

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4 using System.Windows;
5 using System.Windows.Controls;
6 using System.Windows.Data;
7 using System.Windows.Documents;
8 using System.Windows.Input;
9 using System.Windows.Media;
10 using System.Windows.Media.Imaging;
11 using System.Windows.Shapes;
12 using System.Text.Json;
13 using System.Text.Json.Serialization;
14 using System.IO;
15 using Newtonsoft.Json;
16 using MySql.Data.MySqlClient;
17 using PlotterDataGH.Properties;
18 using PlotterDataGH;
19 using Microsoft.Data.Sqlite;
20 using System.Data;
21 using System.Net.Http;
22 using System.Threading.Tasks;
23 using Microsoft.Win32.TaskScheduler;
24 using System.Reflection;
25
26 namespace WpfApp2
27 {
28     /// <summary>
29     /// Interaction logic for InitSetup.xaml
30     /// </summary>
31     public partial class InitSetup : Window
32     {
33         bool editMode = false;
34         bool sendData = true;
35         private static readonly HttpClient client = new HttpClient();
36         public InitSetup()
37         {
38             InitializeComponent();
39
40             //If the account is not being edited create a new instance
41             if (Settings.Default.editingAccount == false)
42             {
43                 DataTable dataTable = new DataTable();
44                 SQLiteConnection cnn;
45                 SQLiteCommand cmd = null;
46                 cnn = new SQLiteConnection("Data Source=plotterData.db;");
47                 cnn.Open();
48
49                 string query = "select bedrijfs_Naam from users";
50                 cmd = new SQLiteCommand(query, cnn);
51
52                 SQLiteDataReader reader = cmd.ExecuteReader();
53                 dataTable.Load(reader);
54
55                 if (dataTable.Rows.Count == 1)
56                 {
```

```
57         MainWindow mw = new MainWindow();
58         Settings.Default.bedrijfsNaam = dataTable.Rows[0]
59         ["bedrijfs_Naam"].ToString();
60         Settings.Default.Save();
61         mw.Show();
62         MessageBox.Show("Hallo " + dataTable.Rows[0]
63         ["bedrijfs_Naam"]);
64         this.Close();
65     }
66     cbxSendData.IsChecked = false;
67 }
68 else
69 {
70     editMode = true;
71 }
72 }
73
74 public void editingMode()
75 {
76     //If the account is being edited
77     DataTable dataTable = new DataTable();
78     SqlConnection cnn;
79     SqlCommand cmd = null;
80     cnn = new SqlConnection("Data Source=plotterData.db;");
81     cnn.Open();
82
83     string query = "select * from users";
84     cmd = new SqlCommand(query, cnn);
85
86     SqlDataReader reader = cmd.ExecuteReader();
87     dataTable.Load(reader);
88
89     tbxBedrijfsnaam.Text = dataTable.Rows[0]["bedrijfs_Naam"].ToString
90     ();
91     tbxContactpersoon.Text = dataTable.Rows[0]
92     ["contactpersoon"].ToString();
93     tbxEmail.Text = dataTable.Rows[0]["email"].ToString();
94     tbxTelefoonnummer.Text = dataTable.Rows[0]
95     ["telefoonnummer"].ToString();
96     cbxSendData.IsChecked = Settings.Default.sendData;
97 }
98
99 private void btnCancel_Click(object sender, RoutedEventArgs e)
100 {
101     this.Close();
102 }
103
104 private void btnNext_Click(object sender, RoutedEventArgs e)
105 {
106     if(editMode == false)
107     {
108         if (tbxBedrijfsnaam.Text != "" && tbxContactpersoon.Text != "" &&
109             && IsValidEmail(tbxEmail.Text) != false &&
```

```
        tbxTelefoonnummer.Text != "")
    {
        Settings.Default.sendData = false;
        if (cbxSendData.IsChecked == true)
        {
            //Send account data to Goedhart Group
            Await();
        }

        //Create a new account
        SqlConnection cnn1;
        cnn1 = new SqlConnection("Data Source=plotterData.db;");
        cnn1.Open();

        string query = string.Format("INSERT INTO users
        (bedrijfs_Naam, contactpersoon, email, telefoonnummer)
        VALUES('{0}', '{1}', '{2}', '{3}')" , tbxBedrijfsnaam.Text,
        tbxContactpersoon.Text, tbxEmail.Text,
        tbxTelefoonnummer.Text);
        //create command and assign the query and connection from
        the constructor
        SqlCommand cmd1 = new SqlCommand(query, cnn1);

        //Execute command
        cmd1.ExecuteNonQuery();

        cnn1.Close();

        Settings.Default.bedrijfsNaam = tbxBedrijfsnaam.Text;

        Settings.Default.Save();

        MainWindow mw = new MainWindow();
        mw.Show();
        this.Close();
    }
    else
    {
        MessageBox.Show("Invalid Info");
    }
}
else
{
    if(IsValidEmail(tbxEmail.Text))
    {
        Settings.Default.sendData = false;
        if (cbxSendData.IsChecked == true)
        {
            //Send account data to Goedhart Group
            Await();
        }

        //Update the current account
        SqlConnection cnn1;
```

```

... \Plotter Data\WpfApp2\WpfApp2\WpfApp2\InitSetup.xaml.cs 4
156     cnn1 = new SqlConnection("Data Source=plotterData.db;");
157     cnn1.Open();
158
159     string query = string.Format("UPDATE users SET
bedrijfs_Naam = '{0}', contactpersoon = '{1}', email =
'{2}', telefoonnummer = '{3}' WHERE id = 1 ",
tbxBedrijfsnaam.Text, tbxContactpersoon.Text,
tbxEmail.Text, tbxTelefoonnummer.Text);
160     //create command and assign the query and connection from
the constructor
SqliteCommand cmd1 = new SqliteCommand(query, cnn1);
161
162
163     //Execute command
164     cmd1.ExecuteNonQuery();
165
166     cnn1.Close();
167
168     Settings.Default.bedrijfsNaam = tbxBedrijfsnaam.Text;
169
170     Settings.Default.Save();
171
172     MainWindow mw = new MainWindow();
173     mw.Show();
174     this.Close();
175 }
176 else
177 {
178     MessageBox.Show("False Email");
179 }
180 }
181
182 }
183
184 async System.Threading.Tasks.Task Await()
185 {
186     //Send data to Goedhart Group
187     int timeout = 5000;
188     var task = SendUserAsync(tbxBedrijfsnaam.Text,
tbxContactpersoon.Text, tbxEmail.Text, tbxTelefoonnummer.Text);
189     if (await System.Threading.Tasks.Task.WhenAny(task,
System.Threading.Tasks.Task.Delay(timeout)) == task)
190     {
191         // task completed within timeout
192     }
193     else
194     {
195         MessageBox.Show("Verbinding met de Goedhart Servers kon niet
gemaakt worden");
196     }
197 }
198
199 async System.Threading.Tasks.Task SendUserAsync(string bedrijfsnaam,
string contactpersoon, string email, string telefoonnummer)
200 {
201     //Send data to Goedhart Group

```

```
202     var values = new Dictionary<string, string>
203     {
204         { "postType", "Users" },
205         { "bedrijfs_Naam", bedrijfsnaam },
206         { "contactpersoon", contactpersoon },
207         { "email", email },
208         { "telefoonnummer", telefoonnummer }
209     };
210
211     var content = new FormUrlEncodedContent(values);
212
213     var response = await client.PostAsync("http://10.0.0.125/",      ↗
        content);
214 }
215
216 bool IsValidEmail(string email)
217 {
218     //Check if the given email is valid
219     try
220     {
221         var addr = new System.Net.Mail.MailAddress(email);
222         return addr.Address == email;
223     }
224     catch
225     {
226         return false;
227     }
228 }
229
230 private void cbxSendData_Checked(object sender, RoutedEventArgs e)
231 {
232     //MessageBox for getting authorization to send data to Goedhart  ↗
        groep
233     if (cbxSendData.IsChecked == false)
234     {
235         sendData = false;
236     }
237     else
238     {
239         MessageBoxResult messageBoxResult =                      ↗
            System.Windows.MessageBox.Show("Goedhart Groep zal alleen de ↗
                opgegeven profieldata en de data van de printers ↗
                verkrijgen", "Fabrieksinstellingen",                ↗
                System.Windows.MessageBoxButton.YesNo);
240         if (messageBoxResult == MessageBoxResult.Yes)
241         {
242             sendData = true;
243         }
244         else
245         {
246             cbxSendData.IsChecked = false;
247         }
248     }
249 }
250 }
251 }
```