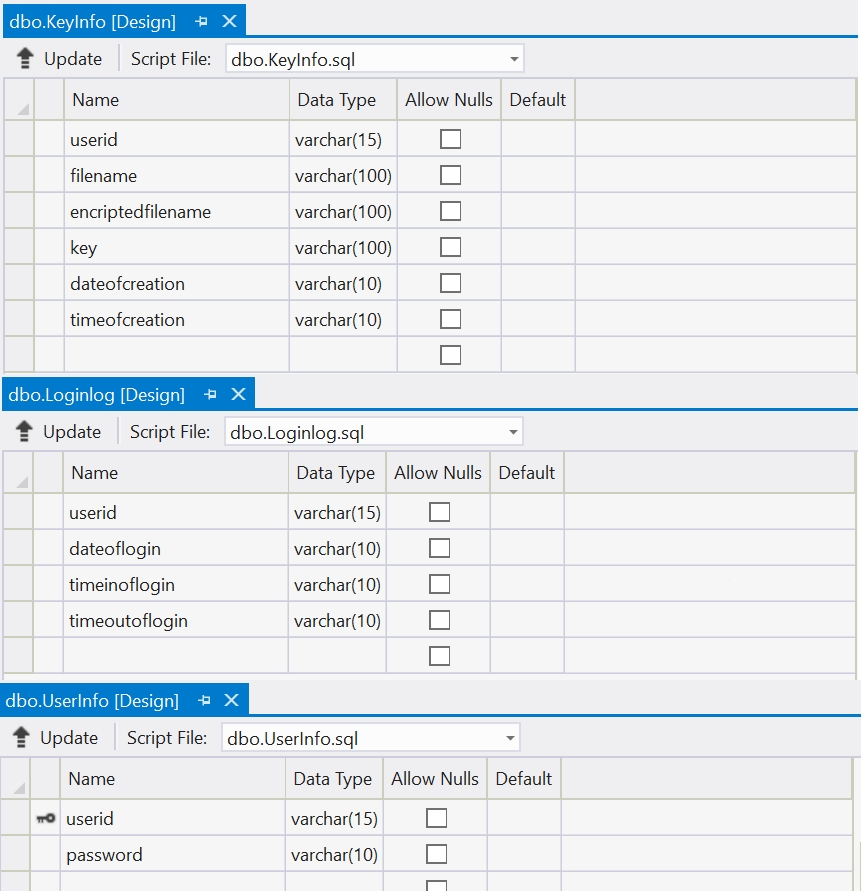
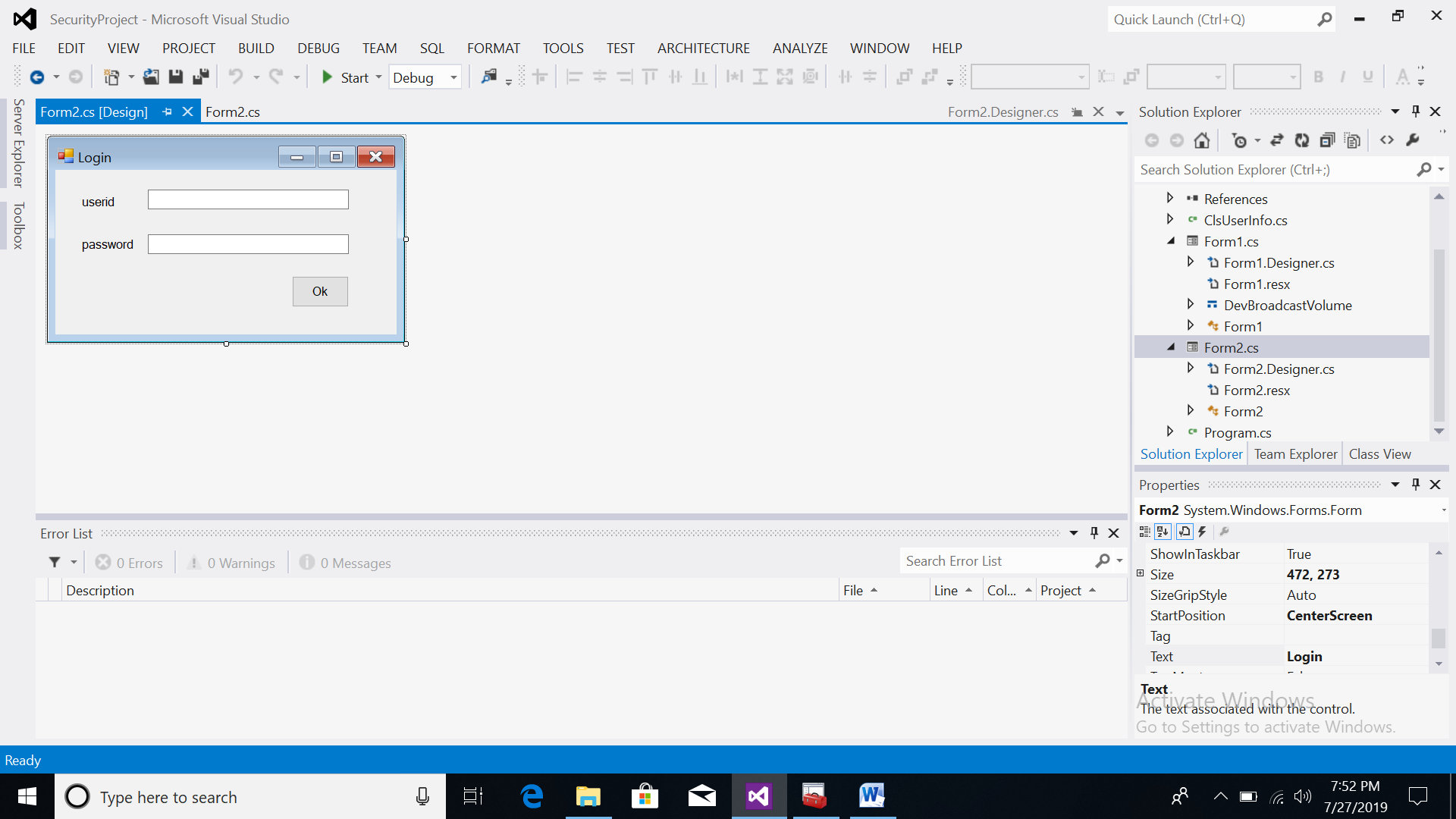
**SECURITY PROJECT**

**DATABASE TABLES**



**FORM2.DESIGN.CS**



**FORM2.CS Program**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace SecurityProject

{

public partial class Form2 : Form

{

string getDate(DateTime dt)

{

string CurrentDate = dt.Day.ToString() + "/" + dt.Month.ToString() + "/" + dt.Year.ToString();

return CurrentDate;

}

string getTime(DateTime t)

{

string CurrentTime = t.ToShortTimeString();

return CurrentTime;

}

public Form2()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void Form2\_Load(object sender, EventArgs e)

{

}

private void btnOk\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection("Data Source=desktop-kr9upqb;Initial Catalog=dataprotector;Integrated Security=True;Pooling=False");

con.Open();

string query = "select \* from userinfo where userid='" + txtId.Text + "' and password='" + txtPassword.Text + "'";

SqlCommand cmd = new SqlCommand(query, con);

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{}

if (dr.HasRows)

{

dr.Close();

ClsUserInfo.userid = txtId.Text;

string dateofLogin = getDate(DateTime.Now);

string timeinofLogin = getTime(DateTime.Now);

ClsUserInfo.dateoflogin = dateofLogin;

ClsUserInfo.timeinoflogin = timeinofLogin;

query = "insert into Loginlog values('" + txtId.Text + "','" + dateofLogin + "','" + timeinofLogin + "','\*\*\*')";

cmd = new SqlCommand(query, con);

cmd.ExecuteNonQuery();

Form1 frm = new Form1();

frm.Show();

con.Close();

this.Visible = false;

}

else

{

MessageBox.Show("Wrong UserId/Password ");

txtId.Text = "";

txtPassword.Text = "";

txtId.Focus();

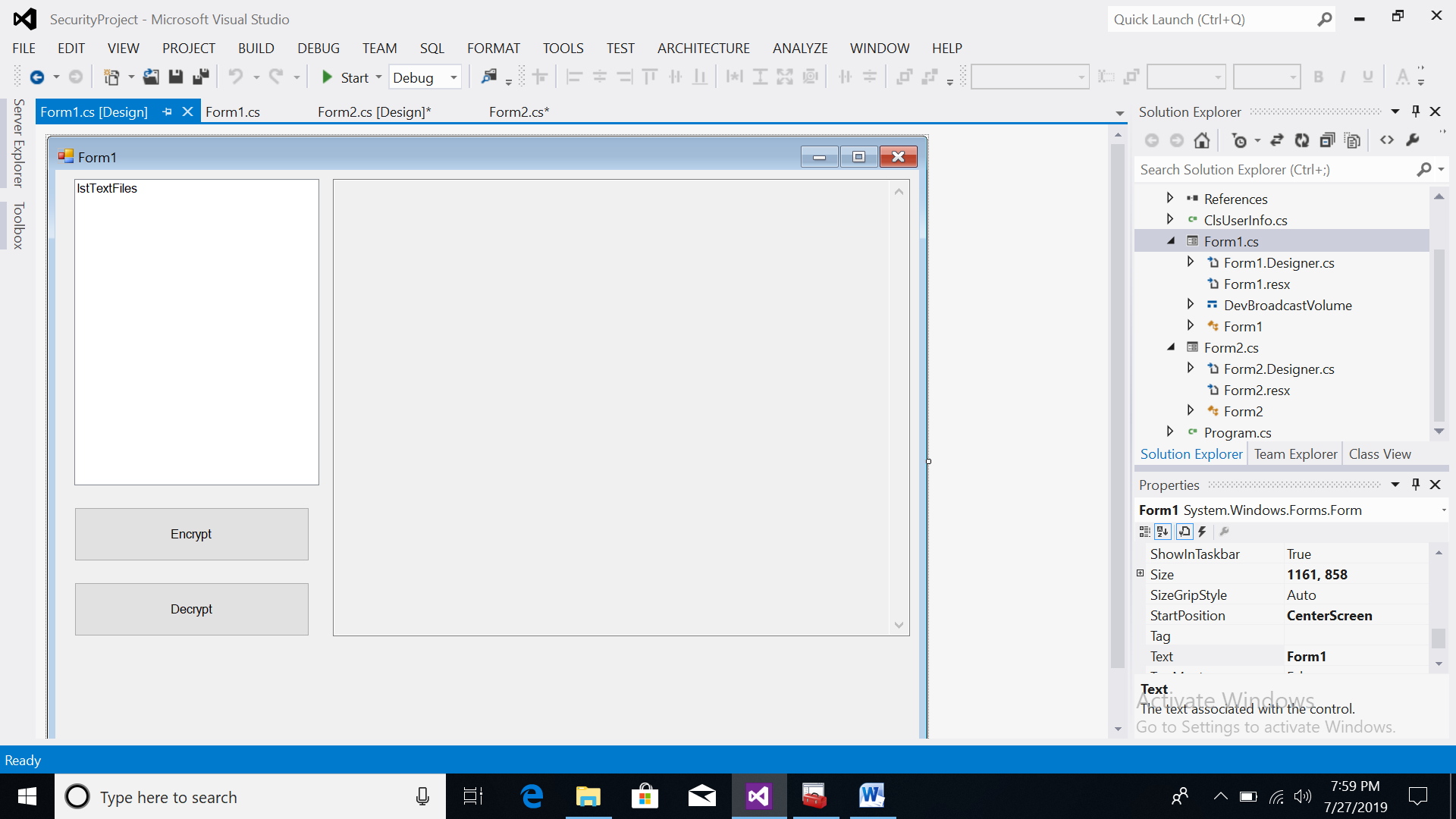
}

}

}

}

**FORM1.DESIGN.CS**



**FORM1.CS Program**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.IO;

using System.Security;

using System.Security.Cryptography;

using System.Runtime.InteropServices;

namespace SecurityProject

{

[StructLayout(LayoutKind.Sequential)]

public struct DevBroadcastVolume

{

public int Size;

public int DeviceType;

public int Reserved;

public int Mask;

public Int16 Flags;

}

public partial class Form1 : Form

{

private const int WM\_DEVICECHANGE = 0x219;

private const int DBT\_DEVICEARRIVAL = 0x8000;

private const int DBT\_DEVICEREMOVECOMPLETE = 0x8004;

private const int DBT\_DEVTYP\_VOLUME = 0x00000002;

string FileName = "";

string path = "";

protected override void WndProc(ref Message m)

{

base.WndProc(ref m);

switch (m.Msg)

{

case WM\_DEVICECHANGE:

switch ((int)m.WParam)

{

case DBT\_DEVICEARRIVAL:

lstTextFiles.Items.Add("New Device Arrived");

int devType = Marshal.ReadInt32(m.LParam, 4);

DriveInfo[] allDrives = DriveInfo.GetDrives();

foreach (DriveInfo d in allDrives)

{

if (d.DriveType == DriveType.Removable)

{

if (d.VolumeLabel == "SSONI PEN 1")

{

path = d.Name;

FileInfo[] AllTextFiles = GetAllTextFiles(d.Name);

for (int i = 0; i < AllTextFiles.Length; i++)

{

if (AllTextFiles[i].FullName.EndsWith(".txt"))

{

lstTextFiles.Items.Add(AllTextFiles[i].FullName);

}

}

}

}

}

break;

case DBT\_DEVICEREMOVECOMPLETE:

lstTextFiles.Items.Clear();

break;

}

break;

}

}

string getDate(DateTime dt)

{

string CurrentDate = dt.Day.ToString() + "/" + dt.Month.ToString() + "/" + dt.Year.ToString();

return CurrentDate;

}

string getTime(DateTime t)

{

string CurrentTime = t.ToShortTimeString();

return CurrentTime;

}

public Form1()

{

InitializeComponent();

}

FileInfo[] GetAllTextFiles(string path)

{

DirectoryInfo dir = new DirectoryInfo(path);

return dir.GetFiles();

}

[System.Runtime.InteropServices.DllImport("KERNEL32.DLL", EntryPoint = "RtlZeroMemory")]

public static extern bool ZeroMemory(IntPtr Destination, int Length);

// Function to Generate a 64 bits Key.

static string GenerateKey()

{

// Create an instance of Symetric Algorithm. Key and IV is generated automatically.

DESCryptoServiceProvider desCrypto = (DESCryptoServiceProvider)DESCryptoServiceProvider.Create();

// Use the Automatically generated key for Encryption.

return ASCIIEncoding.ASCII.GetString(desCrypto.Key);

}

static void EncryptFile(string sInputFilename, string sOutputFilename, string sKey)

{

try

{

FileStream fsInput = new FileStream(sInputFilename, FileMode.Open, FileAccess.Read);

FileStream fsEncrypted = new FileStream(sOutputFilename, FileMode.Create, FileAccess.Write);

DESCryptoServiceProvider DES = new DESCryptoServiceProvider();

DES.Key = ASCIIEncoding.ASCII.GetBytes(sKey);

DES.IV = ASCIIEncoding.ASCII.GetBytes(sKey);

ICryptoTransform desencrypt = DES.CreateEncryptor();

CryptoStream cryptostream = new CryptoStream(fsEncrypted, desencrypt, CryptoStreamMode.Write);

byte[] bytearrayinput = new byte[fsInput.Length];

fsInput.Read(bytearrayinput, 0, bytearrayinput.Length);

cryptostream.Write(bytearrayinput, 0, bytearrayinput.Length);

cryptostream.Close();

fsInput.Close();

fsEncrypted.Close();

}

catch (Exception e) { MessageBox.Show(e.Message); }

}

public void DecryptFile(string sInputFilename,string sKey)

{

DESCryptoServiceProvider DES = new DESCryptoServiceProvider();

DES.Key = ASCIIEncoding.ASCII.GetBytes(sKey);

DES.IV = ASCIIEncoding.ASCII.GetBytes(sKey);

FileStream fsread = new FileStream(sInputFilename, FileMode.Open, FileAccess.Read);

ICryptoTransform desdecrypt = DES.CreateDecryptor();

CryptoStream cryptostreamDecr = new CryptoStream(fsread, desdecrypt, CryptoStreamMode.Read);

txtDecryptedText.Text = (new StreamReader(cryptostreamDecr).ReadToEnd());

}

void SaveKey(string KeyFileName, string Key)

{

FileStream f = new FileStream(KeyFileName, FileMode.Create);

for (int i = 0; i < Key.Length; i++) f.WriteByte((byte)Key[i]);

}

private void btnSave\_Click(object sender, EventArgs e)

{

if (FileName.Length > 0)

{

SqlConnection con = new SqlConnection("Data Source=desktop-kr9upqb;Initial Catalog=dataprotector;Integrated Security=True;Pooling=False");

con.Open();

string userid = ClsUserInfo.userid;

FileInfo fn = new FileInfo(FileName);

string fName = fn.Name;

Random rnd = new Random();

string EncryptedFileName = fName.Substring(0,fName.IndexOf('.')) + "\_encript.txt";

string KeyFileName = fName.Substring(0, fName.IndexOf('.')) + "\_Key.txt";

string key = GenerateKey();

SaveKey(path+KeyFileName, key);

FileInfo finfo = new FileInfo(KeyFileName);

string dateofcreation = getDate(DateTime.Now);

string timeofcreation = getTime(DateTime.Now);

MessageBox.Show(FileName + "," + path + EncryptedFileName + "," + path + KeyFileName);

EncryptFile(FileName, path+EncryptedFileName, key);

fn = new FileInfo(EncryptedFileName);

EncryptedFileName = fn.Name;

fn = new FileInfo(KeyFileName);

KeyFileName = fn.Name;

string query = "insert into KeyInfo values('" + userid + "','" + fName + "','" + EncryptedFileName + "','" + KeyFileName + "','" + dateofcreation + "','" + timeofcreation + "')";

SqlCommand cmd = new SqlCommand(query, con);

cmd.ExecuteNonQuery();

con.Close();

}

else

{

MessageBox.Show("Please Select the file first ");

}

}

private void lstTextFiles\_Click(object sender, EventArgs e)

{

FileName = lstTextFiles.Items[lstTextFiles.SelectedIndex].ToString();

}

private void btnDecrypt\_Click(object sender, EventArgs e)

{

if (FileName.Length > 0)

{

SqlConnection con = new SqlConnection("Data Source=desktop-kr9upqb;Initial Catalog=dataprotector;Integrated Security=True;Pooling=False");

con.Open();

string fname = FileName;

FileInfo fn = new FileInfo(fname);

fname = fn.Name;

string query = "select \* from keyinfo where userid='" + ClsUserInfo.userid + "' and encriptedfilename='" + fname + "'";

SqlCommand cmd = new SqlCommand(query, con);

MessageBox.Show(query);

SqlDataReader dr = cmd.ExecuteReader();

string key="";

if (dr.Read())

{ }

if (dr.HasRows && path.Length>0)

{

key=dr[3].ToString();

dr.Close();

StreamReader sm = new StreamReader(path+key);

String KeyText = sm.ReadToEnd();

MessageBox.Show((path+fname+":")+(path + key + ":") + KeyText);

DecryptFile(path+fname, KeyText);

MessageBox.Show(KeyText);

}

con.Close();

}

}

private void Form1\_FormClosed(object sender, FormClosedEventArgs e)

{

SqlConnection con = new SqlConnection("Data Source=desktop-kr9upqb;Initial Catalog=dataprotector;Integrated Security=True;Pooling=False");

con.Open();

string query = "update loginlog set timeoutoflogin='" + DateTime.Now.ToShortTimeString() + "' where userid='" + ClsUserInfo.userid + "' and dateoflogin='" + ClsUserInfo.dateoflogin + "' and timeinoflogin='"+ClsUserInfo.timeinoflogin + "'";

SqlCommand cmd = new SqlCommand(query, con);

cmd.ExecuteNonQuery();

}

}

}

**ClsUserInfo**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace SecurityProject

{

class ClsUserInfo

{

public static string userid;

public static string dateoflogin;

public static string timeinoflogin;

}

}

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace SecurityProject

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

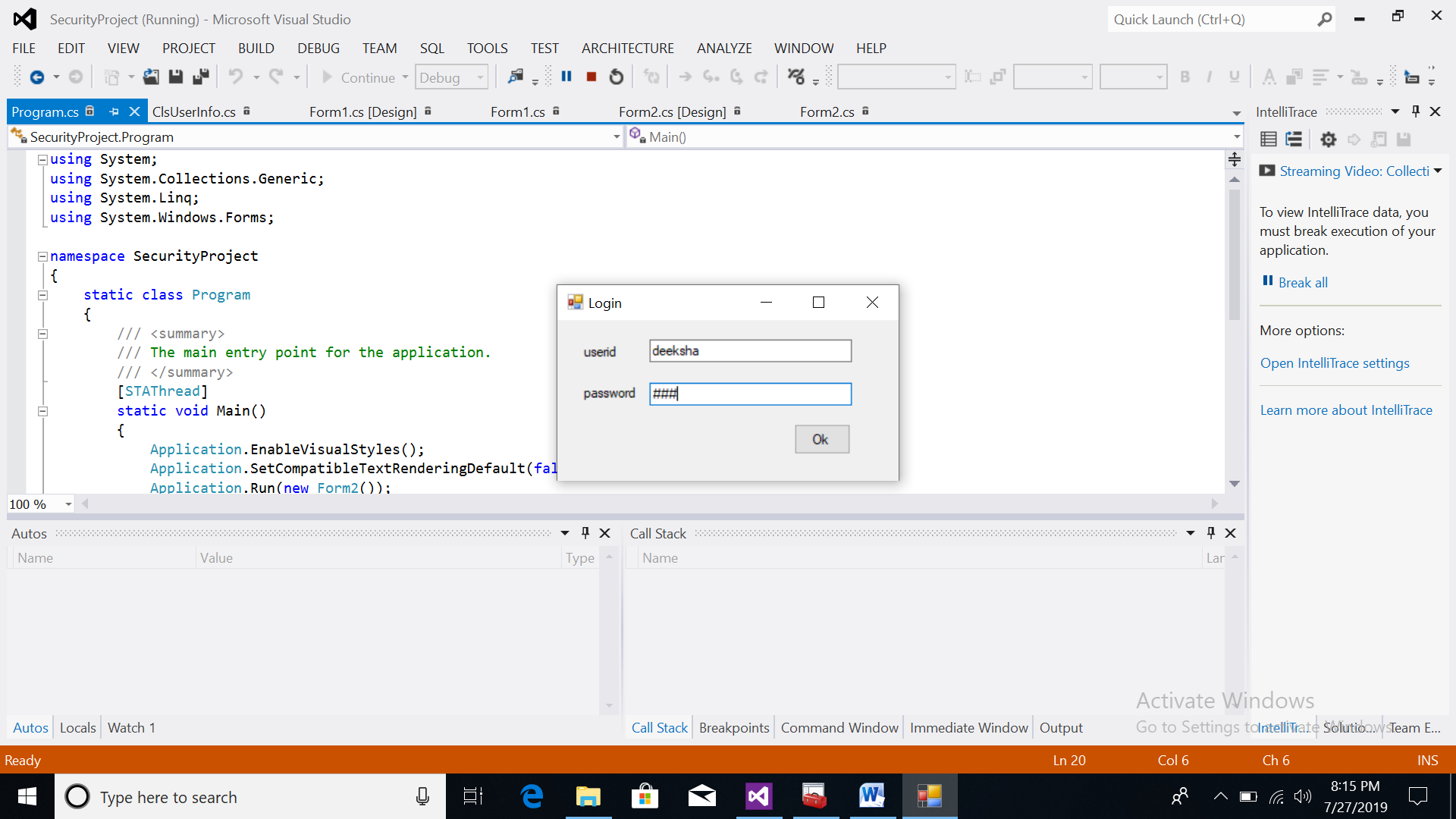
Application.Run(new Form2());

}

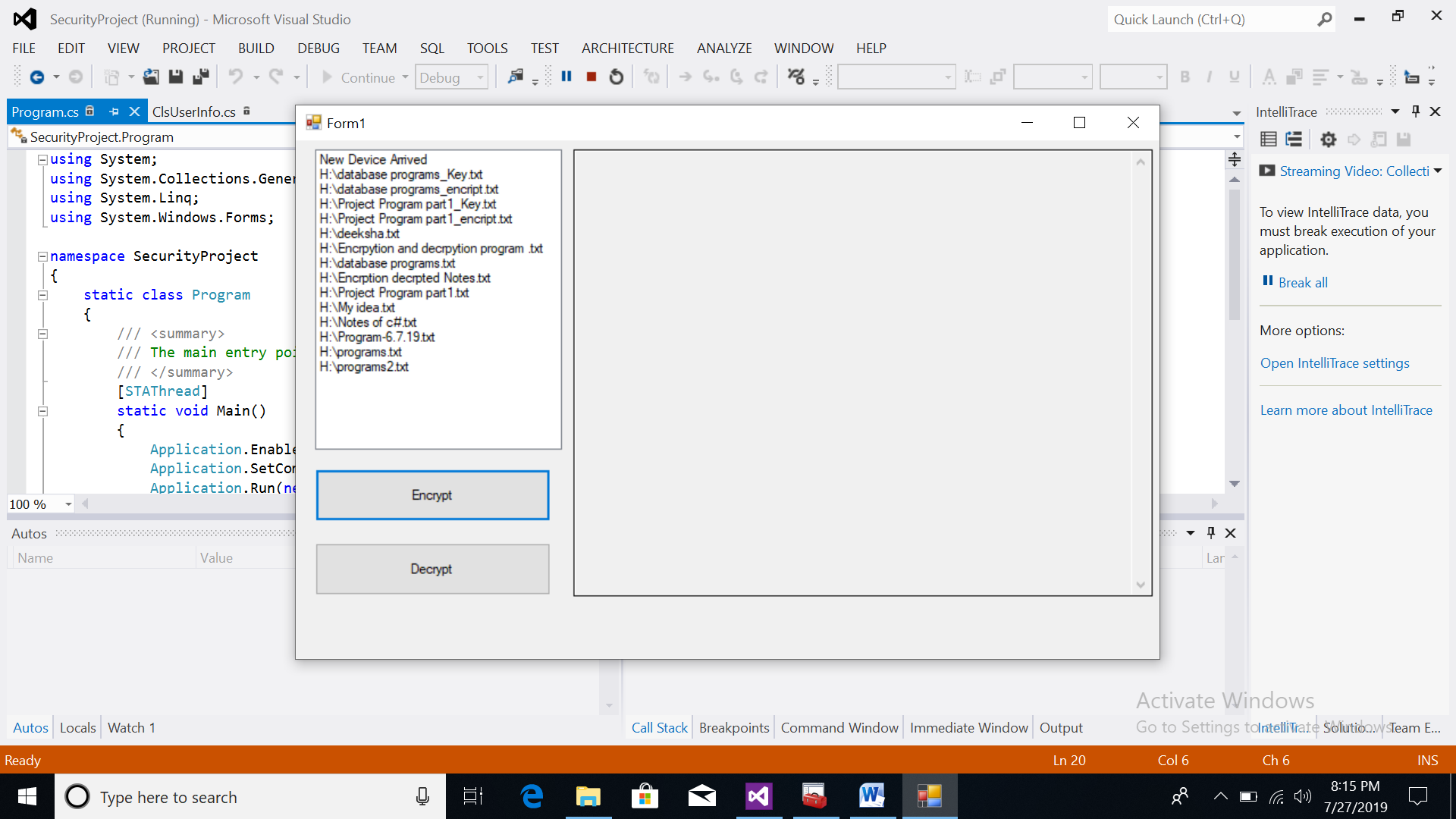
}

}

**RUN THE PROJECT**



**PEN DRIVE INSERT**



**DECRYPED FILE**

