SOFTWARE APPLICATION FOR MANAGING STUDENTS ADMISSION TO THE UNIVERSITY

Elena-Georgiana Deaconu[1], Cosmin Ştirbu[2], Florentina Enescu[3]
Dept. of Electronics, Communications and Electrical Engineering, University of Pitesti, Romania
[1]georgydeaconu1993@gmail.com, [2]cosmin.stirbu@upit.ro,enescu_flor@yahoo.com

Keywords: software application, networking, Microsoft .NET Framework 3.5, Visual Studio, SQL Server.

Abstract: *The paper presents all the steps for designing and implementing a software application for students' registration and matriculation and their allocation to a faculty depending on their chosen options. The system is composed of a Client application, which can be installed on any computer, and communicates through IP address with the database on the Server.*

1. INTRODUCTION

The application aims to present all stages of a software system for enrollment and student admission and their distribution depending on the options chosen. The system consists of client application that can be installed on any computer, communicating with the database located on a server via an IP.

The application is developed in Visual Studio and can be installed on any number of computers [1,2]. This way, every student can fill in their personal data as well as their options depending on their priorities [1,2].

In order to implement this system, the application and the database, we used the following developer tools: Visual Studio and Sql Server.

The database uses Sql Server 2012 and saves all the information regarding the student and the options from which he can choose.

In our school systems are similar, but my application differs primarily by the fact that there is a user and an administrator so that the pupil's personal data cannot be seen by other users. I guess in the country are applications like this.

2. DESIGN AND IMPLEMENTATION

The purpose of this application is to use a computer inside the university, where every student can access it in order to apply for a university program. The student can fill in this information on his own, without assistance from a representative from the university.

After following some steps where the student inserts his personal information, data regarding the previous schooling and the chosen options, he must talk to the university personnel in order to verify these data.

Since the application is "Networking", the personnel can access and visualize all the students enrolled wherever the application is installed by accessing the database through IP address.

The application can run under Windows, minimum Windows 7, as it necessary to run along .NET Framework 3.5.

If the host computer does not run this framework, the application will automatically download and install it. [3, 4]

After installing the executable program made with Visual Studio, a shortcut will be created on the desktop call "Admission to University".

In order to start the application, it is necessary to double click the shortcut on the desktop and the application will be opened in Maximized mode.

The system contains a desktop application for the student to insert his personal information and for the university personnel to distribute the student on specialties.

Module Desktop Application

The purpose of this module is to get students' information in order to enroll him to a certain specialty. When launching this application, a login window will open requiring for user and password.

The application is completely configurable and can be used for any faculty.

It is necessary to create an account when the student enrolls and a unique number will be associated in the system. Thus, the application will always know which student applies for that time. This account is automatically created and has the type "User".

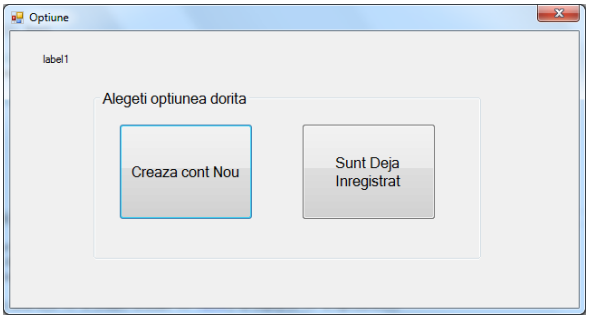

Fig.1. The registration form

The application can handle more types of users:

User: when he logs in, he can only access the information he introduced and can update these if the period for applying has not yet expired.

In order for a student to be taken into account for repartition on his desired specialty, the information must be validated by an Admin. It does not suffice to fill in the mandatory information.

After the student inserted all the necessary information, an Admin will verify the correctness of data. If he considers them correct, he can validate the respective student and the student will be taken into consideration for the repartition.

Because the application uses a Login method, composed from unique user and password, after the Admin validates the application form, the student can change any information and this fact requires another Admin validation.

The second type of user is **Admin**. When they login, they are directed to another page

Where:

- They can visualize all the enrolled users.
- They can modify the information students have inserted.
- They can start the automatic repartition process depending on chosen specialties.
- They can modify the name of the options and the total number of positions with school fee or paid by the state.
- They can configure the application for any faculty.
- The can matriculate the students after they have enrolled.

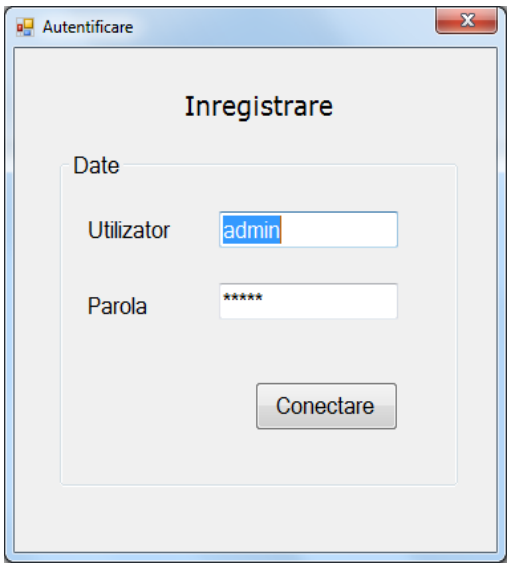

Fig.2. The login form

Fig. 3.The class diagram of the project

Description class diagram is shown below:

- **AdresaServer** - set the address of the database server (the connection between the application and database)
- **Autentificare** - form where the user authenticates
- **DatePersonale** - form where the student enter their personal data
- **FacultateAbsolvita** - if the student has graduated from university
- **FirstPage** - page establishing the type of user (new or existing user)
- **HighSchool** - graduated high school information
- **Inmatriculare** - after the distribution was conducted following registration process
- **Utilizatori** - are displayed both types of users : Administrators and Users
- **ListaOptiuni** - the list of options that students can choose
- **MainPage** - the main form in which administrators can view information about students enrolled , and the list of options
- **Optiuni** - the form in which the student chooses the desired options

The database is created with Microsoft Sql Server 2012 and contains all the information necessary for the application to work.

The database schema is shown in Fig. 4.

As we can notice in the schema, a main table, **Users**, links all the other tables. The primary key, *UserID* from *Users* table is also foreign key for the related tables. All these relations have the type: *Cascade Update*, *Cascade Delete*. When the system administrator wishes to erase a student, he will erase it only from the *Users* table because it will be automatically erased from the other tables (e.g., *ChosenOptions, Students*, *GraduatedFaculties*) due to the link between them.

The *Options* tables are not related to any other table because it keeps the options from where the student can choose.

A series of tables responsible with configuring the faculties can be also found in the database, such as *HighSchools*.

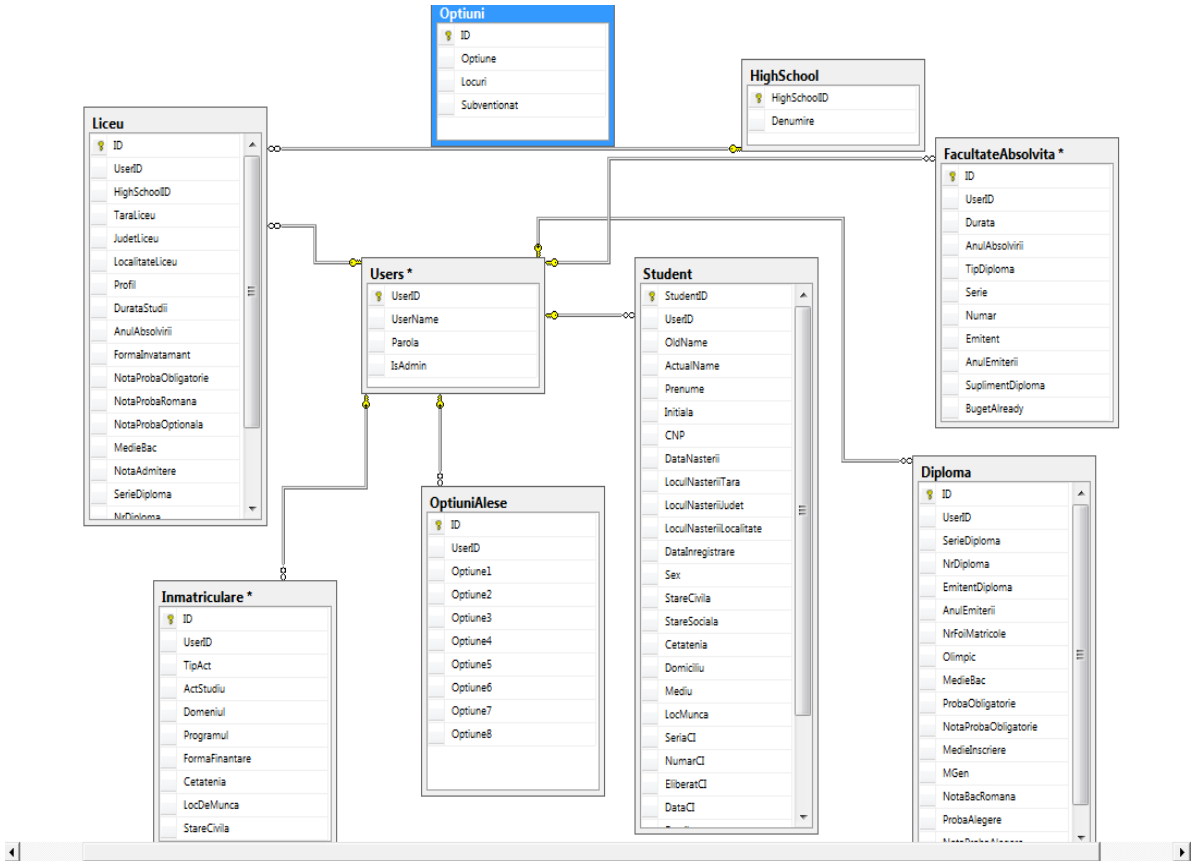

Fig.4. The database schema

3. CONCLUSIONS

The application had the purpose of enrolment and matriculation of students to the Faculty of Electronics, Communication and Computers, Computers field.

As such an application is not yet implemented in the university, it seemed very interesting for me to implement one. From the tests, it is proved that the system works properly.

We chose to develop a desktop application because it is easier to implement, especially when it comes to design part where we did not need to use HTML and CSS. It has the same functionality as a web application.

Being a desktop application, we consider that it is easier to use by any student who wants to apply to university courses, without having IT knowledge. This system can be configured for any faculty. In the same application, the Administrator part is also integrated. Establishing the user type is done using the account name.

4. REFERENCES

[1]. Liviu Negrescu, C # for beginners, vol 6: Visual Programming Environment, Ed. ALBASTRA, ISBN: 9789736502736

[2]. Constantin Gălăţan, Susana Gălăţan - Course C # - Programming in Visual C #

[3]. Academia Credis, Introducere in C#, http://www.academiacredis.ro/curs-introducere-in-C?gclid=CNf4gpm66c0CFfEV0wod9IALcQ-Introduction%20to%20C#, Accessed: 02.03.2016

[4]. Wikipedia, Microsoft Visual Studio, https://en.wikipedia.org/wiki/Microsoft_Visual_Studio, Accessed: 15.03.2016