

## APPLICATION FOR TRACKING THE WORKLOAD OF LECTURERS AND STUDENTS AT UNIVERSITIES

**Mariana Trifonova, Rossita Nesheva, Milen Kiryakov**

University of Mining and Geology "St Ivan Rilski", 1700 Sofia; trifonova.m@mgu.bg

**ABSTRACT.** This article presents modules developed for tracking the student workload, forming groups for tutorials and streams for lectures, as well as accounting the workload of lecturers from the University of Mining and Geology. These modules have been integrated in a comprehensive information system developed by a team of students and lecturers from the Department of Informatics. The modules provide a wide range of inquiry options for the different user categories. The focus lies on the interface and the interaction with the user and the options for mistake finding in the process of filling out the database.

**Keywords:** information system, database, object-oriented programming

### ПРИЛОЖЕНИЕ ЗА ПРОСЛЕДЯВАНЕ НА УЧЕБНАТА ЗАЕТОСТ НА ПРЕПОДАВАТЕЛИ И СТУДЕНТИ ВЪВ ВУЗ

**Мариана Трифонова, Росита Нешева, Милен Киряков**

Минно-геоложки университет "Св. Иван Рилски", 1700 София

**РЕЗЮМЕ.** Настоящата статия представя разработените и интегрирани към обща информационна система модули за проследяване на учебната заетост на студенти, формирането на подгрупи за упражнения и потоци за лекции, както и отчитане на натоварването на преподавателите от Минно-геоложкия университет. Предвиден е богат набор от справки за отделните категории потребители на системата, която се разработва от екип студенти и преподаватели от катедра „Информатика“. Акцентира се върху интерфейса и диалога с потребителя и възможностите за откриване на грешки в процеса на попълване на базата данни.

**Ключови думи:** информационна система, бази данни, обектно-ориентирано програмиране

### Introduction

Most universities in Bulgaria have integrated and are using an information system (IS) for tracking and reporting of educational workload. Such a system has not been introduced to the work of UMG yet. As a first attempt in this direction an IS has been developed in 2018 as a part of the thesis of Milen Kiryakov. In it, modules have been introduced, whose main purpose is the automated generation of reports about planned and realised classes of UMG lecturers according to the regulations of the University. The system was developed in Visual Studio .NET in the language C#. The database was built up using MS Access and contains 10 tables. In the current year the work in this direction has been continued with two further master theses of CTE students under the guidance of Assoc. Prof. Yanev and Assoc. Prof. Trifonova. Thus, the first version of the IS has been developed in two directions:

1. Migrating of the database to MySQL Server. This has a number of advantages. As Yanev (2013) points out, the server provides the functions for creating and processing the database, keeps its comprehensiveness, offers back-ups of the database, and more. Furthermore, it provides restructuring the data from a relational to a hierarchical format – e.g. XML (Yanev, 2011).

2. Extending the functionality of the system. This is the topic of the current article.

### Database Structure

The database of the application contains 20 tables (Fig. 1). It is in the third normal form. Some of the tables, in which the main data is contained, are as follows:

- Lecturers – first and last name, e-mail address, academic title, contract type, department;
- Course – code, full and abbreviated name according to the curriculum, timetable of lectures and tutorials, pursued degree;
- Stream – class type (lectures/tutorials), education form (regular/distance), automatically generated field for naming the stream showing the subject and the course of studies forming the stream, as well as an external key to the subject;
- Curriculum – subject number, course of studies number, semester number, number of weeks, course type, examination form, credit points.

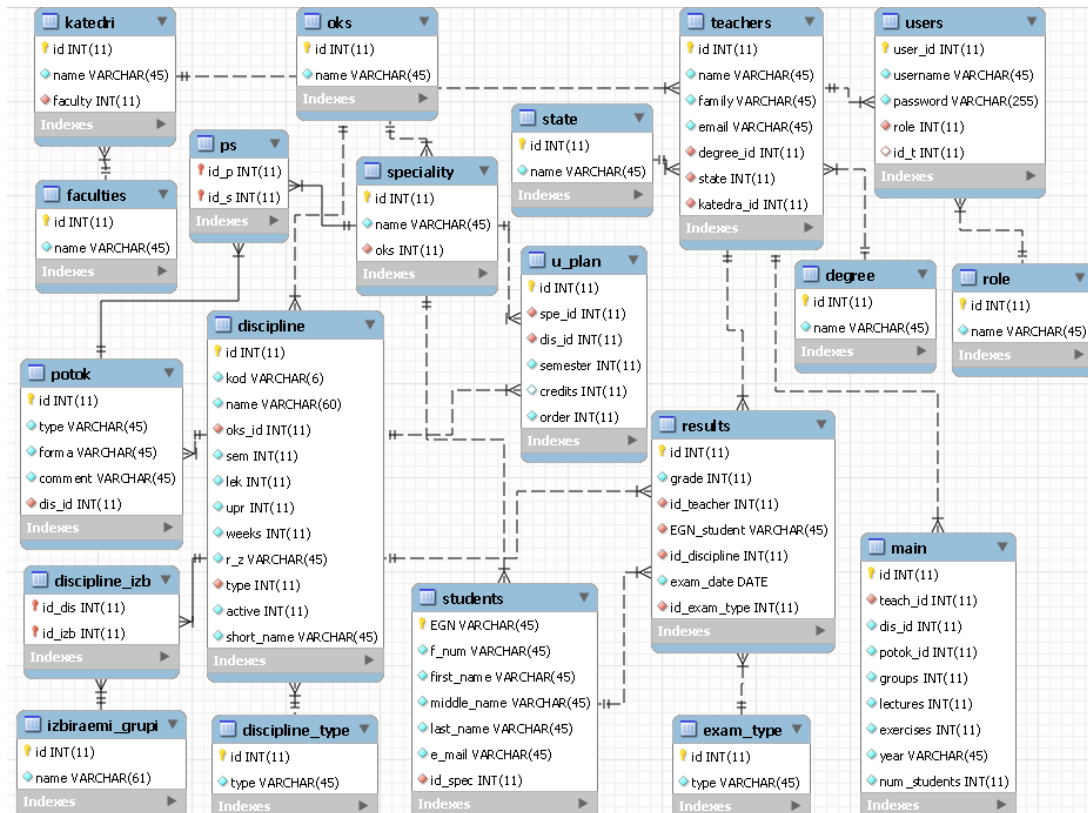


Fig. 1. Scheme of database

### Potential of C# for database processing

To access database .NET Framework uses ADO.NET - set of libraries for database processing. ADO.NET is ADO-ActiveX Data Objects successor and offers a programming model for data processing. This model corresponds to both models for access to the data - linked and not linked (Nakov et al., 2005).

Main data providers in ADO.NET are as follows:

- **SqlClient** – meant for access to MS SQL Server (version 7.0 or later). This data provider is optimised for work with SQL Server;
- **OleDb** – this provider is applied for access to databases maintaining OLE DB standard. Most of the known servers aimed at databases processing (for instance, Oracle, DB2, SQL Server, MySQL, Interbase, PostgreSQL and others) have OleDb drivers and can be used through a OleDb provider;
- **Odbc** – is applied for access to databases corresponding to the ODBC standard. All better known database servers are maintained. OLE DB and ODBC are in general competing standards but OLE DB is the more contemporary one and should be preferred to ODBC because it provides more flexibility and has better maintenance.
- **Oracle** – aimed at access to Oracle sources of data.

Besides the standard data providers that are incorporated in ADO.NET, there are also providers for direct link with other RDBMS that are offered by third developers. Such exist for IBM DB2, MySQL, PostgreSQL, Borland Interbase and others.

As Nakov recommends (Nakov et al., 2005) for accessing databases that are not supported according to the standard by .NET Framework, it is desirable to use the .NET Data

Providers, which is specific for them, instead of OLE DB or ODBC as far as this reduces the productivity and limits the accessible functionality that is specific for the relevant database. That is why, for the purpose of the developed system the driver MySQL Connector/.NET has been installed in advance. This driver allows .NET applications to work with MySQL.

MySQL Connector/.NET comprises several classes that are used to connect to the database, execute queries and statements, and manage query results. The following are the major classes of Connector/.NET (*MySQL Connector*, 2019):

- **SqlConnection**: Represents an open connection to a MySQL database.
- **SqlConnectionStringBuilder**: Aids in the creation of a connection string by exposing the connection options as properties.
- **SqlCommand**: Represents an SQL statement to execute against a MySQL database.
- **SqlCommandBuilder**: Automatically generates single-table commands used to reconcile changes made to a DataSet with the associated MySQL database.
- **MySqlDataAdapter**: Represents a set of data commands and a database connection that are used to fill a data set and update a MySQL database.
- **MySqlDataReader**: Provides a means of reading a forward-only stream of rows from a MySQL database.
- **MySqlException**: The exception that is thrown when MySQL returns an error.
- **MySqlHelper**: Helper class that makes it easier to work with the provider.
- **MySqlConnection**: Represents an SQL transaction to be made in a MySQL database.

### Application interface

On Figure 2 a generalised scheme of application interface is shown

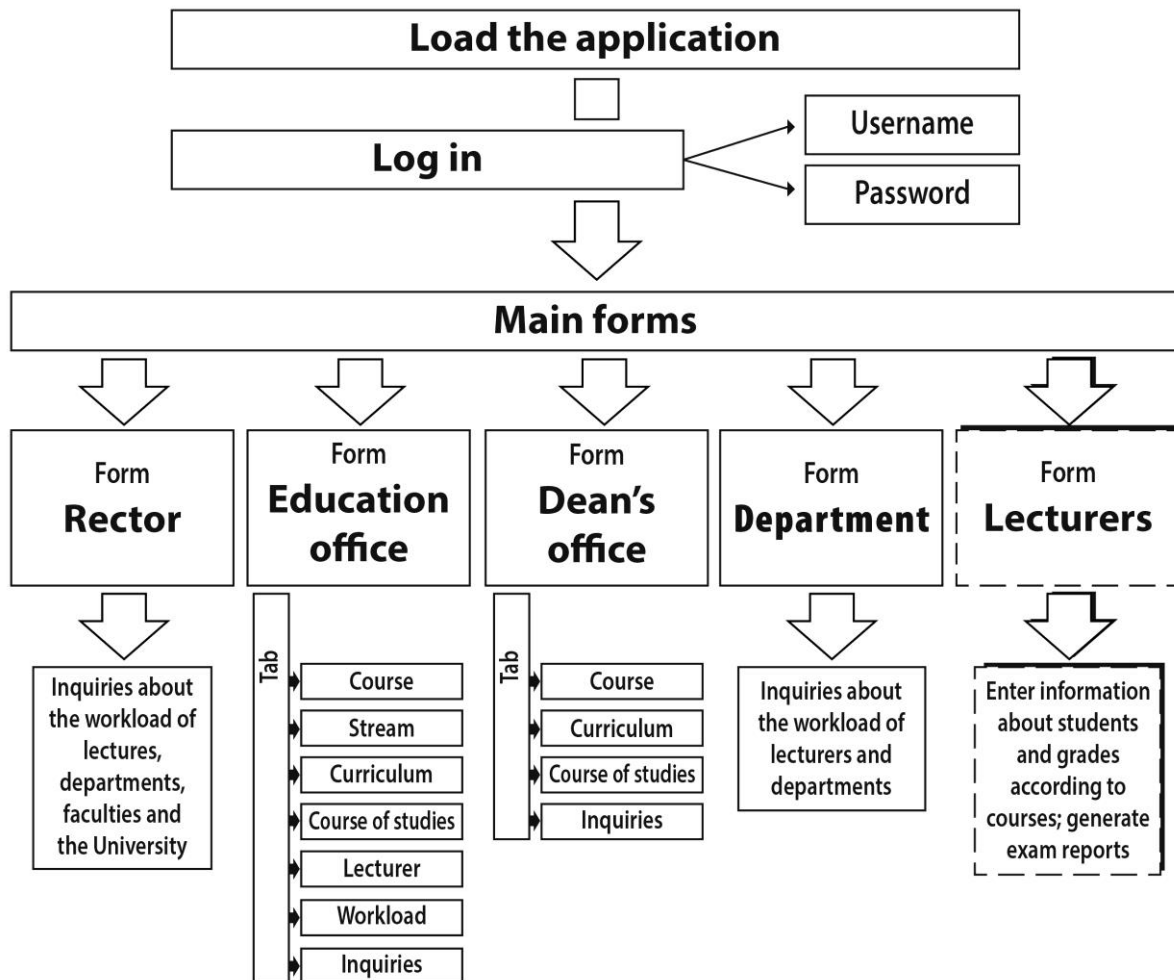


Fig. 2. Generalised scheme of the application

At the system start, the user must type in valid name and password. Currently, besides a system administrator the following five categories of users ("roles") are envisaged: rector, education office, Dean's office (in this case 3: the Faculty of Mining Technology (MTF), the Faculty of Mining Electromechanics (MEMF) and the Faculty of Geology and Exploration (GPF), lecturer (each of the lecturers in the University who will rate the students on a given subject), department (each one of the departments in the University).

Upon the successful login in the system, depending on the user's category, access to a given part of the database and given functions of IS is allowed. Each user can change his password.

#### User "Education Office"

This user is allowed to enter and edit as follows:

- Course. For each course, its code, full and abbreviated name (the abbreviated name is applied at streams forming), number of classes for lectures and exercises and pursued degree must be typed in. The course's code must not be used as an identifier because a course may be found in a number of various courses of studies under the same name and code but with different workload. That is

why, in the table of the courses a course (for instance, Introduction to the Computer Technologies) can be found six times with six different workloads but under the same code. On Figure 3 the dialogue for introducing of courses is shown. Tabular information in the main part of the window shows the entered courses into the database up to the moment. The user (in this case Education Office, but it can be some of the Dean's offices) can introduce and edit a new course using the controls in the right part of the form.

- Courses of studies.
- Curricula for courses of studies. The system can be used even if more than one curriculum for a course of studies is active at a moment. If so, the curriculum should be entered for the concrete academic year only. For example, the course of studies CTE, Bachelor's degree is typed in 4 curricula – for the first, second, third and fourth year, respectively. Thus, the system supports at a moment only one curriculum for each course of studies and course and it's the one in power at that moment. This makes easy for the changes in the curricula to be made and it is possible all four courses of a given course of studies to be taught according to different curricula.

Код	Наименование	ОКС	Л	у	Съкратено име
371102	Въведение в компютърните технологии	Бакалавър	2	3	ВКТ
371104	Основи на програмирането	Бакалавър	2	3	ОП
371152	Теоретични основи на информатиката	Бакалавър	2	2	ТОИ
371106	Обектно ориентирано програмиране	Бакалавър	2	2	ООП
371153	Компютърни мрежи и комуникации I	Бакалавър	2	2	КМК1
371109	Визуално програмиране	Бакалавър	2	4	ВП
371164	Основи на C#	Бакалавър	2	3	C#
371154	Компютърни мрежи и комуникации II	Бакалавър	2	2	КМК2
371110	Операционни системи	Бакалавър	2	2	ОС
371111	Системи за управление на бази данни	Бакалавър	2	4	СУБД
371155	Web дизайн	Бакалавър	1	3	WEB
371112	Разработване на приложен софтуер	Бакалавър	1	2	РПС
371108	Структури от данни и алгоритми	Бакалавър	2	3	СПА

Fig. 3. Form for introduction, editing and deleting of courses in the database

- Lecturers;
- Classes streams. "Stream" means a group of students (it can be a part of the students of a course of studies, but it can be, as well, students of a number of different courses of studies) who attend classes (lectures or exercises) in a group. Notion "stream" is used not only for lectures but also for seminars. The reason is that sometimes several courses of studies are combined in a subgroup (for example, for the elective courses). The user chooses the course, fixes the classes type (lectures/seminars), the form of education (regular/distance) and marks the courses of studies forming the stream. To make the dialogue for defining the workload easier, the system generates automatically a short name for the stream on the basis of the abbreviated names of the course and the courses of studies included in it. On Figure 4 is shown the dialogue for introducing of streams. The tabular information in the main part of the window shows the streams that have been entered in the database up to the moment. Using the controls in the right part of the form the user can edit, delete or introduce a new stream.
- Workload - fixes the lecturer for each concrete stream. While the information about courses, courses of studies, lecturers, curricula and streams is relatively constant and is to be corrected fairly rarely, the workload (or the lecturer for a concrete stream) is to be typed in each and every semester and it corresponds to the agenda. According to the University requirements related to the information about the lectures and seminars led by the lecturers and on the base of the workload that have already been entered MS Excel tables can be generated.

At the Education Office's disposal is a set of information about the workload of a concrete lecturer, department, faculty or the University as whole.

#### User "Dean's Office"

As a matter of fact there is no user "Dean's Office" in the system generally speaking. There are concrete Dean's offices, in this case MEMF, MTF and GPF. This user's functions and rights are part of the Education Office's functions, namely introduction and editing of course, curriculum and course of studies as well as reports about the workload of a concrete lecturer, a department or a faculty as a whole. The main difference between the Dean's office and the Education office as users is that the concrete Dean's office has access to the information about courses, courses of studies and departments of its own faculty, while the Education Office processes this information for the whole university. A concrete Dean's office has no access to the information about the courses of studies and the departments of a different faculty. The introduction in the database of the Department of Foreign Languages and Sports as an user in the Dean's office category is forthcoming.

#### Users "Rector" and "Department"

These users are only allowed to access information. The rector can get information about a concrete lecturer's workload, a department, a faculty or the University as whole. "Workload" means only the teaching activity of the lecturers. The user "Department" is allowed to get information about one or more lecturers in the respective department only. When completed, the module for tracking the teaching process and the students' grades is going to provide the users "Rector" and "Department" with information about the average grade of a given student, course of studies or a faculty.

№	Дисциплина	Специалност	Л/У	Р/З
1	Въведение в компютърните технологии	КТИД 1	Л	редовно
2	Въведение в компютърните технологии	КТИД 1	У	редовно
3	Основи на програмирането	КТИД 1	Л	редовно
4	Основи на програмирането	КТИД 1	У	редовно
5	Теоретични основи на информатиката	КТИД 1	Л	редовно
6	Теоретични основи на информатиката	КТИД 1	У	редовно
7	Обектно ориентирано програмиране	КТИД 1	Л	редовно
8	Обектно ориентирано програмиране	КТИД 1	У	редовно
9	Компютърни мрежи и комуникации I	КТИД 2	Л	редовно
10	Компютърни мрежи и комуникации I	КТИД 2	У	редовно
11	Визуално програмиране	КТИД 2	Л	редовно
12	Визуално програмиране	КТИД 2	У	редовно
17	Основи на C#	КТИД 2	Л	редовно

Fig. 4. Form for introduction, editing and deleting of streams in the database

#### User "Lecturer"

Currently, this module is under construction. Its purpose is to allow lecturers responsible for the grading to insert course marks. Additionally, the module tracks each student's performance. Based on the average grades, this module is going to provide information about different kinds of suitable scholarships. Further functions of this module are the introducing of new students or deleting students. The programme also generates exam reports and automatically moves students to the next course year at the end of each academic year.

#### Conclusion

Governing the teaching process in a university has a lot of individual features, specific for a given university. Therefore, it is difficult to develop a universal information system which can be applied at all universities. Thus, such a system created specifically for the needs of a specific university is necessary. Thus, reflecting the dynamics of the teaching process is made easier. The IS introduced in this article is the first attempt at creating such a system for the needs of UMG "St. Ivan Rilski".

Currently, around 5% of the database of the system is filled in with actual data. After completion of the database with real information about courses, lecturers and curricula, the maintenance of the system is going to be much easier, and the system itself is going to simplify the work of the respective experts.

#### References

- MySQL Connector: Chapter 5 Connector/NET Programming; <https://dev.mysql.com/doc/connector-net/en/connector-net-programming.html>.
- Nakov, S. et al. 2005, *Programirane za NET Framework*, 1. Faber, Sofia, 801 p. (in Bulgarian).
- Yanev, N. 2011. Opportunities for implementation of modern XML based technologies in the mining industry. – *Annual of the University of Mining and Geology "St. Ivan Rilski"*, 54, 4, 127–131 (in Bulgarian with English abstract).
- Yanev, N. 2013. *Metodologii i tehnologii za razrabotvane na informatsionni sistemi*. Izd. kushta "Sv. Ivan Rilski", Sofia, 148 p. (in Bulgarian).