```
using Microsoft.Data.Sqlite;
using Microsoft.Win32.TaskScheduler;
using PlotterDataGH.Properties;
using System.Collections.Generic;
using System.Net.Http;
using System.Windows;

namespace WpfApp2
{
    /// <summary>
    /// Interaction logic for Settings.xaml
    /// </summary>
    public partial class SettingsPage : Window
    {
        private static readonly HttpClient client = new HttpClient();
        public SettingsPage()
        {
            InitializeComponent();
        }

        private void btnCancel_Click(object sender, RoutedEventArgs e)
        {
            MainWindow mw = new MainWindow();
            mw.Show();
            this.Close();
        }

        private void btnFactorySettings_Click(object sender, RoutedEventArgs e)
        {
            MessageBoxResult messageBoxResult = System.Windows.MessageBox.Show("Weet je het zeker?", "Fabrieksinstellingen", System.Windows.MessageBoxButton.YesNo);
            if (messageBoxResult == MessageBoxResult.Yes)
            {
                // Get the service on the local machine
                using (TaskService ts = new TaskService())
                {
                    if (ts.GetTask("Plotter Scanner") != null)
                    {
                        ts.RootFolder.DeleteTask("Plotter Scanner");
                    }
                }

                SqliteConnection cnn1;
                cnn1 = new SqliteConnection("Data Source=plotterData.db;");
                cnn1.Open();

                string query = string.Format("Delete from users");
                //create command and assign the query and connection from the constructor
                SqliteCommand cmd1 = new SqliteCommand(query, cnn1);

                //Execute command
                cmd1.ExecuteNonQuery();

                cnn1.Close();

                InitSetup ISP = new InitSetup();
                ISP.Show();
                this.Close();
            }
        }

        private void btnVerder_Click(object sender, RoutedEventArgs e)
        {
            MainWindow mw = new MainWindow();
            mw.Show();
            this.Close();
        }

        private void btnEditSettings_Click(object sender, RoutedEventArgs e)
        {
            Settings.Default.editingAccount = true;
            InitSetup mw = new InitSetup();
            mw.editingMode();
            mw.Show();
            this.Close();
            Settings.Default.editingAccount = false;
            Settings.Default.Save();
        }


        private async System.Threading.Tasks.Task btnSendData_ClickAsync()
        {
            var values = new Dictionary<string, string>
            {
                { "postType", "Plotter" },
                { "serial_number", "CN61U7H02Q" },
                { "model_id", "2" },
                { "meters_printed", "52383.33 m²" },
                { "naam", "SQLITE test" },
                { "IP", "10.0.0.48" }
            };

            var content = new FormUrlEncodedContent(values);

            var response = await client.PostAsync("http://localhost/Interface/", content);

            var responseString = await response.Content.ReadAsStringAsync();
        }

        private void btnSendData_Click(object sender, RoutedEventArgs e)
        {
            btnSendData_ClickAsync();
        }
    }
}
```