

System Monitor Program

Finley Shaun Michael
Clarke (1910794)

Applied Computing

CPU	
NAME:	AMD Ryzen 7 3700X
ARCHITECTURE:	AmdZen
SOCKET TYPE:	AM4
CORE COUNT:	8
THREAD COUNT:	16
BASE CLOCK:	3600 MHZ
BOOST CLOCK:	4400 MHZ
LOAD:	1.79 %
MINMUM LOAD:	0.62 %
MAXIMUM LOAD:	10.27 %
CURRENT CLOCK:	3600 MHZ
TEMPERATURE:	31.84 °C

Memory	
LOAD:	6.42 GB
RESERVE:	9.49 GB
Modules	
BANK 0	
NAME:	Kingston KHX2666C16/8G
CLOCK:	2666 MHZ
CAPACITY:	8192 MB
BANK 2	
NAME:	Kingston KHX2666C16/8G
CLOCK:	2666 MHZ
CAPACITY:	8192 MB

GPU	
NAME:	NVIDIA GeForce GTX 1080
LOAD:	0 %
MEMORY (VRAM):	8192 MB
CURRENT CLOCK:	1189.5 MHZ
TEMPERATURE:	54 °C

```
1 using System.Windows;
2 using LibreHardwareMonitor.Hardware;
3 using System.Collections.Generic;
4 using Models;
5 using LibreHardwareMonitor.Hardware.Motherboard;
6 using System;
7 using System.Windows.Threading;
8 namespace Hardware_Monitor
9 {
10     2 references
11     public partial class MainWindow : Window
12     {
13         public bool isFirstRun = true;
14         //The following declare the hardware models that will be populated below
15         public PC pc = new PC()
16         {
17             CPUs = new List<CPU>(),
18             GPUs = new List<GPU>(),
19             Memory = new Memory(),
20             Storage = new List<Storage>(),
21             mainboardName = ""
22         };
23         CPU cpu = new CPU();
24         private Computer c = new Computer()
25         {
26             IsGpuEnabled = true, //Reports GPU
27             IsCpuEnabled = true, //Reports CPU
28             IsMemoryEnabled = true, // Reports System Memory (RAM)
29             IsMotherboardEnabled = true, //Reports Motherboard
30             IsStorageEnabled = true, //Reports HDD
31         };
32         1 reference
33         private void ReportSystemInfo(object sender, EventArgs e)
34         {
35             // The follow lines call the hardware data used in the program
36             foreach (IHardware hardware in c.Hardware)
37             {
38                 if (hardware.HardwareType == HardwareType.Cpu) //Calls CPU information
39                 {
40                     hardware.Update();
41                     if (isFirstRun) cpu.Name = hardware.Name; //Only calls and populates if first run
42                     foreach (ISensor sensor in hardware.Sensors) // Calls CPU name
43                     {
44                         if (sensor.SensorType == SensorType.Temperature) //Sensor that handles CPU temperature information
45                         {
46                             cpu.Temperature = (float)Math.Round((float)sensor.Value, 2);
47                         }
48                         if (sensor.SensorType == SensorType.Load && sensor.Name == "CPU Total") //Sensor that handles CPU values/information
49                         {
50                             cpu.Load = (float)Math.Round((float)sensor.Value, 2); //Lists current CPU load (value is rounded to second decimal place)
51                             cpu.Min = (float)Math.Round((float)sensor.Min, 2); //Lists CPU minimum clock (value is rounded to second decimal place)
52                             cpu.Max = (float)Math.Round((float)sensor.Max, 2); //Lists CPU maximum clock (value is rounded to second decimal place)
53                         }
54                         if (sensor.SensorType == SensorType.Clock && sensor.Name == "Core #1") //Sensor that handles CPU clockspeed
55                         {
56                             cpu.Clockspeed = (float)Math.Round((float)sensor.Value, 2);
57                         }
58                     }
59                     if (isFirstRun) pc.CPUs.Add(cpu);
60                 }
61             }
62             int i = 0;
63             if (hardware.HardwareType == HardwareType.GpuAmd || hardware.HardwareType == HardwareType.GpuNvidia || hardware.HardwareType == HardwareType
```

Program Interface

The program in development in Visual Studio
(The program was written in C#)