Производственная практика.

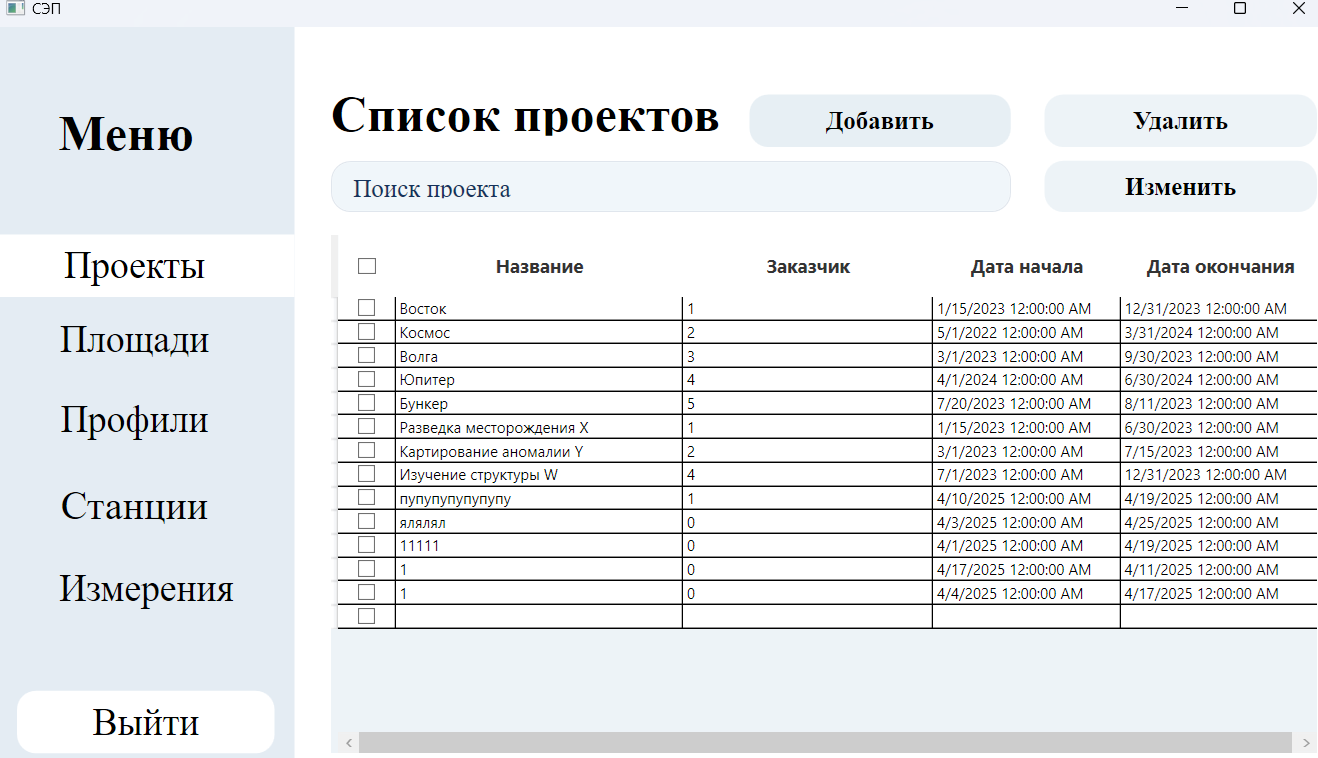
Разработка, администрирование и защита баз данных

Отчет

Тема: «Электроразведочное профилирование на постоянном токе (СЭП).»

Анастасия Бородина Владимировна 2207д2

17.04.2025

Описание окон и функциональных возможностей приложения.  
**1. Проекты**Отображает проекты из БД, можно изменять, добавлять и удалять, совершить поиск.  


Код:

namespace ElectricalProfiling.Views

{

public partial class ProjectPage : Page

{

private ProjectControllerDB projectDB = new();

public ProjectPage()

{

InitializeComponent();

LoadProject();

}

public void LoadProject()

{

List<Projects> listProject = projectDB.GetProject();

Projects\_DataGrid.ItemsSource = listProject;

}

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

{

var addProjectControl = new AddProject();

MainGrid.Children.Add(addProjectControl);

// Позиционируем UserControl по центру

addProjectControl.HorizontalAlignment = HorizontalAlignment.Center;

addProjectControl.VerticalAlignment = VerticalAlignment.Center;

}

private void Find\_TextBox\_KeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

string findTextBox = Find\_TextBox.Text;

if (string.IsNullOrEmpty(findTextBox))

{

LoadProject();

}

else

{

FilterProjects(findTextBox);

}

}

private string \_placeholderText = "Поиск проекта";

private void Find\_TextBox\_GotFocus(object sender, RoutedEventArgs e)

{

if (Find\_TextBox.Text == \_placeholderText)

{

Find\_TextBox.Text = "";

Find\_TextBox.Foreground = Brushes.Black;

}

}

private void Find\_TextBox\_LostFocus(object sender, RoutedEventArgs e)

{

if (string.IsNullOrEmpty(Find\_TextBox.Text))

{

Find\_TextBox.Text = \_placeholderText;

Find\_TextBox.Foreground = new SolidColorBrush(Color.FromRgb(0x15, 0x32, 0x5B)); // Восстанавливаем цвет

}

}

private void FilterProjects(string searchText)

{

// Фильтрация проектов по введённому тексту

List<Projects> filteredProjects = projectDB.GetProject()

.Where(project => project.Name.Contains(searchText, StringComparison.OrdinalIgnoreCase))

.ToList();

Projects\_DataGrid.ItemsSource = filteredProjects;

}

private void Edit\_Project\_Click(object sender, RoutedEventArgs e)

{

var selectedProject = Projects\_DataGrid.SelectedItem as Projects;

if (selectedProject != null)

{

var editProjectControl = new EditProject();

editProjectControl.ProjectName\_TextBox.Text = selectedProject.Name;

editProjectControl.StartDate\_Picker.SelectedDate = selectedProject.Start\_date;

editProjectControl.EndDate\_Picker.SelectedDate = selectedProject.End\_date;

MainGrid.Children.Add(editProjectControl);

// Позиционируем UserControl по центру

editProjectControl.HorizontalAlignment = HorizontalAlignment.Center;

editProjectControl.VerticalAlignment = VerticalAlignment.Center;

}

else

{

MessageBox.Show("Выберите проект для редактирования.");

}

}

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

{

var selectedProject = Projects\_DataGrid.SelectedItem as Projects;

if (selectedProject != null)

{

string name = selectedProject.Name;

projectDB.DeleteProject(name);

MessageBox.Show("Проект удален!");

LoadProject();

}

else

{

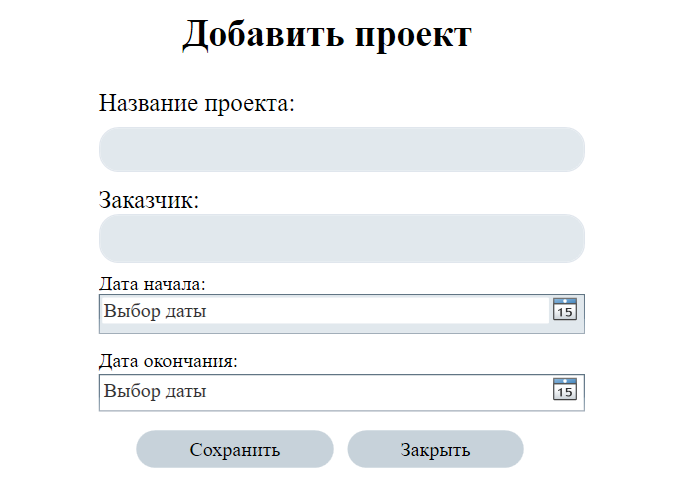
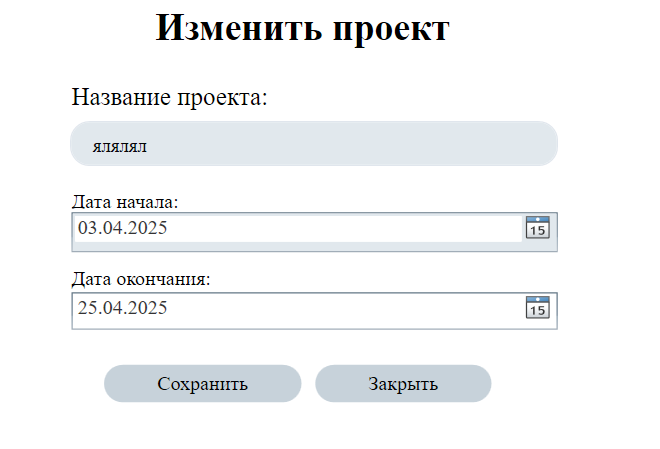
MessageBox.Show("Выберите проект для удаления");

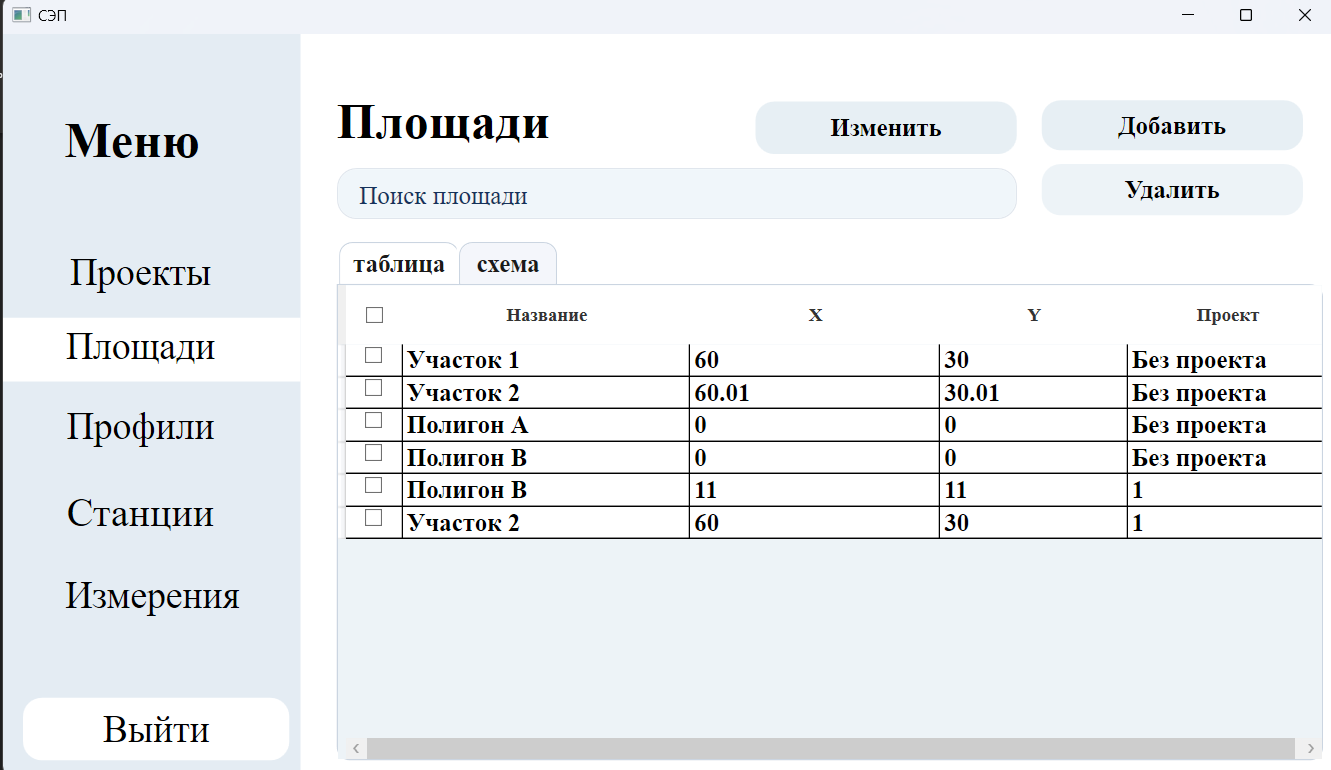
}

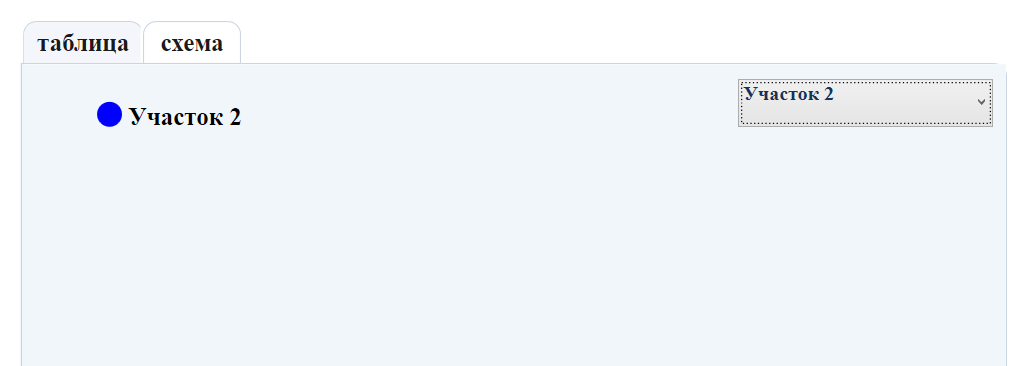
}

}

}

2. Площади  
Отображает площади из БД, можно изменять, добавлять и удалять, можно совершить поиск. Во вкладке схема просмотреть график расположение площади.  




Код:

namespace ElectricalProfiling.Views

{

public partial class AreasPage : Page

{

public AreasPage()

{

InitializeComponent();

LoadArea();

LoadComboBox();

}

public void LoadArea()

{

using (var db = new ApplicationContext())

{

areas = db.Area.ToList()

.Select(area =>

{

var project = db.Project.FirstOrDefault(p => p.Id == area.Project\_ID);

var coordinate = db.AreaCoordinate.FirstOrDefault(c => c.area\_ID == area.ID);

return new AreaView

{

Name = area.Name,

Project = project != null ? project.Name : "Без проекта", //РЕШИТЬ ПРОБЛЕМУ. БЕЗ ПРОЕКТА

X = coordinate?.X ?? 0,

Y = coordinate?.Y ?? 0

};

}).ToList();

Area\_DataGrid.ItemsSource = areas;

}

} //очень важно создать класс для визуализации DataGrid(AreaView)

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

{

var addAreaControl = new AddArea();

MainGrid.Children.Add(addAreaControl);

addAreaControl.HorizontalAlignment = HorizontalAlignment.Center;

addAreaControl.VerticalAlignment = VerticalAlignment.Center;

}

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

{

var selectedArea = Area\_DataGrid.SelectedItem as AreaView;

if (selectedArea != null)

{

using (var db = new ApplicationContext())

{

var areaToDelete = db.Area.FirstOrDefault(a => a.Name == selectedArea.Name);

if (areaToDelete != null)

{

var coordinates = db.AreaCoordinate.Where(c => c.area\_ID == areaToDelete.ID);

db.AreaCoordinate.RemoveRange(coordinates);

db.Area.Remove(areaToDelete);

db.SaveChanges();

MessageBox.Show("Площадь успешно удалена!");

LoadArea();

}

}

}

else

{

MessageBox.Show("Выберите площадь для удаления");

}

}

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

{

var selectedArea = Area\_DataGrid.SelectedItem as AreaView;

if (selectedArea != null)

{

var editAreaControl = new EditArea();

editAreaControl.AreaName\_TextBox.Text = selectedArea.Name;

editAreaControl.X\_TextBox.Text = selectedArea.X.ToString();

editAreaControl.Y\_TextBox.Text = selectedArea.Y.ToString();

editAreaControl.ProjectName\_TextBox.Text = selectedArea.Project;

MainGrid.Children.Add(editAreaControl);

// Позиционируем UserControl по центру

editAreaControl.HorizontalAlignment = HorizontalAlignment.Center;

editAreaControl.VerticalAlignment = VerticalAlignment.Center;

}

else

{

MessageBox.Show("Выберите площадь для редактирования.");

}

}

private List<AreaView> areas;

public void LoadComboBox()

{

AreaSelector.ItemsSource = areas;

AreaSelector.DisplayMemberPath = "Name"; // Для отображения только имени

AreaSelector.SelectedValuePath = "Id"; // Для выбора по Id

Area\_DataGrid.ItemsSource = areas;

}

private void AreaSelector\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (AreaSelector.SelectedItem is AreaView selectedArea)

{

DrawArea(selectedArea);

}

}

private void DrawArea(AreaView area)

{

// Очищаем старые элементы на Canvas

AreaCanvas.Children.Clear();

// Создаем новый элемент для отображения площади (например, круг)

var ellipse = new Ellipse

{

Width = 20,

Height = 20,

Fill = Brushes.Blue

};

// Размещение по координатам X и Y

Canvas.SetLeft(ellipse, area.X);

Canvas.SetTop(ellipse, area.Y);

// Добавляем элемент на Canvas

AreaCanvas.Children.Add(ellipse);

// Добавляем текстовое описание площади

var textBlock = new TextBlock

{

Text = area.Name,

Foreground = Brushes.Black

};

Canvas.SetLeft(textBlock, area.X + 25);

Canvas.SetTop(textBlock, area.Y);

// Добавляем текст на Canvas

AreaCanvas.Children.Add(textBlock);

}

private string \_placeholderText = "Поиск площади";

private void Find\_TextBox\_GotFocus(object sender, RoutedEventArgs e)

{

Find\_TextBox.Text = "";

Find\_TextBox.Foreground = Brushes.Black;

}

private void Find\_TextBox\_LostFocus(object sender, RoutedEventArgs e)

{

if (string.IsNullOrEmpty(Find\_TextBox.Text))

{

Find\_TextBox.Text = \_placeholderText;

Find\_TextBox.Foreground = new SolidColorBrush(Color.FromRgb(0x15, 0x32, 0x5B));

}

}

private void Find\_TextBox\_KeyUp(object sender, System.Windows.Input.KeyEventArgs e)

{

string searchText = Find\_TextBox.Text.Trim();

if (string.IsNullOrEmpty(searchText))

{

LoadArea();

}

else

{

FilterAreas(searchText);

}

}

private async void FilterAreas(string searchText)

{

using (var db = new ApplicationContext())

{

var filteredAreas = await db.Area

.Where(area => area.Name.Contains(searchText) ||

db.Project.Any(p => p.Id == area.Project\_ID && p.Name.Contains(searchText)))

.Select(area => new AreaView

{

Name = area.Name,

Project = db.Project.FirstOrDefault(p => p.Id == area.Project\_ID).Name,

X = db.AreaCoordinate.FirstOrDefault(c => c.area\_ID == area.ID).X,

Y = db.AreaCoordinate.FirstOrDefault(c => c.area\_ID == area.ID).Y

})

.ToListAsync();

Area\_DataGrid.ItemsSource = filteredAreas;

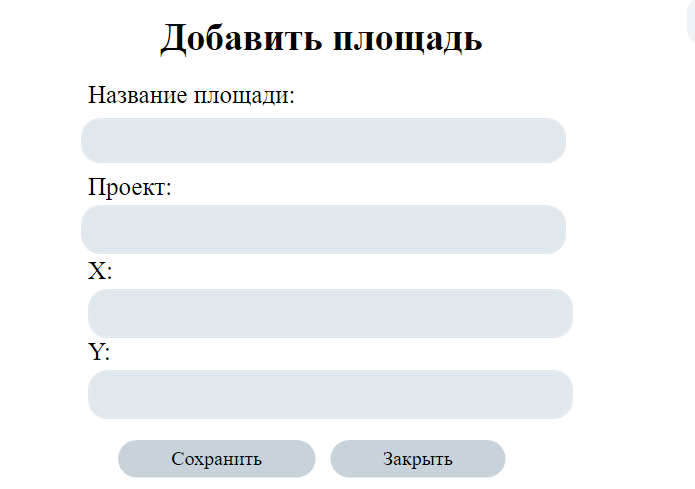
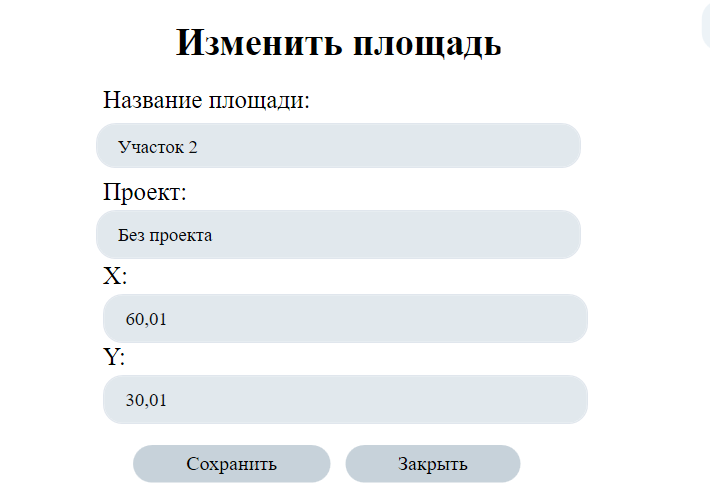
}

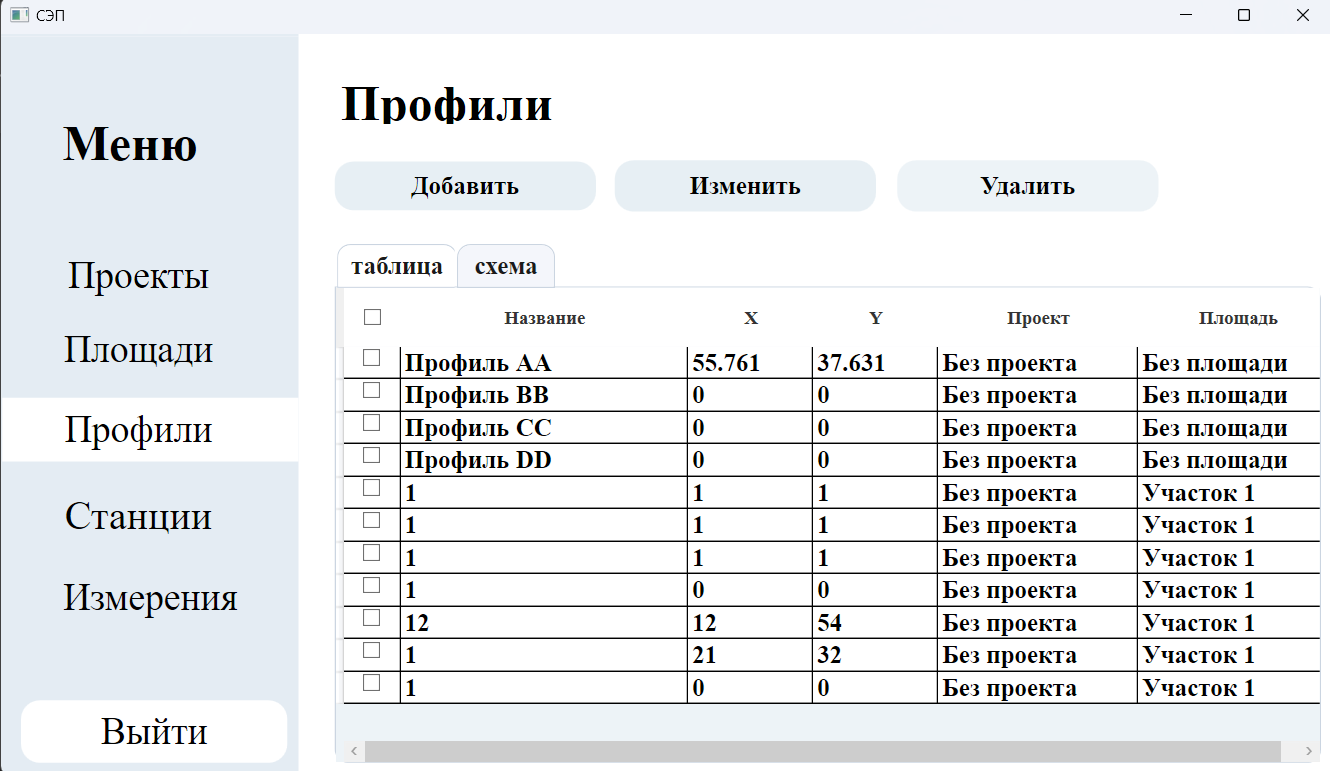
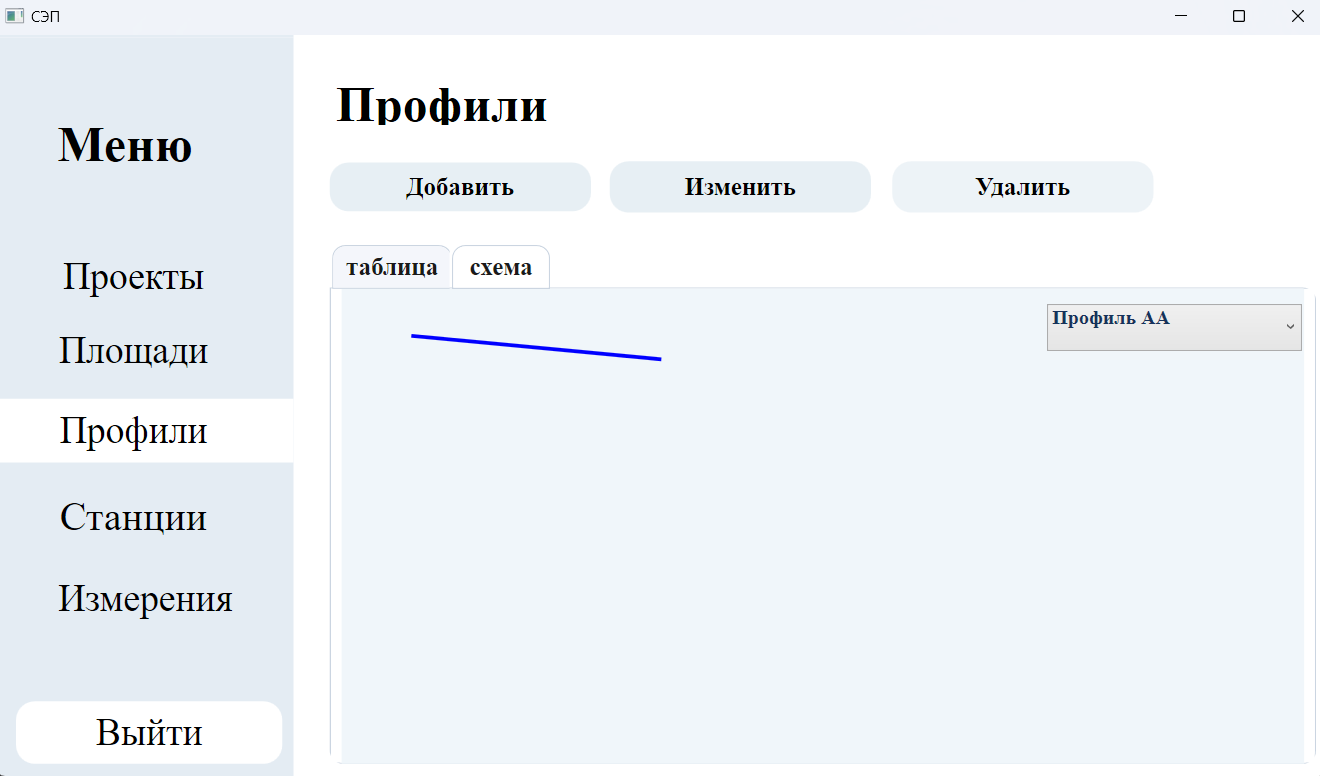
}

}

}

}

3. Профили  
Отображает профили из БД, можно изменять, добавлять и удалять. Можно просмотреть график с прямой. Расстояние между точками.  
  
  
  
Код:

namespace ElectricalProfiling.Views

{

public partial class ProfilePage : Page

{

public ProfilePage()

{

InitializeComponent();

LoadProfile();

LoadComboBox();

}

public void LoadProfile()

{

using (var db = new ApplicationContext())

{

this.profiles = db.Profile.ToList()

.Select(profile =>

{

var area = db.Area.FirstOrDefault(a => a.ID == profile.Area\_ID);

var project = area != null ? db.Project.FirstOrDefault(p => p.Id == area.Project\_ID) : null;

var coordinate = db.ProfileCoordinate.FirstOrDefault(c => c.Profile\_ID == profile.ID && c.Point\_Type == "start");

return new ProfileView

{

Name = profile.Name,

Project = project?.Name ?? "Без проекта",

Area = area?.Name ?? "Без площади",

X = coordinate?.X ?? 0,

Y = coordinate?.Y ?? 0

};

}).ToList();

Profile\_DataGrid.ItemsSource = this.profiles;

}

}

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

{

var selectedProfile = Profile\_DataGrid.SelectedItem as ProfileView;

if (selectedProfile != null)

{

using (var db = new ApplicationContext())

{

var profileToDelete = db.Profile.FirstOrDefault(p => p.Name == selectedProfile.Name);

if (profileToDelete != null)

{

var coordinates = db.ProfileCoordinate.Where(c => c.Profile\_ID == profileToDelete.ID);

db.ProfileCoordinate.RemoveRange(coordinates);

db.Profile.Remove(profileToDelete);

db.SaveChanges();

MessageBox.Show("Профиль успешно удалён!");

LoadProfile();

}

}

}

else

{

MessageBox.Show("Выберите профиль для удаления");

}

}

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

{

var selectedProfile = Profile\_DataGrid.SelectedItem as ProfileView;

if (selectedProfile != null)

{

var editProfileControl = new EditProfile();

editProfileControl.ProfileName\_TextBox.Text = selectedProfile.Name;

editProfileControl.AreaName\_TextBox.Text = selectedProfile.Area;

editProfileControl.X\_TextBox.Text = selectedProfile.X.ToString();

editProfileControl.Y\_TextBox.Text = selectedProfile.Y.ToString();

editProfileControl.ProjectName\_TextBox.Text = selectedProfile.Project;

MainGrid.Children.Add(editProfileControl);

// Позиционируем UserControl по центру

editProfileControl.HorizontalAlignment = HorizontalAlignment.Center;

editProfileControl.VerticalAlignment = VerticalAlignment.Center;

}

else

{

MessageBox.Show("Выберите профиль для редактирования.");

}

}

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

{

var addAreaControl = new AddProfile();

MainGrid.Children.Add(addAreaControl);

addAreaControl.HorizontalAlignment = HorizontalAlignment.Center;

addAreaControl.VerticalAlignment = VerticalAlignment.Center;

}

private List<ProfileView> profiles;

public void LoadComboBox()

{

ProfileSelector.ItemsSource = profiles;

ProfileSelector.DisplayMemberPath = "Name"; // Для отображения только имени

ProfileSelector.SelectedValuePath = "Id"; // Для выбора по Id

Profile\_DataGrid.ItemsSource = profiles;

}

private void ProfileSelector\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (ProfileSelector.SelectedItem is ProfileView selectedProfile)

{

DrawProfile(selectedProfile);

}

}

private void DrawProfile(ProfileView profile)

{

// Очистить холст

ProfileCanvas.Children.Clear();

// Используем координаты из profile (для примера, по оси X будет фиксированное значение)

double startX = profile.X; // Начальная точка X

double startY = profile.Y; // Начальная точка Y

double endX = startX + 200; // Конечная точка X (добавляем фиксированное смещение по оси X)

double endY = startY + (startY / 2); // Конечная точка Y (половина от начальной Y)

// Нарисовать линию между точками

var line = new System.Windows.Shapes.Line

{

X1 = startX,

Y1 = startY,

X2 = endX,

Y2 = endY,

Stroke = System.Windows.Media.Brushes.Blue,

StrokeThickness = 3

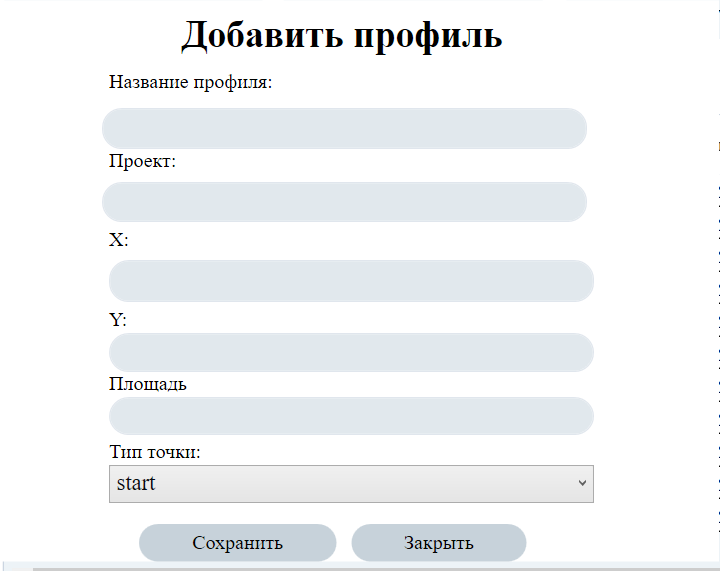
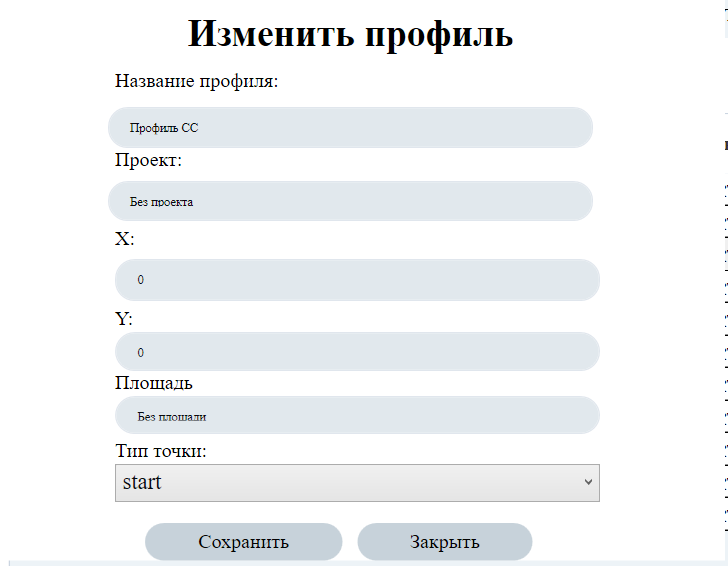
};

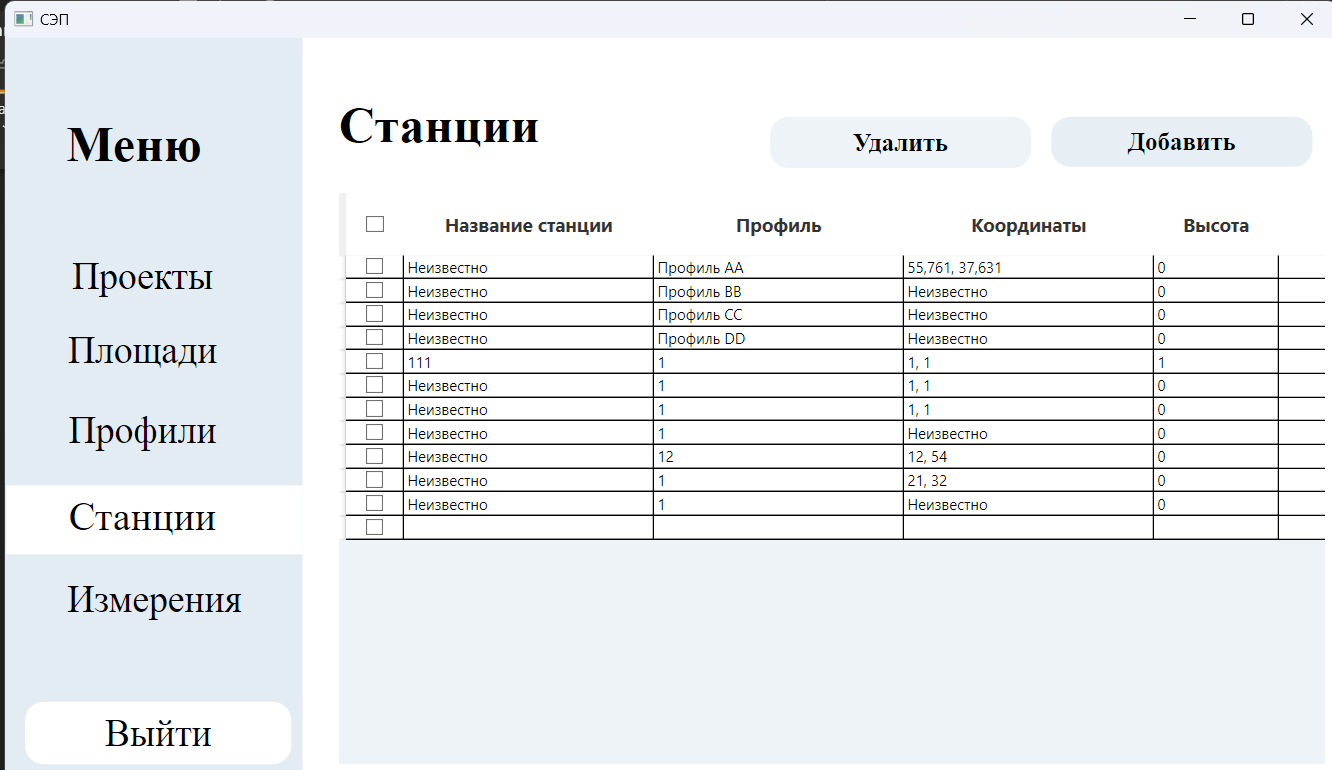
ProfileCanvas.Children.Add(line);

}

}

}



4. Станции  
Отображает станции из БД, можно добавлять, удалять.  
  
Код:

namespace ElectricalProfiling.Views

{

public partial class StationsPage : Page

{

public StationsPage()

{

InitializeComponent();

LoadStantion();

}

public void LoadStantion()

{

using (var db = new ApplicationContext())

{

var stations = db.Profile.ToList()

.Select(profile =>

{

var area = db.Area.FirstOrDefault(a => a.ID == profile.Area\_ID);

var project = area != null ? db.Project.FirstOrDefault(p => p.Id == area.Project\_ID) : null;

var coordinate = db.ProfileCoordinate.FirstOrDefault(c => c.Profile\_ID == profile.ID && c.Point\_Type == "start");

var station = db.Stations.FirstOrDefault(s => s.Profile\_ID == profile.ID);

return new StationView

{

StationNames = station != null ? station.station\_name : "Неизвестно",

ProfileName = profile.Name,

Coordinates = coordinate != null ? $"{coordinate.X}, {coordinate.Y}" : "Неизвестно",

Elevation = station != null ? station.Elevation : 0

};

}).ToList();

Stations\_DataGrid.ItemsSource = stations;

}

}

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

{

var selectedStation = Stations\_DataGrid.SelectedItem as StationView;

if (selectedStation != null)

{

using (var db = new ApplicationContext())

{

// Ищем станцию по имени профиля

var profile = db.Profile.FirstOrDefault(p => p.Name == selectedStation.ProfileName);

if (profile != null)

{

// Ищем соответствующую станцию

var stationToDelete = db.Stations.FirstOrDefault(s => s.Profile\_ID == profile.ID);

if (stationToDelete != null)

{

db.Stations.Remove(stationToDelete);

db.SaveChanges();

MessageBox.Show("Станция успешно удалена!");

LoadStantion();

}

else

{

MessageBox.Show("Станция не найдена.");

}

}

else

{

MessageBox.Show("Профиль для этой станции не найден.");

}

}

}

else

{

MessageBox.Show("Выберите станцию для удаления.");

}

}

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

{

var addAreaControl = new AddStation();

MainGrid.Children.Add(addAreaControl);

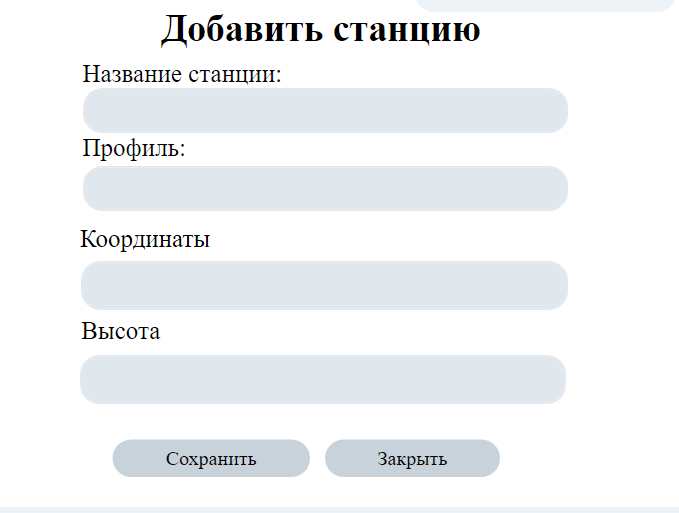
addAreaControl.HorizontalAlignment = HorizontalAlignment.Center;

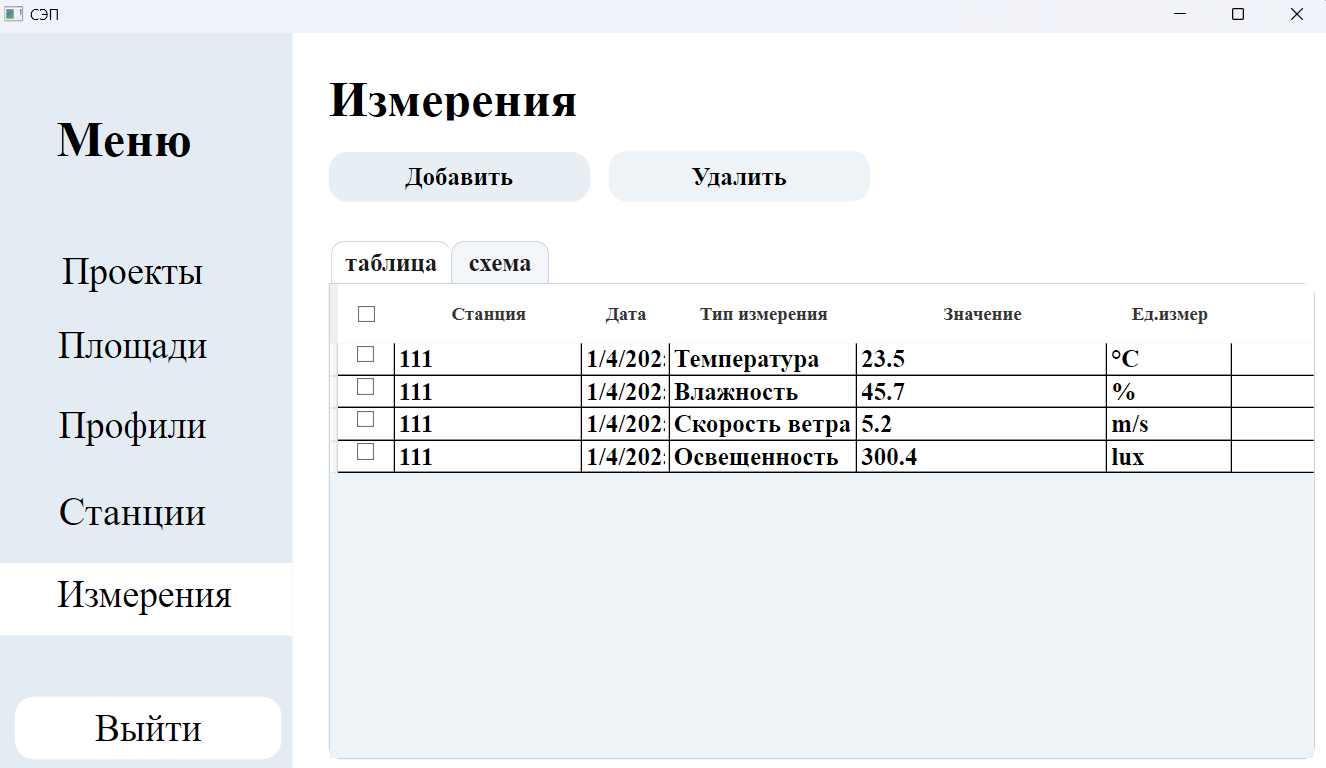
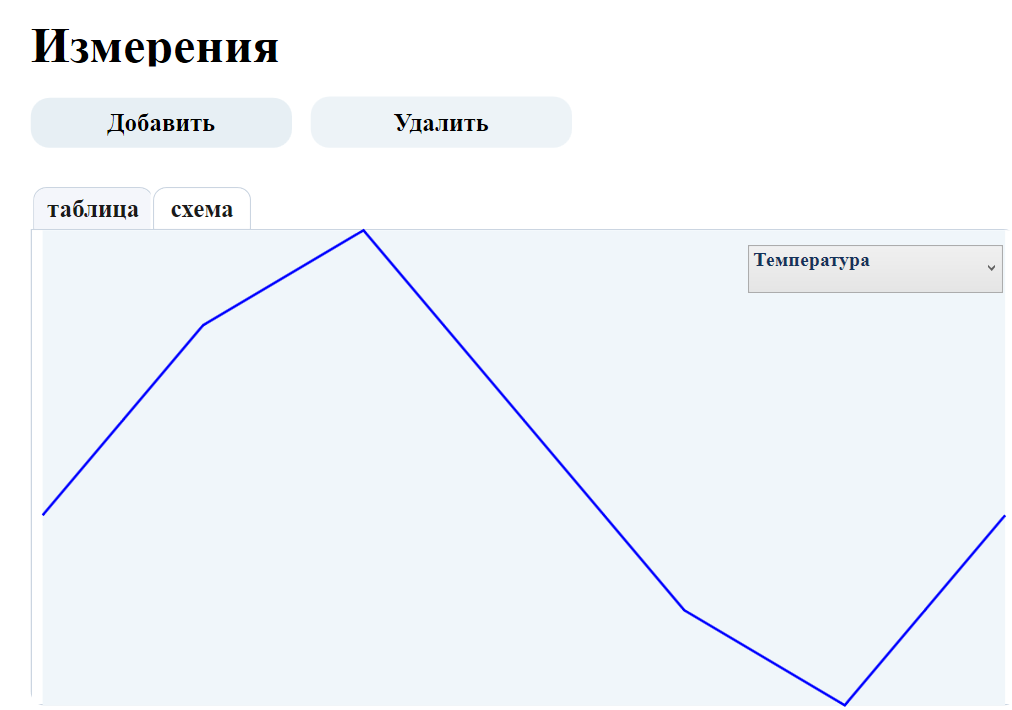
addAreaControl.VerticalAlignment = VerticalAlignment.Center;

}

}

}



5. Измерения  
Отображает измерения из БД, можно добавлять, удалять. Просмотреть во вкладке схема график с измерениями.  
  


Код:

namespace ElectricalProfiling.Views

{

public partial class MeasurementsPage : Page

{

public MeasurementsPage()

{

InitializeComponent();

LoadMeasurements();

LoadComboBox();

}

private List<MeasurementView> measurements = new List<MeasurementView>();

public void LoadMeasurements()

{

using (var db = new ApplicationContext())

{

measurements = db.Measurement.ToList()

.Select(measurement =>

{

var station = db.Stations.FirstOrDefault(s => s.ID == measurement.Station\_ID);

return new MeasurementView

{

StationName = station?.station\_name ?? "Неизвестно",

MeasurementDate = measurement.Date ?? DateTime.MinValue,

MeasurementType = measurement.measurement\_type ?? "Неизвестно",

Value = measurement.Value ?? 0.0,

Unit = measurement.Units ?? "Неизвестно",

ID = measurement.ID

};

}).ToList();

}

Measurements\_DataGrid.ItemsSource = measurements;

}

public void LoadComboBox()

{

MeasurementsSelector.ItemsSource = measurements;

MeasurementsSelector.DisplayMemberPath = "MeasurementType";

MeasurementsSelector.SelectedValuePath = "ID";

if (measurements.Any())

{

MeasurementsSelector.SelectedIndex = 0;

}

}

private void TypeMeasurementsSelector\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (MeasurementsSelector.SelectedItem is MeasurementView selectedMeasurement)

{

DrawChart(selectedMeasurement);

}

}

private void DrawChart(MeasurementView selectedMeasurement)

{

MeasurementsCanvas.Children.Clear(); // Очистить Canvas перед рисованием

// Пример данных для каждого типа измерения

List<double> dataPoints = new List<double>();

string measurementType = selectedMeasurement.MeasurementType;

// Заполняем данные в зависимости от типа измерения

if (measurementType == "Температура")

{

// Примерные данные для температуры

dataPoints = new List<double> { 20, 22, 23, 21, 19, 18, 20 };

}

else if (measurementType == "Давление")

{

// Примерные данные для давления

dataPoints = new List<double> { 1013, 1015, 1012, 1010, 1013, 1014 };

}

else if (measurementType == "Влажность")

{

// Примерные данные для влажности

dataPoints = new List<double> { 60, 62, 65, 70, 75, 80 };

}

else

{

MessageBox.Show("Неизвестный тип измерения.");

return;

}

// Параметры графика

double canvasWidth = MeasurementsCanvas.ActualWidth;

double canvasHeight = MeasurementsCanvas.ActualHeight;

double maxDataValue = dataPoints.Max();

double minDataValue = dataPoints.Min();

// Масштабирование данных

double scaleX = canvasWidth / (dataPoints.Count - 1); // Расстояние по оси X

double scaleY = canvasHeight / (maxDataValue - minDataValue); // Масштаб по оси Y

// Создание линии для графика

Polyline graphLine = new Polyline

{

Stroke = Brushes.Blue,

StrokeThickness = 2

};

// Добавление точек на график

for (int i = 0; i < dataPoints.Count; i++)

{

double x = i \* scaleX; // Координата по X

double y = canvasHeight - (dataPoints[i] - minDataValue) \* scaleY; // Координата по Y

graphLine.Points.Add(new Point(x, y));

}

// Добавление линии на Canvas

MeasurementsCanvas.Children.Add(graphLine);

}

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

{

var selectedMeasurement = Measurements\_DataGrid.SelectedItem as MeasurementView;

if (selectedMeasurement != null)

{

using (var db = new ApplicationContext())

{

var measurementToDelete = db.Measurement.FirstOrDefault(m => m.ID == selectedMeasurement.ID);

if (measurementToDelete != null)

{

db.Measurement.Remove(measurementToDelete);

db.SaveChanges();

MessageBox.Show("Измерение успешно удалено!");

LoadMeasurements();

}

}

}

else

{

MessageBox.Show("Выберите измерение для удаления");

}

}

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

{

var addMeasurementsControl = new AddMeasurements();

MainGrid.Children.Add(addMeasurementsControl);

addMeasurementsControl.HorizontalAlignment = HorizontalAlignment.Center;

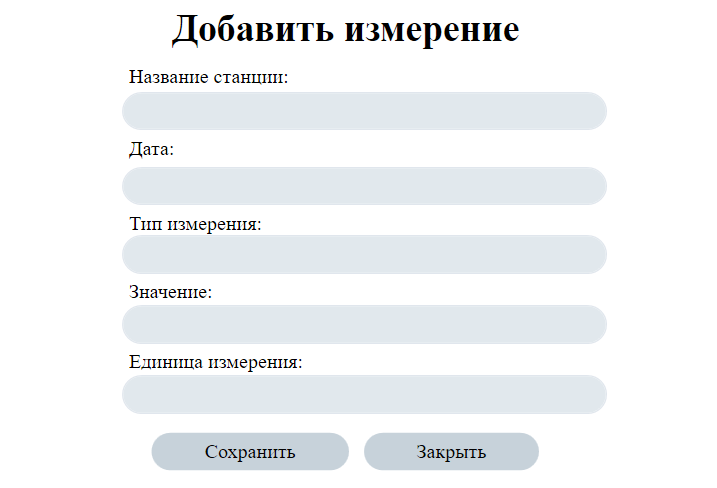
addMeasurementsControl.VerticalAlignment = VerticalAlignment.Center;

}

}

}

}



6. T-SQL query.  
6.1. Код добавления таблиц.  
use [ElectricalProfiling]

-- Таблица Заказчик (Customer)

CREATE TABLE Customer (

ID INT PRIMARY KEY IDENTITY(1,1),

company\_name NVARCHAR(255) NOT NULL,

phone\_number NVARCHAR(50)

);

-- Таблица Проект (Project)

CREATE TABLE Project (

ID INT PRIMARY KEY IDENTITY(1,1),

name NVARCHAR(255) NOT NULL,

customer\_id INT FOREIGN KEY REFERENCES Customer(ID),

start\_date DATE,

end\_date DATE

);

-- Таблица Площадь исследований (Area)

CREATE TABLE Area (

ID INT PRIMARY KEY IDENTITY(1,1),

name NVARCHAR(255) NOT NULL,

project\_id INT FOREIGN KEY REFERENCES Project(ID)

);

-- Таблица Координаты площади (AreaCoordinate)

CREATE TABLE AreaCoordinate (

ID INT PRIMARY KEY IDENTITY(1,1),

area\_id INT FOREIGN KEY REFERENCES Area(ID),

X FLOAT NOT NULL,

Y FLOAT NOT NULL

);

-- Таблица Профиль (Profile)

CREATE TABLE Profile (

ID INT PRIMARY KEY IDENTITY(1,1),

area\_id INT FOREIGN KEY REFERENCES Area(ID),

name NVARCHAR(255) NOT NULL

);

-- Таблица Координаты профиля (ProfileCoordinate)

CREATE TABLE ProfileCoordinate (

ID INT PRIMARY KEY IDENTITY(1,1),

profile\_id INT FOREIGN KEY REFERENCES Profile(ID),

X FLOAT NOT NULL,

Y FLOAT NOT NULL,

point\_type NVARCHAR(50) CHECK (point\_type IN ('start', 'end', 'breakpoint'))

);

-- Таблица Станция (Station)

CREATE TABLE Stations (

ID INT PRIMARY KEY IDENTITY(1,1),

profile\_id INT FOREIGN KEY REFERENCES Profile(ID),

coordinates NVARCHAR(100), -- координаты местоположения станции

elevation FLOAT, -- высота станции над уровнем моря

station\_name NVARCHAR(100) -- имя станции

);

-- Таблица Измерения (Measurement)

CREATE TABLE Measurement (

ID INT PRIMARY KEY IDENTITY(1,1),

station\_id INT FOREIGN KEY REFERENCES Station(ID),

date DATETIME,

measurement\_type NVARCHAR(100),

value FLOAT, --значение

units NVARCHAR(50) --единица измерения

);