

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:

API:

Au fost adaugate clase noi. Codul a fost de asemenea modificat pentru a nu mai exista erori si pentru corectitudine. Exemple:

```
public void EditData(string pathForEdit, string editEvent, string editPerson, string editPeisaj, string editLoc, string editAltele)
{
    using (var context = new ModelContainer())
    {
        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 Create: " + data.DataCreate.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,
                DataCreate = creationDate.ToString()
            };
            context.Media.Add(newMedia);
            context.SaveChanges();
        }
    }
}

```