Desktop Application for Replacing the Traditional Register

Ioan – Albert Florea, Delilah Florea
Informatics and Computer Science Department
"Samuel von Brukenthal" National College
Huet Square, No 5, Sibiu, Romania
berti_florea@yahom.com, delilah_florea@yahoo.com

ABSTRACT: *This work presents Catalog Virtual, a desktop application designed for computers, which aims to record the students' performance in school. Also, another target is to replace the traditional register which is made of paper. In this way we reduce deforestation protecting the nature and everyone can access the register very easily through a computer. For parents it is a more efficient way to access the catalogue, because they do not have to leave from work or to do any other things to come to school to see the grades of their kids. The application was developed in Microsoft Visual Studio Community 2017 v15.9.3, with tests carried to the compiler of programming environment and on different laptops with different systems.*

Keywords: Desktop Application, ASP.NET, Microsoft VS, MSSQL, Entity Framework, WEB API2, Windows Presentation Foundation (WPF), Data Transfer Objects (DTO)

Received: 24 October 2018, Revised 29 November 2018, Accepted 2 December 2018

DOI: 10.6025/jet/2019/10/1/10-14

1. Introduction

Nowadays, almost everyone has contact with a computer, if it is at work, in school or at home, etc. we use them and we need them to do our work more efficiently. Let's take the following example: you are a parent and you work too many hours a day. That means that you don't spend too much time with your children, which is not good. Your child might be careless of the grades he receives in school and you need to keep tracking of his situation and to step in. But how can you do that when you work too much and don't have time to go to his school and talk to his teacher? This is the moment where CatalogVirtual steps in.

Catalog Virtual is an application designed to replace the traditional register with grades and to keep tracking of student's grades and attendance in class. Teachers don't need different versions of the application for each class, because it is constructed in such a way that they can access every class they teach to from just one app. The idea of developing this application derived in the 11th grade while I was taking part in an informatics contest. It uses modern programming techniques like DTO (Data Transfer Objects), MVVM (Model-View-View Model) which help at the efficiency of the product.

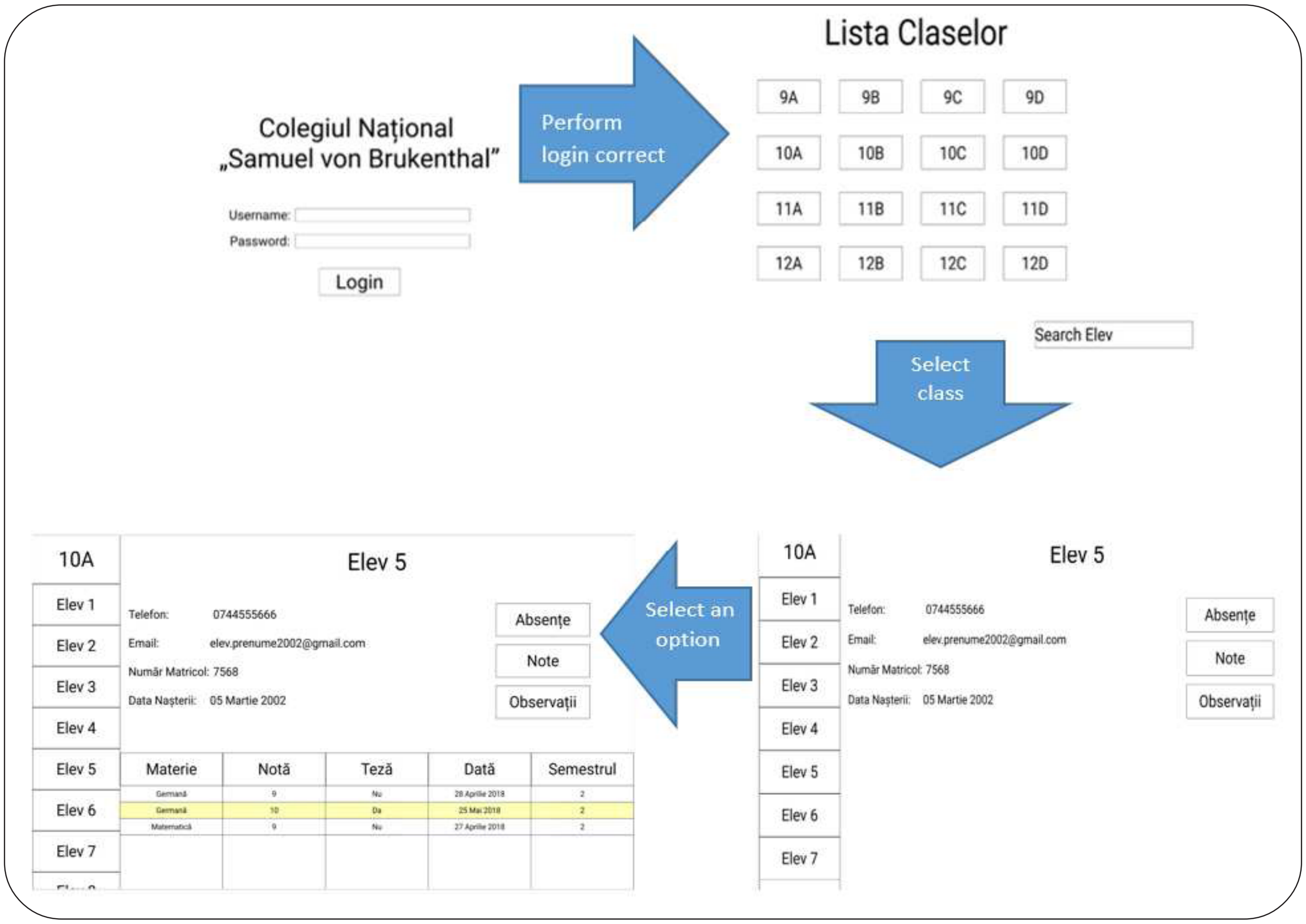
Some schools in Romania have already implemented such a system in their school, but here in Sibiu we don't have something like that. This is why we tried to develop in our own way such an app, which doesn't operate on the database directly, because the performance would be very slow, in case many users access the app. Those type of software's need approval from institutions like the school itself, where it is implemented, the county board of education, but also from the parents of the student (the well-known European General Data Protection Regulation (GDPR)).

The next section describes the UI and application functionality. Section 3 is then dedicated to aspects concerning the application development software: the programming environment, database layer and services. Section 4 discusses themes that constitute avenues for further study, concluding the paper.

2. User Interface and Application Functionality

The first view you see when you start the application is a login view, which show the school name and the sections in which you can enter your credentials, more exactly the username and the password.

Afterwards, if your credentials are correct, depending on the type of user you are (teacher, student) you see the next view. We will take the teachers path, because it contains also the student's path and it has more functionalities. On the next page you can see the full path of user (teacher) who teaches at every class and can see different grades or behavior issues.



Telefon: 0744555666
Email: elev.prenume2002@gmail.com
Număr Matricol: 7568
Data Nașterii: 05 Martie 2002

Materie	Notă	Teză	Dată	Semestrul
Germană	9	Nu	28 Aprilie 2018	2
Germană	10	Da	25 Mai 2018	2
Matematică	9	Nu	27 Aprilie 2018	2

Figure 1.CatalogVirtual workflow

In this picture we can see a follow through of the application, in case we login as a teacher. For each student is shown the following information: phone number, email, serial number, birth date. On the right hand side, there are displayed three different buttons. If pressed, each button displays information in the bottom half of the view related to the button clicked before. In the example above, the button which displays the grades was pressed; therefore we can see student 5 grades.

3. Application Development Software

3.1. Programming Environment – Microsoft Visual Studio

Application development involves the following steps:

• Database design

• Implementation of web services

• Designing the application design

• Testing the application

• Changes required after testing

• Completion and approval of the application

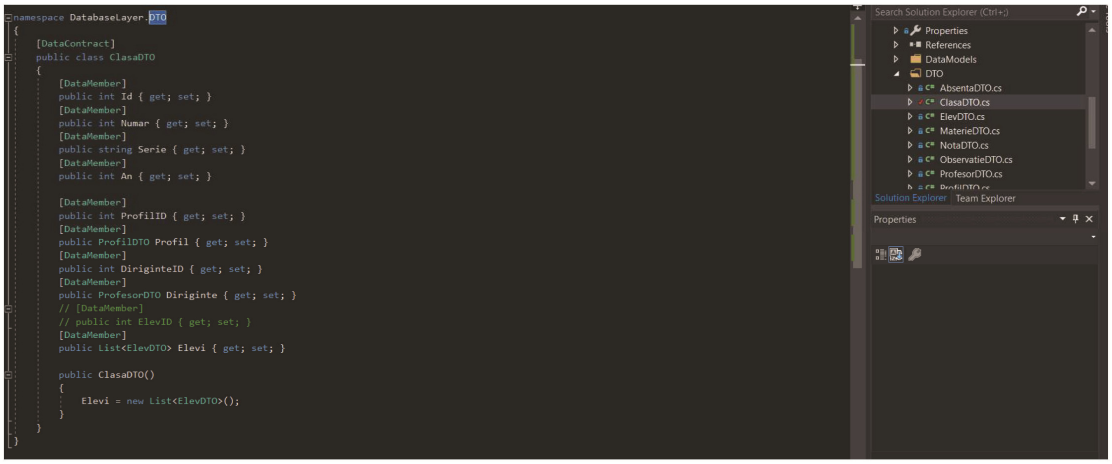
CatalogVirtual was developed in Microsoft Visual Studio 2017 v15.9.3, using C# language skills. Throughout its development I used different programming techniques like DTO (Data Transfer Object), MVVM (Model-View-View Model). The development of the application included learning a little bit of XAML language in order to edit the design part of the views and to establish the connections between the Views and ViewModels. In the background of CatalogVirtual there is a web service constantly running, which helps completing CRUD (Create, Read, Update, Delete) operations on data. Other technologies or programming environment used are GitKraken, Figma, .NET Framework, Entity Framework and REST ful web services.

3.2. Construction of the Application

CatalogVirtual is divided in 3 big projects. The first one, Database Layer, handles, using .NET Framework, how we convert data into DTO's in order to do as few actions on the database as possible, which means faster runtime of the application. The second project, NewWebService, a WEB API project, contains the part in which CRUD operations are done, through RESTful applications. REST, or REpresentational State Transfer, is an architectural style for providing standards between computer systems on the web, making it easier for systems to communicate with each other [2]. Catalog Desktop App is the final project, which also contains the front end of the application. There can be seen all the views, which are going to be used by the clients. This project brings everything together and can be also seen.

3.2.1. Database Layer

This is the starting part of the application. In this part, the data is contracted so that it can be transferred easier. In the picture below, we can see that for a class the information of a class like Id, number of the class, series is contracted but also there exist list of students to achieve the students from the wanted class. On the right hand side, of the figure 2 there can be seen that for other things like students, teachers, grades, etc. Exist other classes of DTO. This must be done every time for every object which we use, so data can be transferred easier.

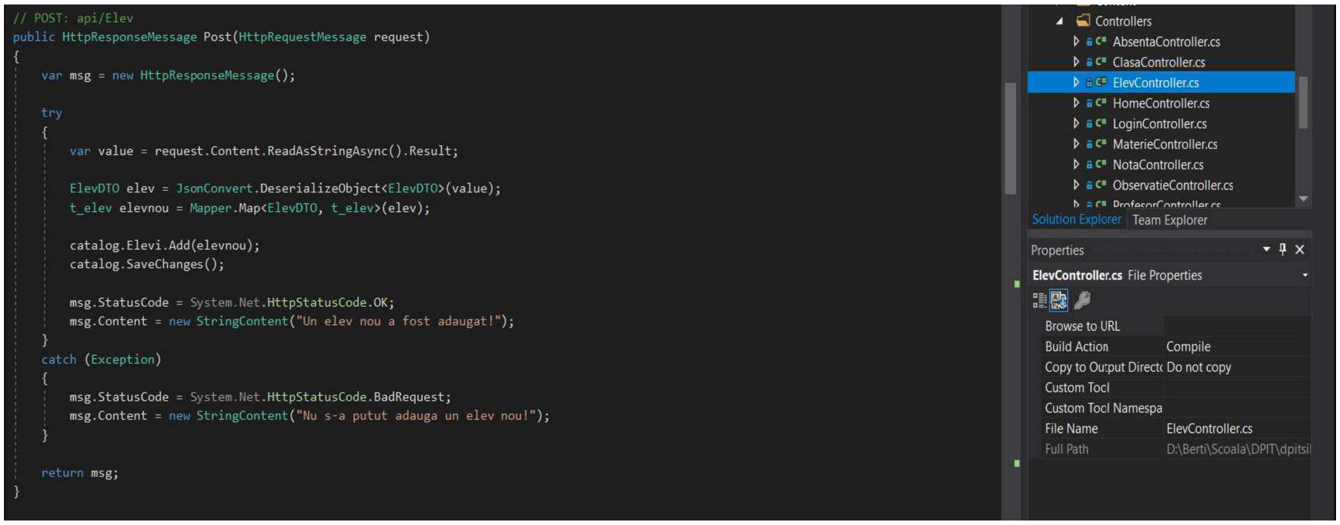


```
namespace DatabaseLayer.DTO
{
    [DataContract]
    public class ClasaDTO
    {
        [DataMember]
        public int Id { get; set; }
        [DataMember]
        public int Numar { get; set; }
        [DataMember]
        public string Serie { get; set; }
        [DataMember]
        public int An { get; set; }

        [DataMember]
        public int ProfilID { get; set; }
        [DataMember]
        public ProfilDTO Profil { get; set; }
        [DataMember]
        public int DiriginteID { get; set; }
        [DataMember]
        public ProfesorDTO Diriginte { get; set; }
        // [DataMember]
        // public int ElevID { get; set; }
        [DataMember]
        public List<ElevDTO> Elevi { get; set; }

        public ClasaDTO()
        {
            Elevi = new List<ElevDTO>();
        }
    }
}
```

Figure 2.CatalogVirtual: Database Layer

3.2.2 New Web Service

In this project, the data transfer and the CRUD operations takes place. For each table we have in our database there will exist a controller, which contains the CRUD operations. In the picture below, you can see for a student a POST operation (Create). If the student is successfully created and saved in the database, a message will be shown. If not, then an unsuccessful message will be shown.



```
// POST: api/Elev
public HttpResponseMessage Post(HttpRequestMessage request)
{
    var msg = new HttpResponseMessage();

    try
    {
        var value = request.Content.ReadAsStringAsync().Result;

        ElevDTO elev = JsonConvert.DeserializeObject<ElevDTO>(value);
        t_elev elevnou = Mapper.Map<ElevDTO, t_elev>(elev);

        catalog.Elevi.Add(elevnou);
        catalog.SaveChanges();

        msg.StatusCode = System.Net.HttpStatusCode.OK;
        msg.Content = new StringContent("Un elev nou a fost adaugat!");
    }
    catch (Exception)
    {
        msg.StatusCode = System.Net.HttpStatusCode.BadRequest;
        msg.Content = new StringContent("Nu s-a putut adauga un elev nou!");
    }

    return msg;
}
```

Controllers
- AbsentaController.cs
- ClasaController.cs
- ElevController.cs
- HomeController.cs
- LoginController.cs
- MaterieController.cs
- NotaController.cs
- ObservatieController.cs

Solution Explorer | Team Explorer

Properties

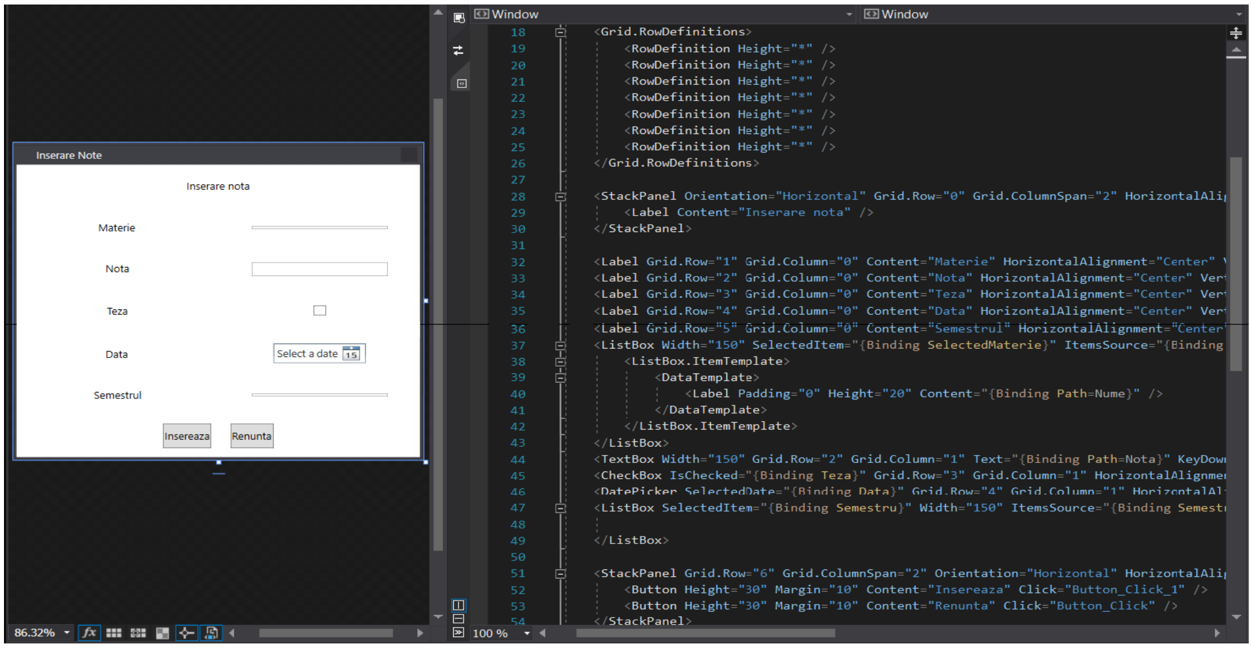
ElevController.cs File Properties

Browse to URL	
Build Action	Compile
Copy to Output Directo	Do not copy
Custom Tool	
Custom Tool Namespa	
File Name	ElevController.cs
Full Path	D:\Berti\Scoala\DPIT\dpitsil

Figure 3. CatalogVirtual: try-catch protocol for data insertion in database

3.2.3 Catalog Desktop App

This part of the project brings everything together because it contains the front end part of the application. Here it is used the MVVM. For this part there are needed also XAML language knowledge. In the following example there can be seen, on the left the design part of a view and on the right the coding part of the view. In this one we can see a view for inserting grades, which if you click the button "Insereaza", the POST method in the web service is going to be done, which means that a new grade is going to be inserted in the database of the grades.



```
<Grid.RowDefinitions>
    <RowDefinition Height="*" />
    <RowDefinition Height="*" />
    <RowDefinition Height="*" />
    <RowDefinition Height="*" />
    <RowDefinition Height="*" />
    <RowDefinition Height="*" />
    <RowDefinition Height="*" />
</Grid.RowDefinitions>

<StackPanel Orientation="Horizontal" Grid.Row="0" Grid.ColumnSpan="2" HorizontalAli
    <Label Content="Inserare nota" />
</StackPanel>

<Label Grid.Row="1" Grid.Column="0" Content="Materie" HorizontalAlignment="Center"
<Label Grid.Row="2" Grid.Column="0" Content="Nota" HorizontalAlignment="Center" Ver
<Label Grid.Row="3" Grid.Column="0" Content="Teza" HorizontalAlignment="Center" Ver
<Label Grid.Row="4" Grid.Column="0" Content="Data" HorizontalAlignment="Center" Ver
<Label Grid.Row="5" Grid.Column="0" Content="Semestrul" HorizontalAlignment="Center"
<ListBox Width="150" SelectedItem="{Binding SelectedMaterie}" ItemsSource="{Binding
    <ListBox.ItemTemplate>
        <DataTemplate>
            <Label Padding="0" Height="20" Content="{Binding Path=Nume}" />
        </DataTemplate>
    </ListBox.ItemTemplate>
</ListBox>
<TextBox Width="150" Grid.Row="2" Grid.Column="1" Text="{Binding Path=Nota}" KeyDow
<CheckBox IsChecked="{Binding Teza}" Grid.Row="3" Grid.Column="1" HorizontalAlignme
<DatePicker SelectedDate="{Binding Data}" Grid.Row="4" Grid.Column="1" HorizontalAl
<ListBox SelectedItem="{Binding Semestru}" Width="150" ItemsSource="{Binding Semestr

</ListBox>

<StackPanel Grid.Row="6" Grid.ColumnSpan="2" Orientation="Horizontal" HorizontalAli
    <Button Height="30" Margin="10" Content="Insereaza" Click="Button_Click_1" />
    <Button Height="30" Margin="10" Content="Renunta" Click="Button_Click" />
</StackPanel>
```

Figure 4. CatalogVirtual: code description of Insert interface

4. Conclusions and Future Developments

The digitalization process in schools represents a challenge for many East-European countries. At university level the institutions start to introduce such solutions like CatalogVirtual but this might involve high costs. The alternative for high schools could be developing such "in house applications", tested for a period of time in parallel with traditional registering method on paper and finally after detecting potential inconsistencies, errors to be remediated and using only the digital registering method. Our intention is to develop also an Android based application, so the users can feel free with the application even on their phone. Inserting some charts with the class and student progress for the teachers would be great, so that the teachers at the end of the semester do not need to waste time with them. Also, another feature that might be introduce refers to parental email notification about the child's absence. I realize that there is a lot to improve, but there is a lot of potential in this application which is a serious problem for us nowadays.

References

[1] Florea A.C., Florea D. (2017). TestIT, *In*: Proceedings of the Ninth International Conference on Applied Informatics Imagination, Creativity, Design, Development (ICDD), May 25-27, 2017, Sibiu, Romania.

[2] Code Academy, https://www.codecademy.com/articles/what-is-rest, 3 March 2019

[3] MVVM- Messenger and View Services in MVVM by Laurent Bugnion March 2013, https://msdn.microsoft.com/en-us/magazine/jj694937.aspx

[4] Recommendations and best practices for implementing MVVM and XAML/.NET applications, January 30, 2015, (updated on June 15, 2018) by Rico Suter, https://blog.rsuter.com/recommendations-best-practices-implementing-mvvm-xaml-net-applications/

[5] Microsoft XAML in WPF, https://docs.microsoft.com/en-us/dotnet/framework/wpf/advanced/xaml-in-wpf, 03/30/2017

[6] Microsoft Security, Authentication, and Authorization in ASP.NET Web API 12/11/2012, https://docs.microsoft.com/en-us/aspnet/web-api/overview/security/