

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Ордена Трудового Красного Знамени федеральное государственное бюджетное
образовательное учреждение высшего образования
«Московский технический университет связи и информатики» (МТУСИ)
Кафедра «Информационные системы и сетевые технологии»

Презентация по защите учебной практике технологической

По теме:

**«Настраиваемая, трехдиапазонная, звуковая сигнализация в
зависимости от веса объекта»**

Подготовил студент группы УБСТ2304
Паокин А.И.

руководитель практики
Кандзюба Е.В.

Москва 2025

Цели и задачи практики

Цель проекта:

Разработка системы контроля веса транспортных средств для предотвращения перегрузки мостов.

Задачи:

- Создание аппаратной части на базе ESP32.
- Разработка базы данных для учёта взвешиваний.
- Реализация графического интерфейса.

Актуальность:

Повышение безопасности и долговечности инфраструктуры.



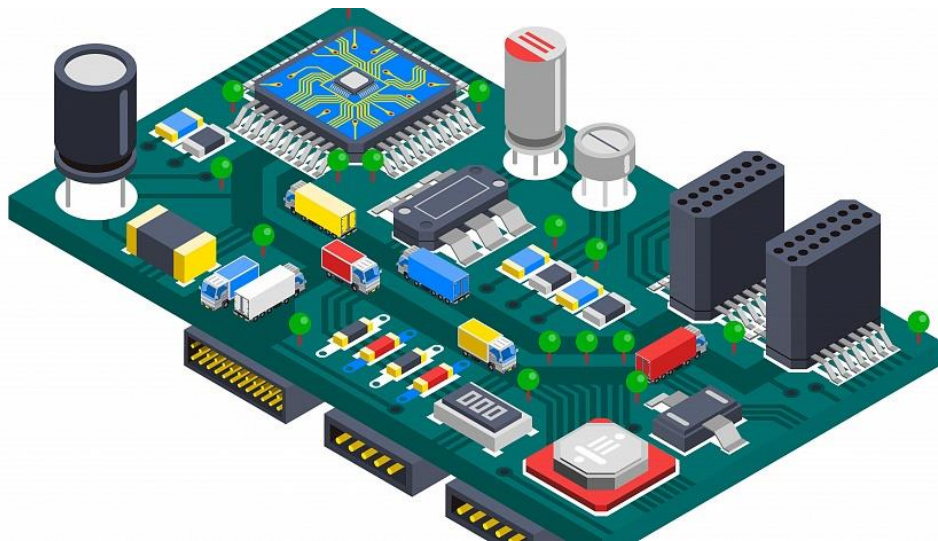
Техника безопасности

Основные требования:

- Проверка оборудования перед началом работы.
- Запрет на самостоятельный ремонт.
- Ограничение непрерывной работы (не более 2 часов).

Рекомендации:

Регулярные перерывы и упражнения для глаз.



Проектирование системы

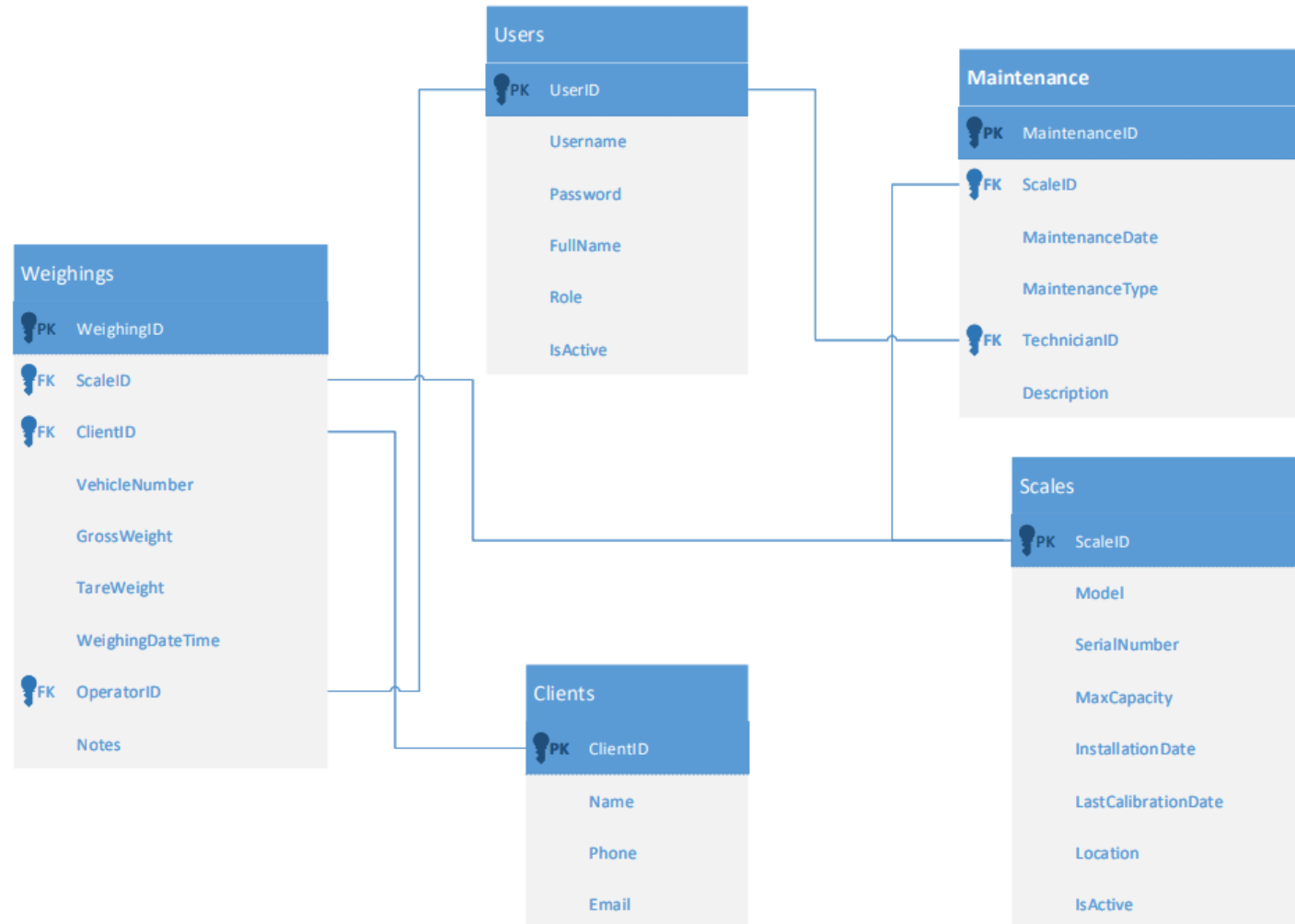
Система учёта весов перед мостами автоматизирует контроль веса грузового транспорта для предотвращения перегрузки мостов.

Основные функции:

- **Взвешивание:** Фиксация массы ТС (брутто/тара), сравнение с нормативами, реакция на перегруз (оповещение, блокировка шлагбаума).
- **Учёт клиентов:** База данных перевозчиков с историей взвешиваний и статистикой.
- **Управление оборудованием:** Цифровые паспорта весовых платформ, контроль поверок и ремонтов.
- **Звуковое оповещение:** Модуль на Arduino анализирует данные с тензодатчиков и подаёт сигнал при перегрузе.
- **Разграничение прав:** Ролевая модель (операторы, техники, менеджеры, администраторы).
- **Аналитика:** Отчёты по нагрузке, статистике перегрузов и загруженности весовых пунктов.

Система оптимизирует процесс взвешивания, обеспечивает безопасность мостов и предоставляет данные для анализа.

Диаграмма «Сущность-связь»



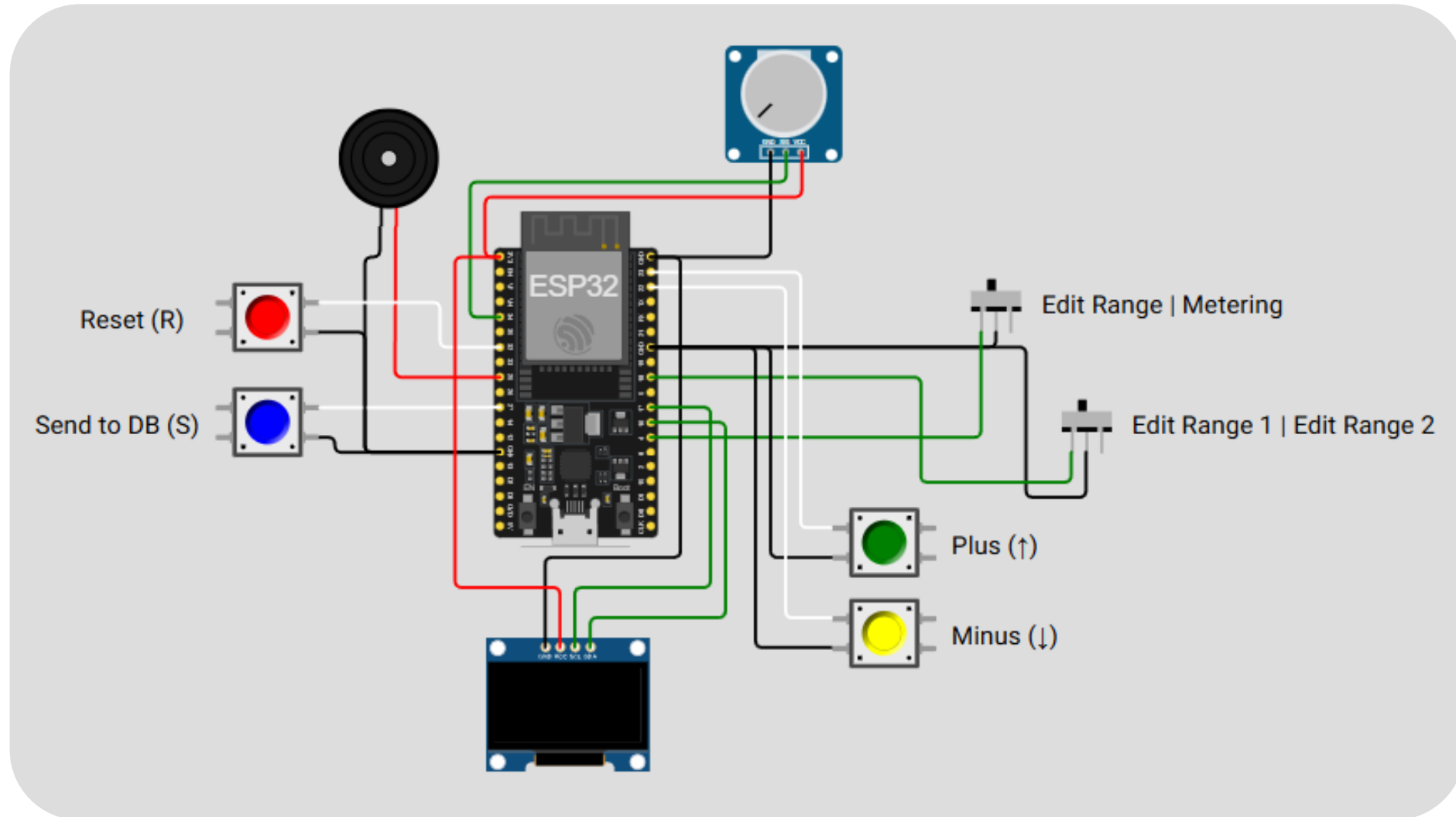
Аппаратная часть

Основные компоненты:

- Микроконтроллер: ESP32 DevKit C v4.
- Датчики: Потенциометр (имитация тензодатчика).
- Звуковая сигнализация: Зуммер с настройкой тона.
- Интерфейс: OLED-дисплей, кнопки управления.



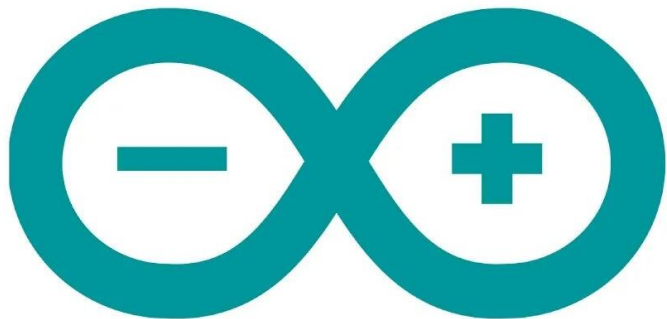
Схема проекта на платформе Arduino



Программная часть

Ключевые функции:

- Обработка данных с датчиков.
- Настройка верхнего и нижнего уровня для звуковых сигналов
- Вывод данных на экран
- Звуковое оповещение при перегрузе.
- Отправка данных в Firebase.



```
if (weight != analogRead(POTENTIOMETER_PIN) * coef)
{
    weight = analogRead(POTENTIOMETER_PIN) * coef;
    display.clearDisplay();
    display.setCursor(0, 0);
    display.println("Weight: ");
    display.println(weight);
    display.display();
    delay(300);
}
```


Программная часть

```
void changeRanges(int typeOperation, volatile int& range, bool isRange1) {
    if (typeOperation == 1 )
    {
        range -= step;
    }
    else if (typeOperation == 2)
    {
        range += step;
    }
    Serial.print(isRange1? "Range1 = ":"Range2 = ");
    Serial.println(range);

    display.clearDisplay();
    display.setCursor(0, 0);
    display.print(isRange1? "Range1 = ":"Range2 = ");
    display.println(range);
    display.display();
    playTone(4);
}
```

```
String uniqueId = String(millis());
HTTPClient http;
String url = "https://" + String(FIREBASE_HOST) + "/sensorData/idScales"
+String(idScales)+"/"+String(uniqueId)+".json?auth=" + String(FIREBASE_AUTH);

String payload = "{\"weight\":" + String(weight) +
                  "\",\"range1\":" + String(range1) +
                  "\",\"range2\":" + String(range2) +
                  "\",\"timestamp\":" + String(uniqueId) +
                  "\",\"event\":" + String(eventType) + "\"}";

http.begin(url);
http.addHeader("Content-Type", "application/json");
```

База данных

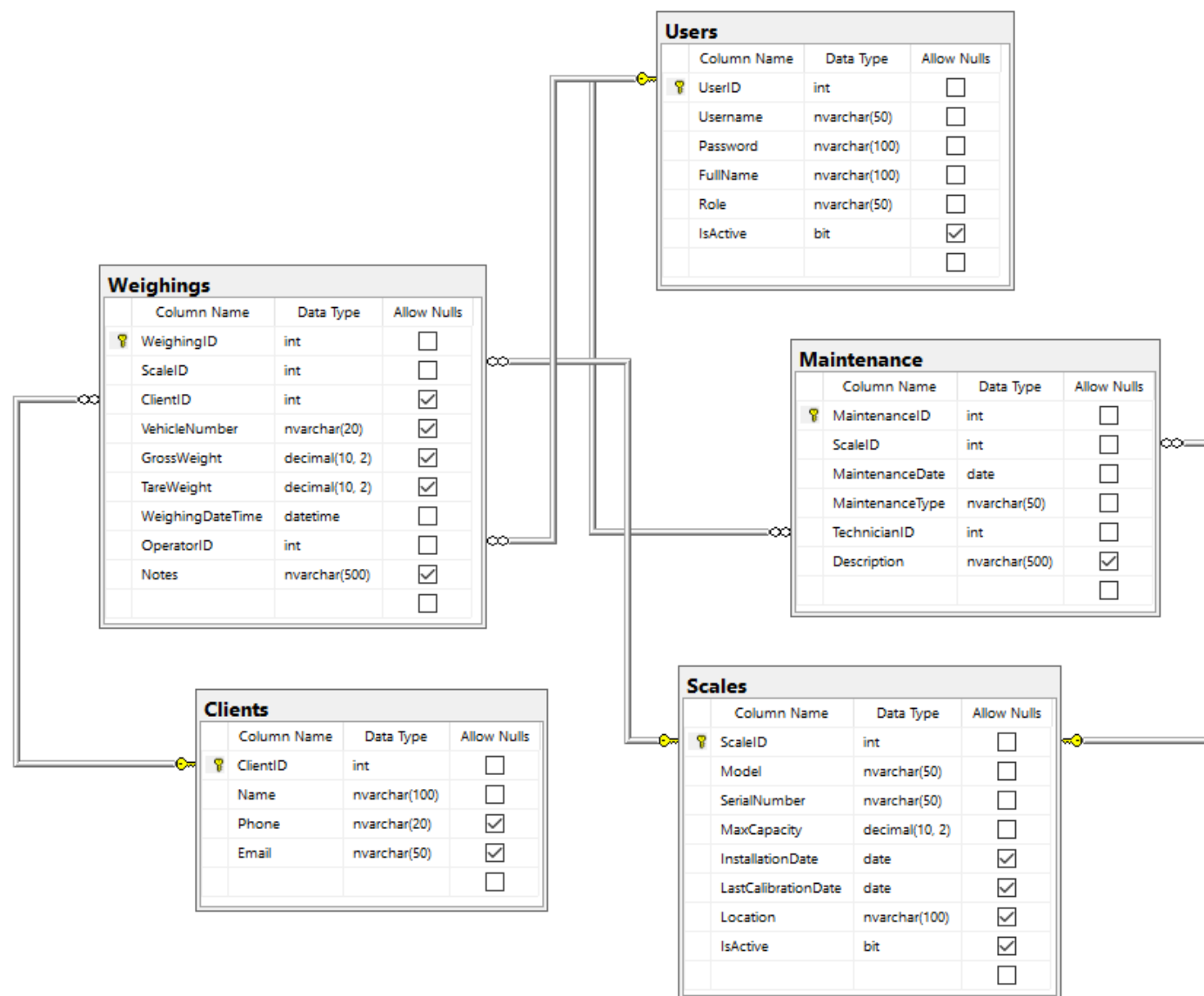
Выбор СУБД: MS SQL Server (надёжность, производительность).

Структура БД:

- Таблицы: Scales, Clients, Weighings, Maintenance.
- Связи: Внешние ключи для целостности данных.



Схема данных



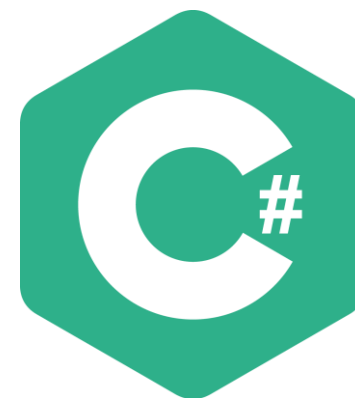
Графический интерфейс и программный код

Технологии:

- Язык: C#, фреймворк: WPF.
- Entity Framework Core

Функционал:

- Авторизация пользователей.
- Просмотр и добавление записей.
- Интеграция с Firebase.
- Работа с MS SQL Server



Примеры программного кода

Ссылка: 1

```
public List<SensorData> GetAllSensorDataById(int Id)
{
    try
    {
        string fullUrl = $" {_firebaseUrl}/{_databasePath}/idScales{Id}.json?auth={_firebaseSecret}";

        HttpResponseMessage response = _httpClient.GetAsync(fullUrl).Result;

        if (response.IsSuccessStatusCode)
        {
            string responseBody = response.Content.ReadAsStringAsync().Result;

            if (!string.IsNullOrEmpty(responseBody) && responseBody != "null")
            {
                using JsonDocument doc = JsonDocument.Parse(responseBody);
                JsonElement root = doc.RootElement;

                var sensorDataList = root.EnumerateObject()
                    .Select(prop => JsonSerializer.Deserialize<SensorData>(prop.Value.GetRawText()))
                    .ToList();

                return sensorDataList!;
            }
        }
    }
}
```

Примеры программного кода

Ссылка: 1

```
private void SaveButton_Click(object sender, RoutedEventArgs e)
{
    if (_sensor == null)
    {
        MessageBox.Show("Весы не выбраны или отсутствуют данные!");
        return;
    }
    if (String.IsNullOrEmpty(VehicleNumberTextBox.Text) && String.IsNullOrEmpty(TareWeightTextBox.Text))
    {
        MessageBox.Show("Заполните данные во всех полях!");
        return;
    }

    _weighing.TareWeight = Convert.ToInt32(TareWeightTextBox.Text);
    _weighing.VehicleNumber = VehicleNumberTextBox.Text;
    _weighing.ClientId = (int)ClientComboBox.SelectedValue;
    _weighing.ScaleId = (int)ScaleComboBox.SelectedValue;
    _weighing.GrossWeight = _sensor.weight;
    _weighing.OperatorId = WindowAuthorization.EnterUser!.UserId;
    _weighing.Notes = NotesTextBox.Text;

    try
    {
        _context.Weighings.Add(_weighing);
        _context.SaveChanges();
        DialogResult = true;
        Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка при сохранении: {ex.Message}", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
    }
}
```

Ссылка: 1

Примеры графического интерфейса

Авторизация

Добро пожаловать

Войдите в свой аккаунт

Логин

Пароль

☐ Запомнить меня

Войти

Новое взвешивание

Обновить данные весов

Выбрать вес:

Клиент:

Номер ТС:

Вес брутто:

*** Data Weight**

Вес тары:

Примечания:

Сохранить **Отмена**

Заключение

В ходе учебной практики был разработан прототип трёхдиапазонной звуковой сигнализации, реагирующей на вес. Система включает:

- Аппаратную часть на ESP32;
- Базу данных для учёта взвешиваний (MS SQL Server);
- Графический интерфейс на C# WPF;
- Интеграцию с Firebase.

Проект успешно решает задачи контроля веса транспорта перед мостами, повышая безопасность. Возможности для развития: добавление датчиков, улучшение аналитики и интерфейса.

Работа позволила закрепить теорию на практике, подтвердив актуальность системы.

Ссылки на материалы и проект

GitHub проекта:

github.com/Fiery-Paks/Praktika_Paokin_UBST_2304

Видео-отчёт:

<https://drive.google.com/file/d/1lH51KnVKd78eVz6V62IIwFOoSB7x1CBE/view>