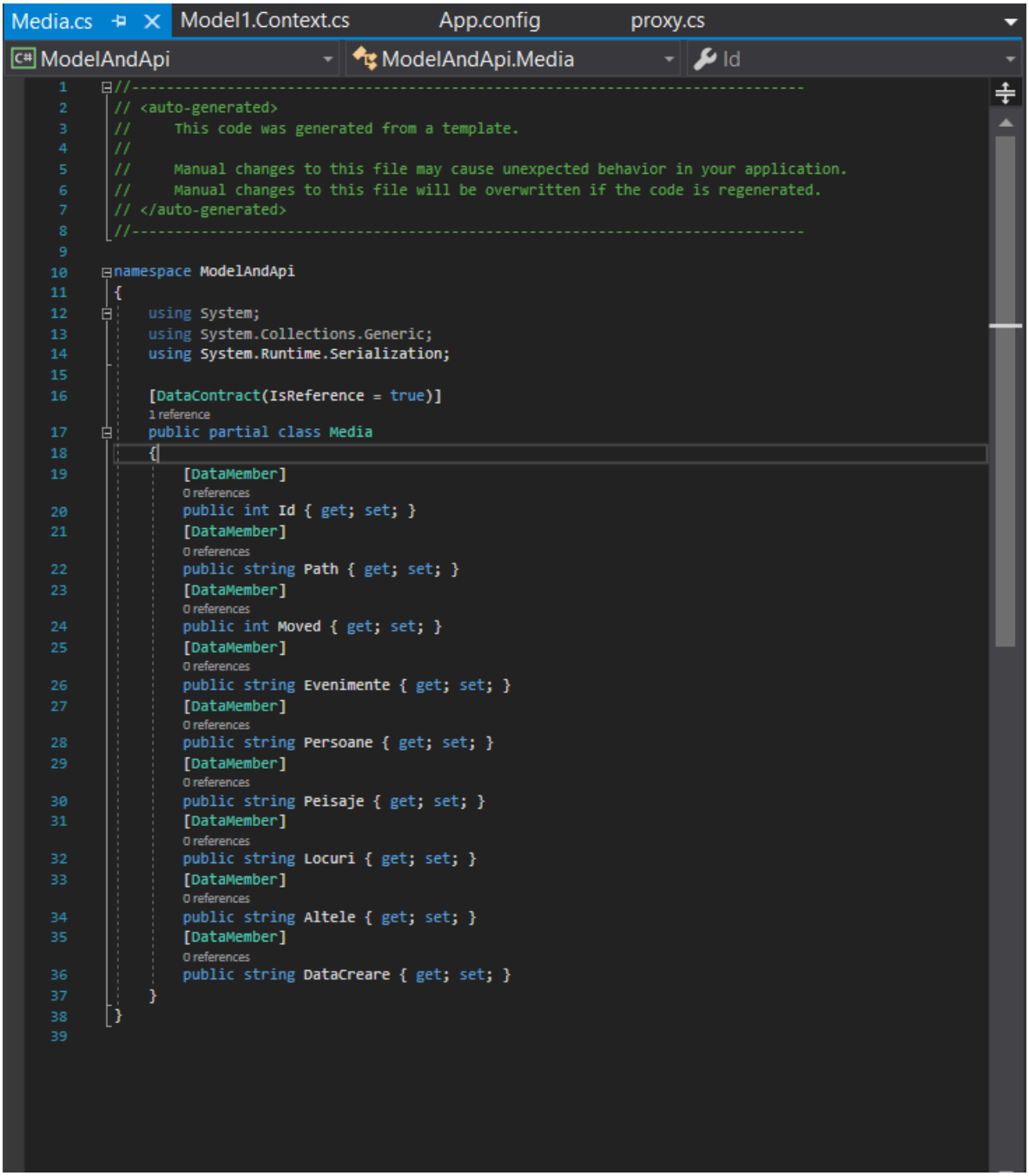
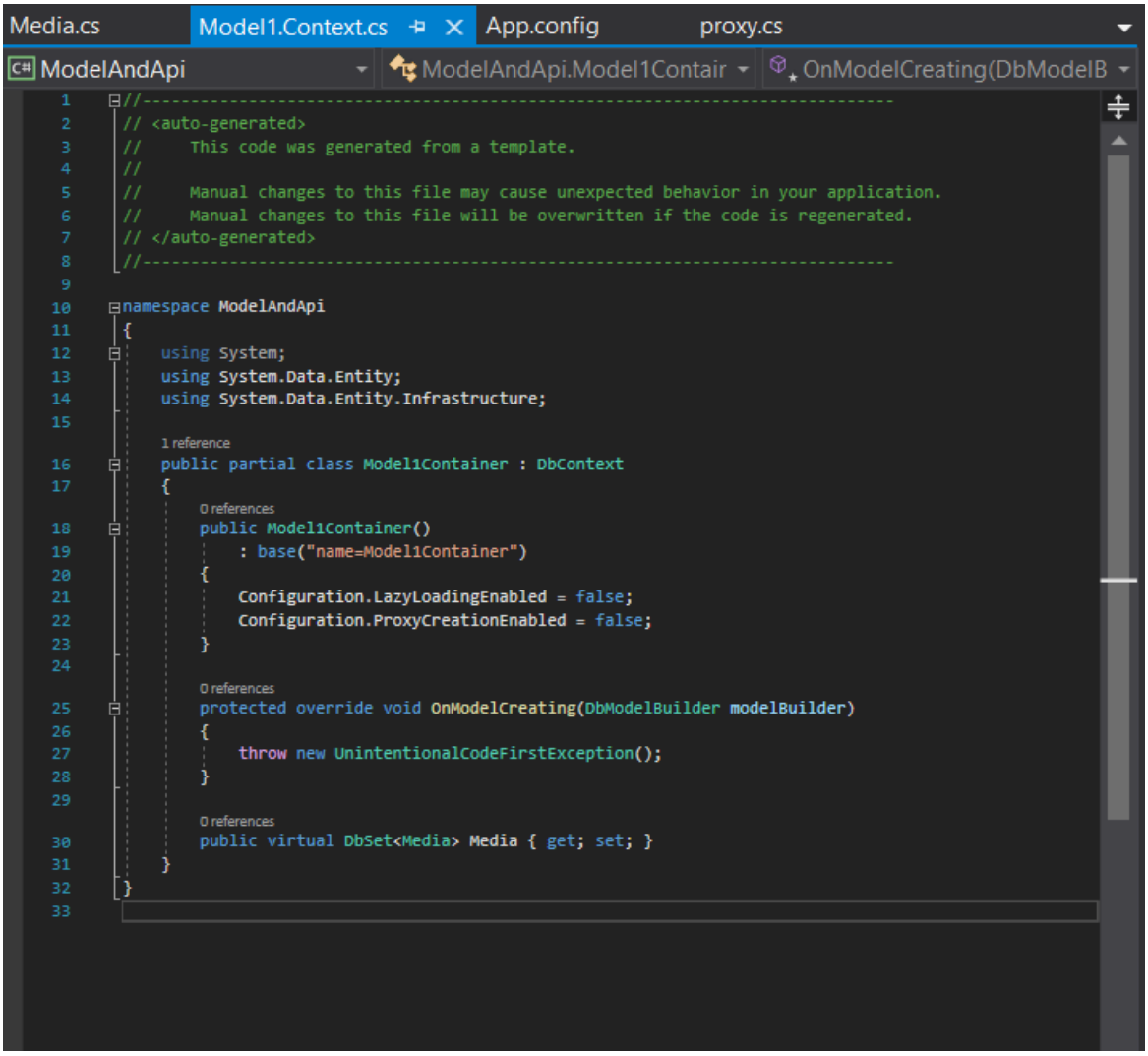
Subsemnata Andra – Elena N. Paduraru declara pe propria raspundere ca acest cod nu a fost copiat din Internet sau din alte surse. Pentru documentare am folosit urmatoarele surse:

Link-uri:
https://profs.info.uaic.ro/~iasimin/Laborator%20C%20S%20H/Laborator%20WCF%202020.pdf
https://profs.info.uaic.ro/~iasimin/Laborator%20C%20S%20H/Laborator1-2016.pdf
https://stackoverflow.com/questions/8986975/c-sharp-return-different-types
https://www.c-sharpcorner.com/UploadFile/mahesh/create-a-text-file-in-C-Sharp/
https://stackoverflow.com/questions/34851800/export-data-from-sql-server-to-text-file-in-c-sharpsaving-to-a-specific-folder
https://www.aspsnippets.com/Articles/Export-data-from-SQL-Server-to-Text-file-in-C-andVBNet.aspx
https://stackoverflow.com/questions/6062192/there-is-already-an-open-datareader-associated-with-thiscommand-which-must-be-c
https://stackoverflow.com/questions/18475195/error-there-is-already-an-open-datareader-associatedwith-this-command-which-mu/18475525
https://docs.microsoft.com/enus/dotnet/api/system.data.sqlclient.sqlparametercollection.addwithvalue?view=netframework-4.8
https://stackoverflow.com/questions/1132422/open-a-folder-using-process-start
https://docs.microsoft.com/en-us/dotnet/framework/winforms/controls/how-to-bind-data-to-thewindows-forms-datagridview-control
https://stackoverflow.com/questions/27874566/getting-values-from-sql-reader-c-sharp
https://forums.asp.net/t/1794939.aspx?How+to+read+output+of+sql+command+C+

Modificari fata de proiectul p1:

Model:
S-a modificat clasa generata din model (Model1.Context.cs) si s-au adaugat in constructor Configuration.LazyLoadingEnabled = false; si Configuration.ProxyCreationEnabled = false;. La fel s-a procedat si pentru clasa Media.cs din model unde s-au adaugat DataContract si DataMember.




Media.cs
Model1.Context.cs
App.config
proxy.cs
ModelAndApi
ModelAndApi.Model1Contair
OnModelCreating(DbModelB
//--
// <auto-generated>
//   This code was generated from a template.
//
//   Manual changes to this file may cause unexpected behavior in your application.
//   Manual changes to this file will be overwritten if the code is regenerated.
// </auto-generated>
//--
namespace ModelAndApi
{
using System;
using System.Data.Entity;
using System.Data.Entity.Infrastructure;
1 reference
public partial class Model1Container : DbContext
{
0 references
public Model1Container()
: base("name=Model1Container")
{
Configuration.LazyLoadingEnabled = false;
Configuration.ProxyCreationEnabled = false;
}
0 references
protected override void OnModelCreating(DbModelBuilder modelBuilder)
{
throw new UnintentionalCodeFirstException();
}
0 references
public virtual DbSet<Media> Media { get; set; }
}
}

API: Au fost adaugate clase noi. Codul a fost de asemenea modificat. Exemple:

```
public void EditData(string pathForEdit, string editEvent, string editPerson, string editPeisaj, string editLoc, string editAltele)
{
    using (var context = new Model1Container())
    {
        IQueryable<Media> mediaList = from elem in context.Media where elem.Path.ToString() == pathForEdit select elem;
        foreach (var data in mediaList.ToList())
        {
            string test = data.ToString();

            if (!String.IsNullOrEmpty(test) && data.Path.ToString() == pathForEdit)
            {
                if (!String.IsNullOrEmpty(editEvent))
                {
                    string eventChange = data.Evenimente.ToString();
                    eventChange = eventChange + ", " + editEvent;
                    data.Evenimente = eventChange;
                }

                if (!String.IsNullOrEmpty(editPerson))
                {
                    string persChange = data.Persoane.ToString();
                    persChange = persChange + ", " + editPerson;
                    data.Persoane = persChange;
                }

                if (!String.IsNullOrEmpty(editPeisaj))
                {
                    string peisajChange = data.Peisaje.ToString();
                    peisajChange = peisajChange + ", " + editPeisaj;
                    data.Peisaje = peisajChange;
                }

                if (!String.IsNullOrEmpty(editLoc))
                {
                    string locChange = data.Locuri.ToString();
                    locChange = locChange + ", " + editLoc;
                    data.Locuri = locChange;
                }

                if (!String.IsNullOrEmpty(editAltele))
                {
                    string alteleChange = data.Altele.ToString();
                    alteleChange = alteleChange + ", " + editAltele;
                    data.Persoane = alteleChange;
                }
                context.SaveChanges();
            }
        }
    }
}
```

```
public string FindData(string result, string editEvent, string editPerson, string editPeisaj, string editLoc, string editAltele)
{
    using (var context = new Model1Container())
    {
        IQueryable<Media> mediaList = from elem in context.Media select elem;
        foreach (var data in mediaList.ToList())
        {
            string test = data.ToString();
            if (!String.IsNullOrEmpty(test))
            {

                if ((!String.IsNullOrEmpty(editEvent) && data.Evenimente.ToString().Contains(editEvent)) ||
                    (!String.IsNullOrEmpty(editPerson) && data.Persoane.ToString().Contains(editPerson)) ||
                    (!String.IsNullOrEmpty(editPeisaj) && data.Peisaje.ToString().Contains(editPeisaj)) ||
                    (!String.IsNullOrEmpty(editLoc) && data.Locuri.ToString().Contains(editLoc)) ||
                    (!String.IsNullOrEmpty(editAltele) && data.Altele.ToString().Contains(editAltele)))
                {
                    result = result + "ID: " + data.Id.ToString() + "\n" +
                        " Path: " + data.Path.ToString() + "\n" +
                        " Moved: " + data.Moved.ToString() + "\n" +
                        " Evenimente: " + data.Evenimente.ToString() + "\n" +
                        " Persoane: " + data.Persoane.ToString() + "\n" +
                        " Peisaje: " + data.Peisaje.ToString() + "\n" +
                        " Locuri: " + data.Locuri.ToString() + "\n" +
                        " Altele: " + data.Altele.ToString() + "\n" +
                        " Data Creare: " + data.DataCreare.ToString() + "\t";
                }
            }
        }
    }
    return result;
}
```

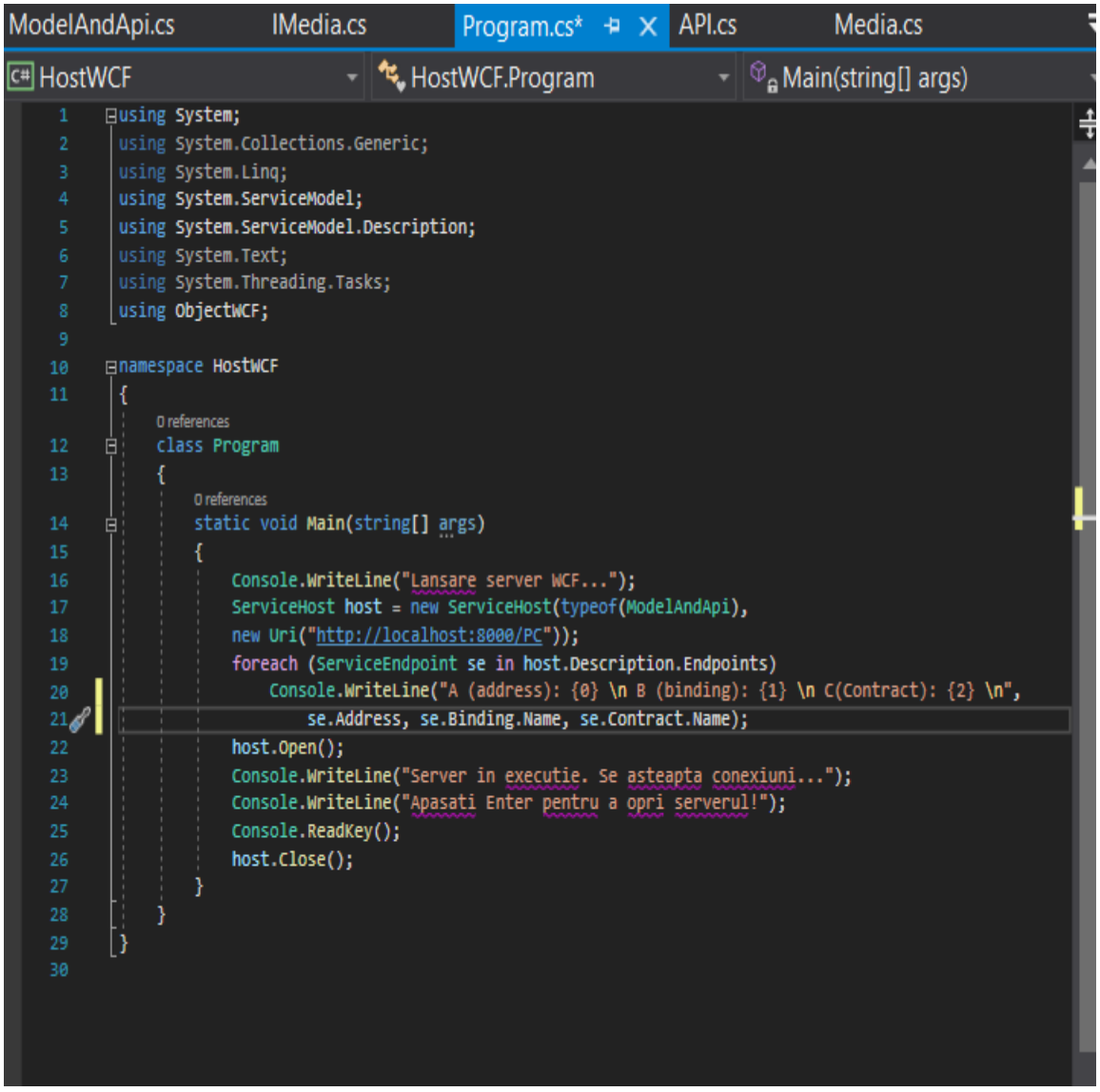
```
public void SaveMedia(string path, string events, string persons, string peisaj, string locatie, string altele, DateTime creationDate)
{
    int check = 1;
    using (var context = new Model1Container())
    {
        foreach (var data in context.Media)
        {
            if (data.Path.ToString() == path)
            {
                check = 0;
            }
        }

        if (check == 1)
        {
            Media newMedia = new Media()
            {
                Path = path,
                Moved = 0,
                Evenimente = events,
                Persoane = persons,
                Peisaje = peisaj,
                Locuri = locatie,
                Altele = altele,
                DataCreare = creationDate.ToString()
            };
            context.Media.Add(newMedia);
            context.SaveChanges();
        }
    }
}
```

ObjectWCF:
S-a creat proiectul cu referinta la ModelAndApi. Au fost adaugate doua clase: Interfata IMedia (echivalentul la InterfaceWCF) si clasa care implementeaza metodele din interfata: ModelAndApi.cs

```cs
using System;
using ModelAndApi;

namespace ObjectWCF
{
    public class ModelAndApi : IMedia
    {
        void IMedia.SaveMedia(string path, string events, string persons,
            string peisaj, string locatie, string altele, DateTime creationDate)
        {
            API api = new API();
            api.SaveMedia(path, events, persons, peisaj, locatie, altele, creationDate);
        }

        void IMedia.DeleteMedia(string path)
        {
            API api = new API();
            api.DeleteMedia(path);
        }

        object IMedia.ShowGridData()
        {
            API api = new API();
            return api.ShowGridData();
        }

        object IMedia.ShowData()
        {
            API api = new API();
            return api.ShowData();
        }

        string IMedia.SaveFile()
        {
            API api = new API();
            return api.SaveFile();
        }

        void IMedia.EditData(string pathForEdit, string editEvent, string editPerson,
            string editPeisaj, string editLoc, string editAltele)
        {
            API api = new API();
            api.EditData(pathForEdit, editEvent, editPerson, editPeisaj, editLoc, editAltele);
        }

        string IMedia.FindData(string result, string editEvent, string editPerson,
            string editPeisaj, string editLoc, string editAltele)
        {
            API api = new API();
            return api.FindData(result, editEvent, editPerson, editPeisaj, editLoc, editAltele);
        }
    }
}
```

```
using System;
using System.ServiceModel;

namespace ObjectWCF
{
    [ServiceContract]
    interface IMedia
    {
        [OperationContract]
        void SaveMedia(string path, string events, string persons,
            string peisaj, string locatie, string altele, DateTime creationDate);
        [OperationContract]
        void DeleteMedia(string path);
        [OperationContract]
        object ShowGridData();
        [OperationContract]
        object ShowData();
        [OperationContract]
        string SaveFile();
        [OperationContract]
        void EditData(string pathForEdit, string editEvent,
            string editPerson, string editPeisaj, string editLoc, string editAltele);
        [OperationContract]
        string FindData(string result, string editEvent,
            string editPerson, string editPeisaj, string editLoc, string editAltele);
    }
}
```

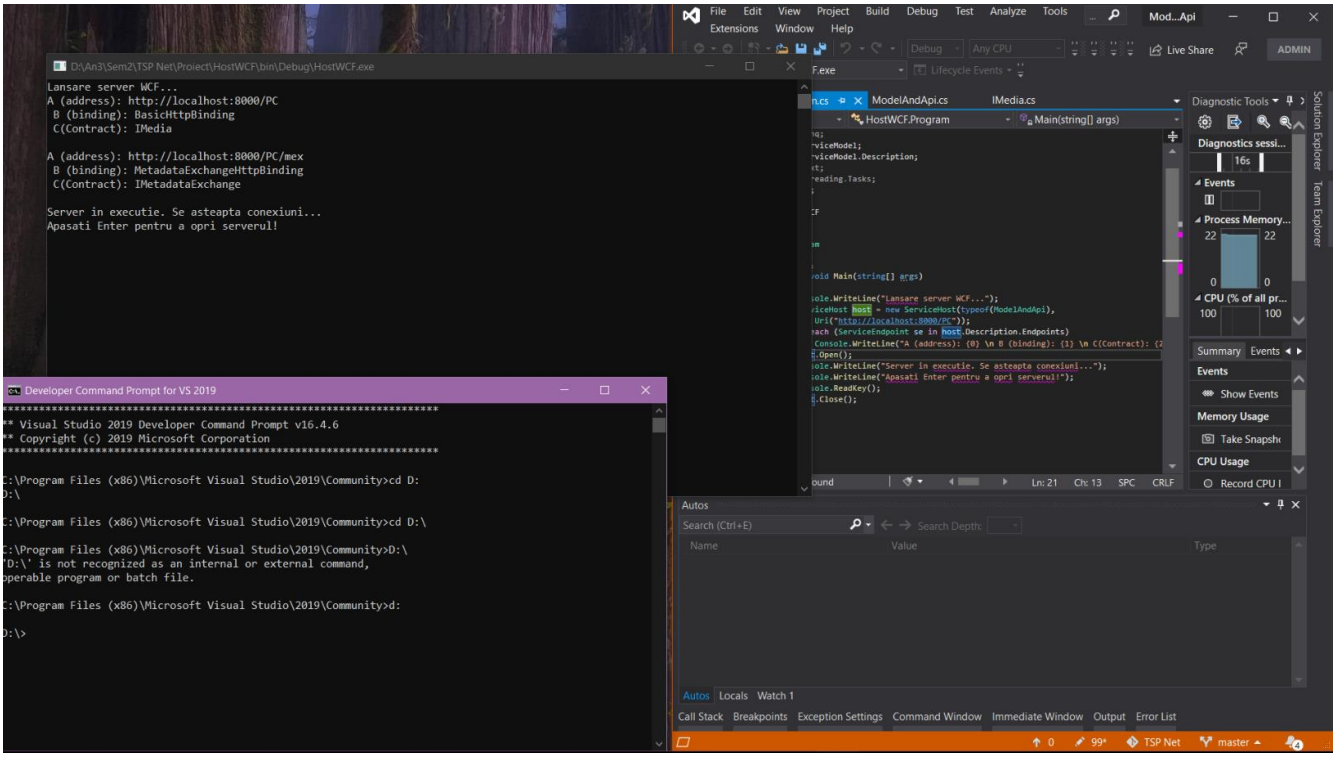
Host:
S-a creat proiectul HostWCF cu referinta la ObjectWCF. S-a adaugat codul pentru completare in program.cs si s-a editat app.config:



App.config* | ModelAndApi.cs | IMedia.cs | Program.cs* | API.cs

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <startup>
        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
    </startup>

  <connectionStrings>
    <add name="ModelAndApiContainer"
      connectionString="metadata=res://*/Model1.csdl|res://*/Model1.ssdl|res://*/Model1.msl;
      provider=System.Data.SqlClient;provider connection string=&quot; data source=DESKTOP-A6HN40I;
      initial catalog=ProiectMedia;integrated security=True;MultipleActiveResultSets=True;
      App=EntityFramework&quot;"
      providerName="System.Data.EntityClient"/>
  </connectionStrings>

  <system.serviceModel>
    <services>
      <service name="ObjectWCF.ModelAndApi" behaviorConfiguration="metadataSupport">

        <endpoint address="http://localhost:8000/PC" binding="basicHttpBinding"
                  contract="ObjectWCF.IMedia" name="BasicHttpBinding_IMedia">
          <identity>
            <dns value="localhost"/>
          </identity>
        </endpoint>
        <endpoint address="mex" binding="mexHttpBinding" contract="IMetadataExchange"
                  name="mexhttp"/>
      </service>
    </services>
    <behaviors>
      <serviceBehaviors>
        <behavior name="metadataSupport">
          <!-- Enables the IMetadataExchange endpoint in services that -->
          <!-- use "metadataSupport" in their behaviorConfiguration -->
          <!-- attribute. -->
          <!-- In addition, the httpGetEnabled and httpGetUrl -->
          <!-- attributes publish-->
          <!-- Service metadata for retrieval by HTTP/GET at the address -->
          <!-- "http://192.168.0.102:8000/SampleService?wsdl" -->
          <serviceMetadata httpGetEnabled="true" httpGetUrl=""/>
          <!-- <serviceMetadata/>-->
          <serviceDebug includeExceptionDetailInFaults="true"/>
        </behavior>
      </serviceBehaviors>
    </behaviors>
  </system.serviceModel>
</configuration>
```

Rezultatul in urma rularii host-ului:



Client:
S-a folosit GUI de la proiectul 1. A fost creat app.config si proxy.cs care arata in felul urmator:

App.config

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <startup>
        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
    </startup>
  <system.serviceModel>
    <bindings>
      <basicHttpBinding>
        <binding name="BasicHttpBinding_IMedia" />
      </basicHttpBinding>
    </bindings>
    <client>
      <endpoint address="http://localhost:8000/PC" binding="basicHttpBinding"
          bindingConfiguration="BasicHttpBinding_IMedia" contract="IMedia"
          name="BasicHttpBinding_IMedia" />
    </client>
  </system.serviceModel>
</configuration>
```

proxy.cs

Miscellaneous Files | IMedia

```
66      [System.Diagnostics.DebuggerStepThroughAttribute()]
67      [System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "4.0.0.0")]
68      public partial class MediaClient : System.ServiceModel.ClientBase<IMedia>, IMedia
69      {
70
71          public MediaClient()
72          {
73          }
74
75          public MediaClient(string endpointConfigurationName) :
76                  base(endpointConfigurationName)
77          {
78          }
79
80          public MediaClient(string endpointConfigurationName, string remoteAddress) :
81                  base(endpointConfigurationName, remoteAddress)
82          {
83          }
84
85          public MediaClient(string endpointConfigurationName, System.ServiceModel.EndpointAddress remoteAddress) :
86                  base(endpointConfigurationName, remoteAddress)
87          {
88          }
89
90          public MediaClient(System.ServiceModel.Channels.Binding binding, System.ServiceModel.EndpointAddress remoteAddress) :
91                  base(binding, remoteAddress)
92          {
93          }
94
95          public void SaveMedia(string path, string events, string persons, string peisaj, string locatie, string altele, System.DateTime creationDate)
96          {
97              base.Channel.SaveMedia(path, events, persons, peisaj, locatie, altele, creationDate);
98          }
99
100         public System.Threading.Tasks.Task SaveMediaAsync(string path, string events, string persons, string peisaj, string locatie, string altele, System.DateTime creationDate)
101         {
102             return base.Channel.SaveMediaAsync(path, events, persons, peisaj, locatie, altele, creationDate);
103         }
104
105         public void DeleteMedia(string path)
106         {
107             base.Channel.DeleteMedia(path);
108         }
109
110         public System.Threading.Tasks.Task DeleteMediaAsync(string path)
111         {
112             return base.Channel.DeleteMediaAsync(path);
113         }
114
115         public object ShowGridData()
116         {
117             return base.Channel.ShowGridData();
118         }
119
120         public System.Threading.Tasks.Task<object> ShowGridDataAsync()
121         {
122             return base.Channel.ShowGridDataAsync();
123         }
```

71 % | No issues found

In urma modificarii GUI prin adaugarea instantei clasei MediaClient codul arata in felul urmator, exemplu:

```
namespace GUI
{
    public partial class Form1 : Form
    {
        MediaClient newMediaClient = new MediaClient();
        string imgLocation = "";
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void buttonAdd_Click(object sender, EventArgs e)
        {
            try
            {
                OpenFileDialog fileDiag = new OpenFileDialog();
                fileDiag.Filter = "JPG Files (*.jpg)|*.jpg|GIF Files (*.gif)|*.gif|PNG Files (*.png)|*.png|MP4 Files (*.mp4)|*.mp4|All Files (*.*)|*.*";
                fileDiag.Title = "Select Media";
                if (fileDiag.ShowDialog() == DialogResult.OK)
                {
                    imgLocation = Path.GetFullPath(fileDiag.FileName.ToString());
                    pictureBox.ImageLocation = imgLocation;
                }
            }
            catch (Exception except)
            {
                MessageBox.Show(except.Message);
            }
        }

        private void buttonSave_Click(object sender, EventArgs e)
        {
                byte[] img = null;
                FileStream fs = new FileStream(imgLocation, FileMode.Open, FileAccess.Read);
                BinaryReader br = new BinaryReader(fs);
                img = br.ReadBytes((int)fs.Length);
                newMediaClient.SaveMedia(imgLocation, textBoxEvent.Text, textBoxPerson.Text, textBoxPeisaj.Text, textBoxLoc.Text, textBoxAltele.Text, dateTimePicker1.Value.Date);

        }

        private void buttonDelete_Click(object sender, EventArgs e)
        {
                string pathToDelete = textBoxDeleteEditPath.Text;
                newMediaClient.DeleteMedia(pathToDelete);
        }
```

```
    private void buttonEdit_Click(object sender, EventArgs e)
    {

            string pathForEdit = textBoxDeleteEditPath.Text;
            string editEvent = textBoxDeleteEditEvent.Text;
            string editPerson = textBoxDeleteEditPerson.Text;
            string editPeisaj = textBoxDeleteEditPeisaj.Text;
            string editLoc = textBoxDeleteEditLoc.Text;
            string editAltele = textBoxDeleteEditAltele.Text;

            newMediaClient.EditData(pathForEdit, editEvent, editPerson, editPeisaj, editLoc, editAltele);
    }
```

Rezultatul in urma pornirii serverului si clientului in VS2019 ca administrator: