```
/*--------------------------------------------------*\
 * Author    : Salvi Cyril
 * Date      : 7th juny 2017
 * Diploma   : RaspiHome
 * Classroom : T.IS-E2B
 *
 * Description:
 *      RaspiHomeSenseHAT is a program who use a
 *   Sense HAT, it's an electronic card who can be
 *   mesured value with sensor. This program use
 *   the Sense HAT to mesure the temperature, the
 *   humidity and the pressure.
\*--------------------------------------------------*/

using System;
using Emmellsoft.IoT.Rpi.SenseHat;
using Windows.UI;
using Windows.UI.Xaml;

namespace RaspiHomeSenseHAT
{
    public class ModelSenseHAT
    {
        #region Fields
        #region Constants
        #endregion

        #region Variables
        private ViewSenseHAT _vSenseHAT;
        private CommunicationWithServer _comWithServer;

        // Sense HAT librairy
        private ISenseHat _senseHat;
        private ISenseHatDisplay _senseHatDisplay;
        private SenseHatData _data;

        // Set default color matrix to OFF
        private Color _uiColor = Color.FromArgb(0, 0, 0, 0);
        #endregion
        #endregion

        #region Properties
        public ViewSenseHAT VSenseHAT
        {
            get
            {
                return _vSenseHAT;
            }

            set
            {
                _vSenseHAT = value;
            }
        }

        public SenseHatData Data
        {
            get
            {
                return _data;
            }

            set
            {
                _data = value;
            }
        }

        public CommunicationWithServer ComWithServer
        {
            get
            {
                return _comWithServer;
            }

            set
            {
                _comWithServer = value;
            }
        }
        #endregion

        #region Constructors
        /// <summary>
        /// Constructor: Initializer
        /// </summary>
        /// <param name="paramView"></param>
        public ModelSenseHAT(ViewSenseHAT paramView)
        {
            // Communication like Model-View
            this.VSenseHAT = paramView;

            // Initilize the Sense HAT (don't need to be initialized before the communication start because it's only a sensor)
            InitializeSenseHat();

            // Initilize the communication with the server
            this.ComWithServer = new CommunicationWithServer(this);

        }
        #endregion

        #region Methods
        /// <summary>
        /// Initialize the Sense HAT
        /// </summary>
        public async void InitializeSenseHat()
        {
            this._senseHat = await SenseHatFactory.GetSenseHat();
            this._senseHatDisplay = this._senseHat.Display;
            this._senseHatDisplay.Fill(_uiColor);

            SetValue();
        }

        /// <summary>
        /// Set the value get with sensor
        /// </summary>
        public void SetValue()
        {
            // Update values
            this._senseHat.Sensors.HumiditySensor.Update();
            this._senseHat.Sensors.PressureSensor.Update();
            this._senseHatDisplay.Update();

            // Set values
            this.Data = new SenseHatData();
            this.Data.Temperature = this._senseHat.Sensors.Temperature;
            this.Data.Humidity = this._senseHat.Sensors.Humidity;
            this.Data.Pressure = this._senseHat.Sensors.Pressure;
        }

        /// <summary>
        /// Send the values with a special format:"TEMP=x;HUMI=y;PRES=z"
        /// /// Values are rounded
        /// </summary>
        /// <returns></returns>
        public string SendValues()
        {
            // Update values of sensors
            SetValue();

            return "TEMP=" + Math.Round((decimal)this.Data.Temperature) + ";"
              + "HUMI=" + Math.Round((decimal)this.Data.Humidity) + ";" +
              "PRES=" + Math.Round((decimal)this.Data.Pressure);
        }
        #endregion
    }
}
```