```
/*
 * Author  : Dubas Loïc
 * Class   : I.FA-P3B
 * School  : CFPT-I
 * Date    : June 2018
 * Descr.  : Create a new modele, with a name, a description and a picture
 * Version : 1.0
 * Ext. dll: LeapCSharp.NET4.5
 */

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
// References to add
using Leap;
using System.Diagnostics;
using System.Xml;
using System.Xml.Serialization;
using System.IO;

namespace fingers_cloner
{
    public partial class frmNewModele : Form
    {
        #region Initialization
        // initialize Leap Motion
        LeapController leapController;

        // initialize Paint functions
        Paint paint;

        // current position
        MyHand currentPosition;

        // initialize serialization functions
        Serialization serialization;

        // name, description and picture of the modele
        string name;
        string description;
        Bitmap loadedPicture;
        string imageAsString;

        List<MyHand> allPositions;
        #endregion

        /// <summary>
        /// create new modele form
        /// </summary>
        /// <param name="fingersNormPos">finger's normalized position</param>
        /// <param name="palmNormPos">palm's normalized position</param>
        public frmNewModele(MyHand handToSave)
        {
            InitializeComponent();
            DoubleBuffered = true;

            leapController = new LeapController();
            paint = new Paint();
            paint.GetPanelSize(pnlModele.Width, pnlModele.Height);
            serialization = new Serialization();

            this.currentPosition = handToSave;
        }

        /// <summary>
        ///  draw hand if there is one
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void pnlModele_Paint(object sender, PaintEventArgs e)
        {
            try
            {
                paint.paintHand(e, currentPosition);
            }
            catch (Exception)
            {
                NoHandDetected();
            }
        }

        /// <summary>
        /// enable save button if there is a name to it and if the name is not already taken
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void tbxModeleName_TextChanged(object sender, EventArgs e)
        {
            if (tbxModeleName.Text.Length <= 0)
            {
                btnSave.Enabled = false;
            }
            else if (checkName())
            {
                btnSave.Enabled = false;
            }
            else
            {
                btnSave.Enabled = true;
            }
        }

        /// <summary>
        /// open file dialog to choose image
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void btnLoadImage_Click(object sender, EventArgs e)
        {
            OpenFileDialog ofd = new OpenFileDialog();

            ofd.InitialDirectory = "C:\\Users";
            ofd.Filter = "Image files (*.png, *.jpg, *.jpeg, *.gif, *.bmp)|*.png;*.jpg;*.jpeg;*.gif;*.bmp";

            if (ofd.ShowDialog() == DialogResult.OK)
            {
                loadedPicture = new Bitmap(ofd.FileName);
                lblFileName.Text = ofd.SafeFileName;
                lblFileName.Visible = true;
                TypeConverter converter = TypeDescriptor.GetConverter(typeof(Bitmap));
                imageAsString = Convert.ToBase64String((Byte[])converter.ConvertTo(loadedPicture, typeof(Byte[])));
            }
        }

        /// <summary>
        /// modify position to save and serialize it
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void btnSave_Click(object sender, EventArgs e)
        {
            name = tbxModeleName.Text;

            // Open a new form to add a description
            frmComment comment = new frmComment();
            comment.ShowDialog();

            // when click on 'OK' on the comment form
            if (comment.DialogResult == DialogResult.OK)
            {
                // add description and name to position to save
                description = comment.Description;
                currentPosition.Description = description;
                currentPosition.Name = name;
                if (loadedPicture != null)
                {
                    currentPosition.Image = imageAsString;
                }

                // serialize the savedHand object
                serialization.serialize(currentPosition);

                // Close comment and newModele form
                this.Close();
            }
        }

        /// <summary>
        /// if there is no hand detected by the Leap, user is informed and send back to main form
        /// </summary>
        private void NoHandDetected()
        {
            MessageBox.Show("Aucune main détectée. Veuillez réessayer.");
            this.Close();
        }

        /// <summary>
        /// get all saved positions
        /// </summary>
        /// <param name="allPositions">saved positions</param>
        public void getAllPositions(List<MyHand> allPositions)
        {
            this.allPositions = allPositions;
        }

        /// <summary>
        /// check if the name is already taken
        /// </summary>
        /// <returns></returns>
        private bool checkName()
        {
            bool nameTaken = false;

            if (allPositions != null)
            {
                for (int i = 0; i < allPositions.Count; i++)
                {
                    if (allPositions[i].Name == tbxModeleName.Text)
                    {
                        nameTaken = true;
                        break;
                    }
                }
            }

            return nameTaken;
        }
    }
}
```