Subsemnata Bulboaca Maria declar pe propria raspundere ca acest cod nu a fost copiat din Internet sau din alte surse.

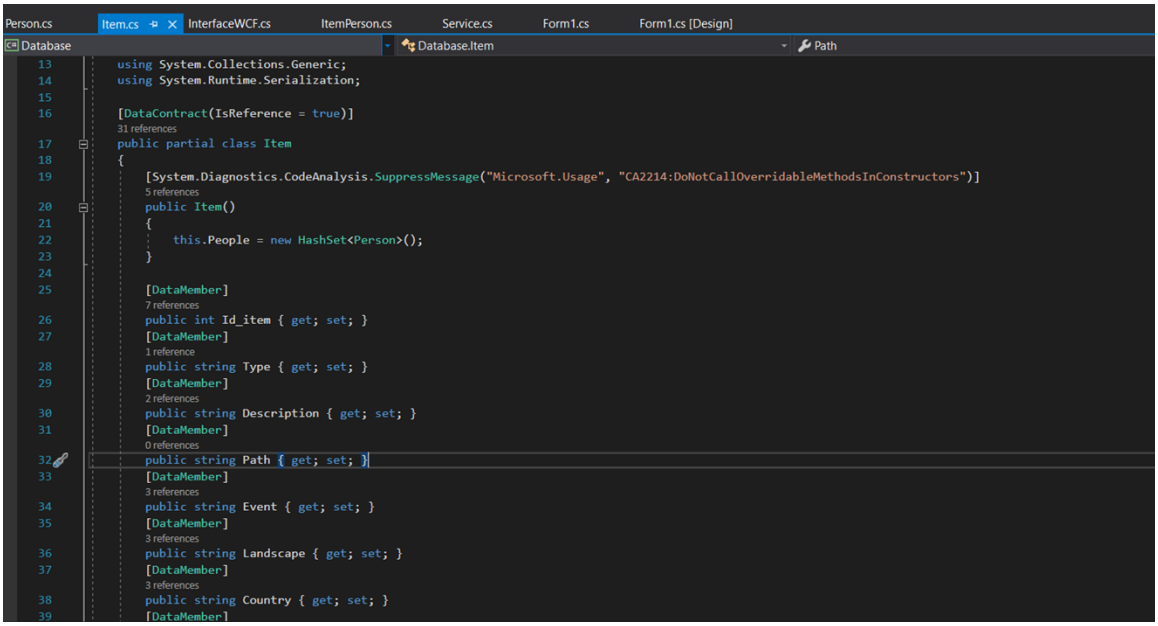
Pentru documentare am folosit urmatoarele surse:

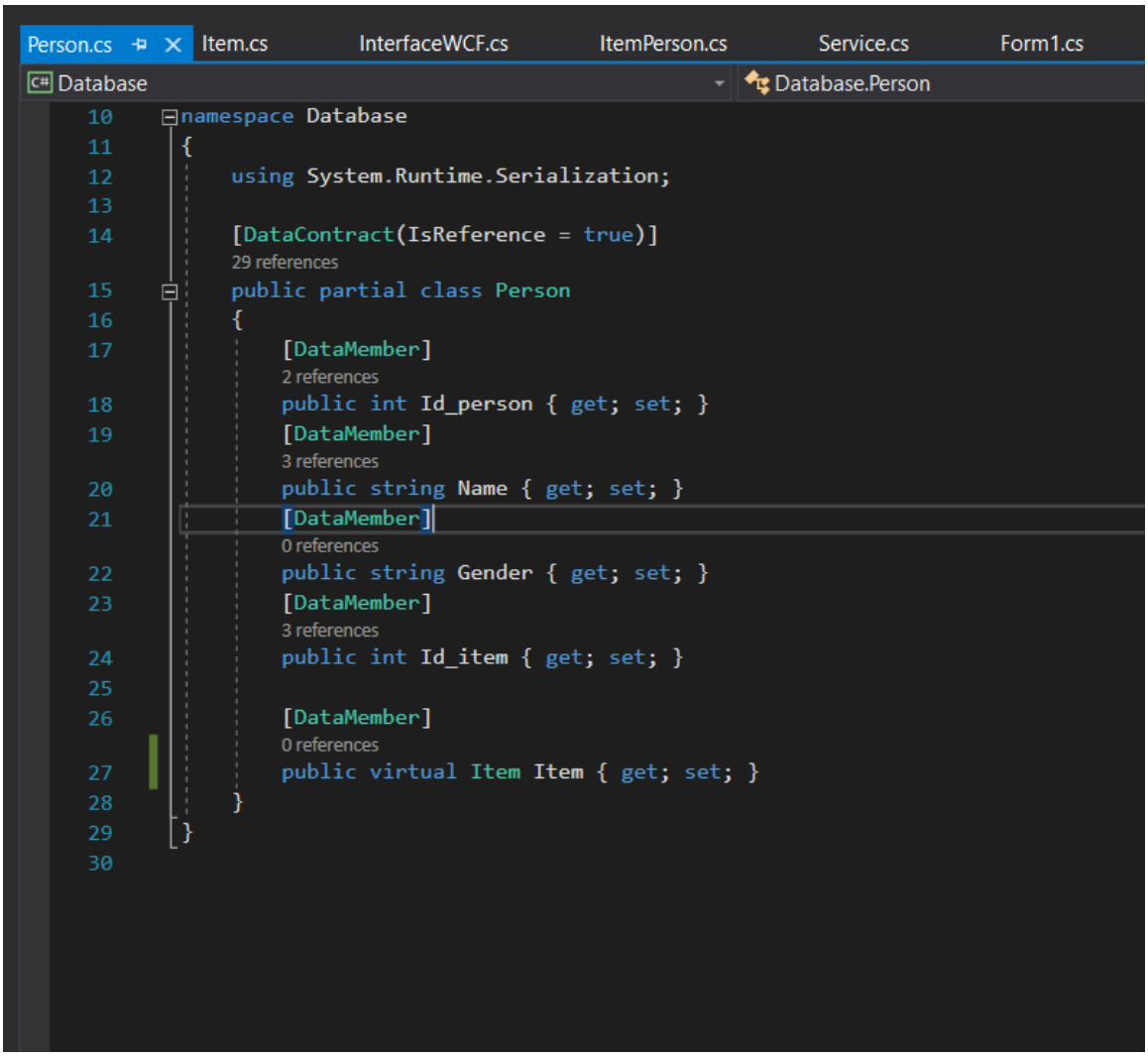
- https://profs.info.uaic.ro/~iasimin/Laborator%20C%20S%20H/Laborator%20WCF%202020.pdf

Proiect 2 - documentatie

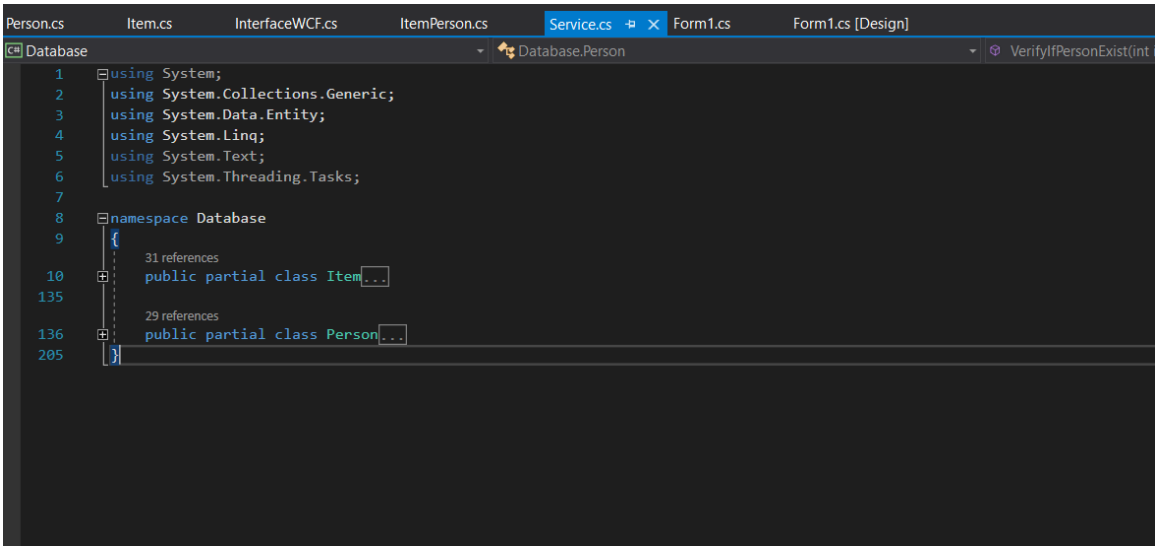
1. API - update-uri

Pentru clasele de la baza de date am adaugat atributele DataContract si DataMember.





Am creat si adaugat la proiect alte clase “partial” Item si Person in care am scris codul pentru metode(API). Aceste clase le-am adaugat intr-un fisier "Service".



2. Clasa "Item" - metode API

2.1 Metode de adaugare

- `public int AddItem()`

- metoda de adaugarea a unui item in baza de date

- output: va returna id-ul itemului care a fost adaugat in baza de date , iar in caz de eroare va returna 0.

```
1 reference
public int AddItem()
{
    using (PhotoModelContainer context = new PhotoModelContainer())
    {
        int bResult = 0;
        if (this.Id_item == 0)
        {
            var it = context.Entry<Item>(this).State = EntityState.Added;
            context.SaveChanges(); bResult = this.Id_item;
        }
        return bResult;
    }

}
```

2.2 Metode de cautare

- `public Item GetById(int id)`

- cautarea unui item dupa id

- input: va avea un singur parametru de tip int si care reprezinta id-ul itemului care trebuie cautat.

- output: va returna itemul cu id-ul gasit

```
1 reference
public Item GetById(int id)
{
    using (PhotoModelContainer context = new PhotoModelContainer())
    {
        return context.Items.SingleOrDefault(item => item.Id_item == id);
    }
}
```

- public List<Item> FilterItems(FilterModel filter)

  - cautarea unor itemuri in functie de filtrele date ca parametru.

  - input: va avea un singur parametru de tip FilterModel

```
3 references
public class FilterModel
{
    2 references
    public string Type { get; set; }
    2 references
    public string Event { get; set; }
    2 references
    public string Landscape { get; set; }
    2 references
    public string Country { get; set; }
    2 references
    public int? Year { get; set; }
    1 reference
    public List<string> Persons { get; set; }
}
```

  - output: va returna lista cu item-uri in functie de filtrele date. Daca nu sunt date filtre vor fi afisate toate itemurile.

```
1 reference
public List<Item> FilterItems(FilterModel filter)
{
    using (PhotoModelContainer context = new PhotoModelContainer())
    {
        var items = context.Items.AsQueryable();
        var persons = context.People.AsQueryable();

        if (filter.Type != "")
            items = items.Where(item => item.Type == filter.Type);
        if (filter.Event != "")
            items = items.Where(item => item.Event == filter.Event);
        if (filter.Landscape != "")
            items = items.Where(item => item.Landscape == filter.Landscape);
        if (filter.Country != "")
            items = items.Where(item => item.Country == filter.Country);
        if (filter.Year != null)
            items = items.Where(item => item.Date_created.Year == filter.Year);

        List<Person> personsList = new List<Person>();

        foreach (var person in filter.Persons)...

        List<Item> final_items = new List<Item>();
        foreach (var person in personsList)...

        if (final_items.Count != 0)
            items = final_items.AsQueryable();
        return items.ToList();
    }
}
```

2.3 Metode de stergere

- public void RemoveItem(int idItem)

- va sterge un item din baza de date

- input: va avea un singur parametru de tip int care reprezinta id-ul itemului de sters

```
public void RemoveItem(int idItem)
{
    using (PhotoModelContainer context = new PhotoModelContainer())
    {
        Person person = new Person();
        var result = context.Items.SingleOrDefault(item => item.Id_item == idItem);
        if (result != null)
        {
            person.RemovePersons(idItem);
            context.Items.Remove(result);
            context.SaveChanges();
        }
    }
}
```

2.4 Metode de update

- public string UpdateItem(int id, Item newItem)

- va face update pentru un item dat.

- input: va avea ca parametri id-ul itemului pentru care trebuie sa se faca update si continutul pentru update.

- output: mesaj de succes sau de eroare.

```
public string UpdateItem(int id, Item newItem)
{
    using (PhotoModelContainer context = new PhotoModelContainer())
    {
        var result = context.Items.First(item => item.Id_item == id);
        result.Description = newItem.Description;
        result.Event = newItem.Event;
        result.Date_created = newItem.Date_created;
        result.Landscape = newItem.Landscape;
        result.Country = newItem.Country;
        try
        {
            context.SaveChanges();
            return "Updated successfully";
        }
        catch
        {
            return "Updated not successful. Try again";
        }
    }
}
```

2.5 Metode de afisare

- public List<Item> GetAllItems() - va oferi toate itemurile in baza de date

```
public List<Item> GetAllItems()
{
    Person person = new Person();
    using (PhotoModelContainer context = new PhotoModelContainer())
    {
        var items = context.Items;

        foreach (var item in items)
        {
            item.People = person.GetPersonsForItem(item.Id_item).ToList();

        }
        return items.ToList();
    }
}
```

3. Clasa "Person" - metode API

3.1 Metode de adaugare

- public int AddPerson()

- metoda de adaugarea a unei persoane in baza de date

- output: va returna id-ul persoanei care a fost adaugata in baza de date , iar in caz de eroare va returna 0.

```
1 reference
public int AddPerson()
{
    using (PhotoModelContainer context = new PhotoModelContainer())
    {
        int bResult = 0;
        if (this.Id_person == 0)
        {
            var it = context.Entry<Person>(this).State = EntityState.Added;
            context.SaveChanges(); bResult = this.Id_person;
        }
        return bResult;
    }

}
```

3.2 Metode de cautare

- public List<Person> GetPersonsForItem(int id)

- input : va avea un singur parametru de tip int care reprezinta id-ul itemului pentru care trebuie sa fie persoanele.

- output: o lista cu persoane

```
3 references
public List<Person> GetPersonsForItem(int id)
{
    using (PhotoModelContainer context = new PhotoModelContainer())
    {
        return context.People.Where(person => person.Id_item == id).ToList();
    }
}
```

- public int VerifyIfPersonExist(int id, string name)

- verifica daca exista o persoana in baza de date

- input: va avea doi parametri , unul pentru id-ul unui item si celalalt numele persoane

- output: va returna numarul de apartitii a numelui persoanei

```
1 reference
public int VerifyIfPersonExist(int id, string name)
{
    using (PhotoModelContainer context = new PhotoModelContainer())
    {
        return context.People
            .Where(person => person.Id_item == id)
            .Where(p => p.Name == name)
            .Count();
    }
}
```

- public List<string> GetNamePersons() - va returna numele persoanelor din baza de date

```
1 reference
public List<string> GetNamePersons()
{
    using (PhotoModelContainer context = new PhotoModelContainer())
    {
        return context.People.Select(person => person.Name).Distinct().ToList();
    }
}
```

3.3 Metode de stergere

- public void RemovePersons(int id)

  - va sterge persoanele pentru un item

```
public void RemovePersons(int id)
{
    using (PhotoModelContainer context = new PhotoModelContainer())
    {
        var persons = GetPersonsForItem(id);
        foreach (var person in persons)
        {
            context.People.Attach(person);
            context.People.Remove(person);
        }
        context.SaveChanges();
    }
}
```

- public void RemovePerson(Person person)

  - va sterge o persoana

```
1 reference
public void RemovePerson(Person person)
{
    using (PhotoModelContainer context = new PhotoModelContainer())
    {
        context.People.Remove(person);
        context.SaveChanges();
    }
}
```