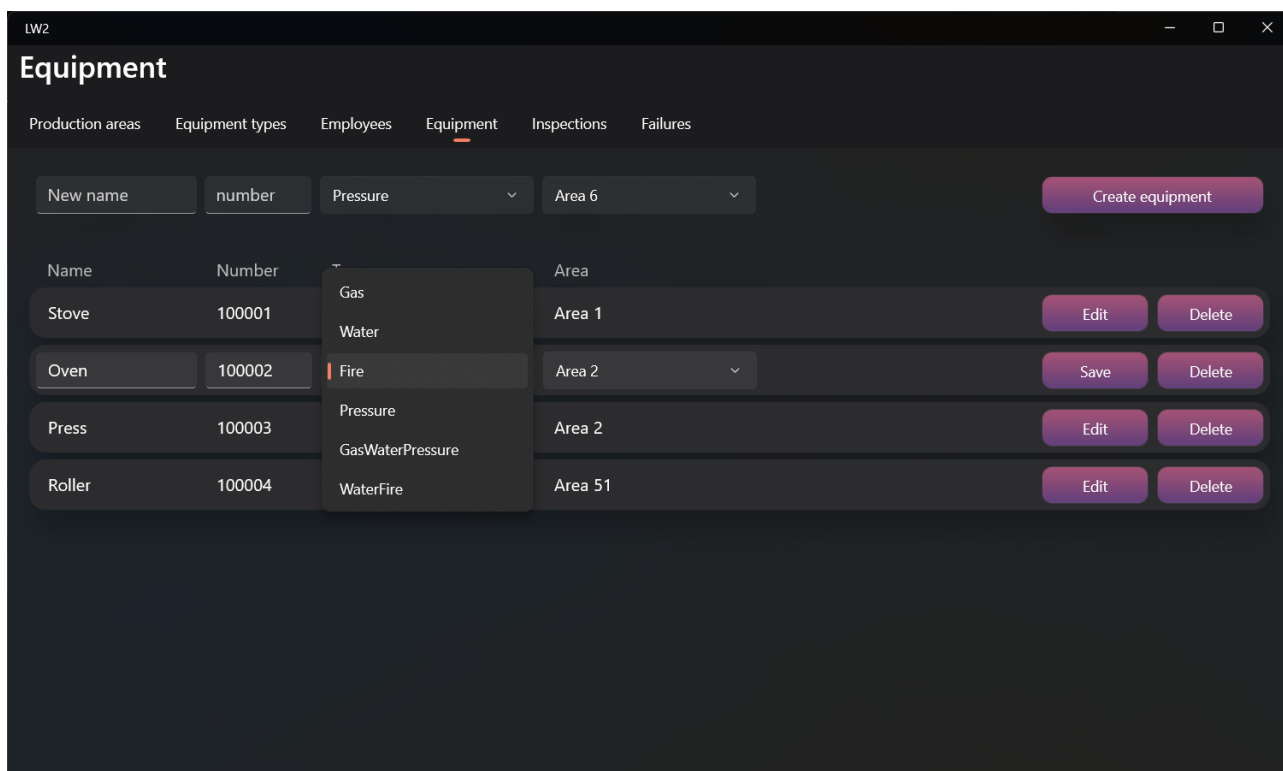


Рис. 12: Редактирование существующих записей в таблице

7 Используемые инструменты

Для создания диаграмм использовались инструменты draw.io, а также редактор BPMN-диаграмм [BPMN.io](https://bpmn.io).

Для создания процедур и их первоначального вызова была использована СУБД MySQL **Workbench**.

Программная реализация выполнена с использованием языка программирования C# и кроссплатформенного UI-фреймворка **.NET MAUI**. Приложение предназначено для работы в операционной системе Windows 10 и новее.