SERVICE ORIENTED APPLICATION FOR RENDERING REPORTS ON THE TRAINING WORKLOAD AT THE UNIVERSITY OF MINING AND GEOLOGY “ST. IVAN RILSKI”

Nikolay Yanev, Lubomir Meshekov

University of Mining and Geology “St. Ivan Rilski”, 1700 Sofia; nikolay.yanev@mgu.bg, lubomir.meshekov@mgu.bg

ABSTRACT. The aim of this article is to present the work of a team from the Department of Informatics on developing and enhancing an information system for rendering reports on the training workload of the academic staff at the University of Mining and Geology “St. Ivan Rilski”. The focus is on the evolution of the application from desktop to server-oriented and on the user account management.

Keywords: information system, database, object-oriented programming

СЪРВЪРНО ОРИЕНТИРАНО ПРИЛОЖЕНИЕ ЗА ОТЧИТАНЕ НА УЧЕБНАТА ЗАЕТОСТ В МГУ “СВ. ИВАН РИЛСКИ“

Николай Янев, Любомир Мешеков
Минно-геоложки университет “Св. Иван Рилски“, 1700 София

РЕЗЮМЕ. Целта на настоящата статия е да представи работата на екип от катедра "Информатика" по разработване и усъвършенстване на информационна система за отчитане на учебната заетост в МГУ "Св. Иван Рилски". Акцентира се на еволюцията на приложението от десктоп към сървърно ориентирано и управлението на потребителските акаунти.

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

Introduction

2018 saw the formation of a team at the Department of Informatics whose task was the development of an information system (IS) for rendering reports on the training workload of the academic staff. The idea was to develop the IS within a series of diploma theses of prominent students taking a Master’s degree in the course of study in *Computer Technologies in Engineering* under the supervision of Assoc. Prof. Trifonova and Assoc. Prof. Yanev. The first step in the implementation of such application was presented in Eng. Milen Kiryakov’s diploma thesis which was successfully defended in the summer of 2018. The aim was to develop an IS for the automatic generation of the “Planning/Report” template form used at the University of Mining and Geology (UMG) “St. Ivan Rilski”. The choice of software products was MS Access for the implementation of the database and C# as a means of developing the user interface.

Having analysed the results obtained, a decision was taken to extend the scope of the system which implied the generation of various types (roles) of system users, respectively the implementation of a Log-in system.

To achieve the new goals, MS Access as the database management tool needed to be replaced by a server-oriented database management tool. The main reason for this was the limited abilities of MS Access for multiuser access (Deliyska et al., 2017).

Our team’s choice was the MySQL database server and was prompted by a number of reasons, the most important of which were:

- The popularity of MySQL. According to DB-Engines Ranking (2019), MySQL ranks second in popularity among modern database management systems and first among those that are distributed for free;
- High security level. The protection and privacy of personal data is the main focus of the work of all institutions dealing with such data, due to the implementation of Regulation 2016/679 of the European Parliament and of the Council of Europe. MySQL largely incorporates the good practices in this area described by Toncheva-Pencheva et al. (2018);
- Excellent response time and performance;
- The significant experience in working with this system gained by the members of the department.

Data migration from MS Access to MySQL

Capabilities of the MS Visual Studio for handling databases

Two data access models are used in building information systems: connected and disconnected. The connected model has a permanent connection to the database, and in the case of the disconnected model the database connection it is not permanent: it is carried out briefly and the work with the data is performed offline by retrieving the data which are downloaded and stored on the local machine. Due to its indisputable advantages (better performance, more precise control of competitive access in the multiuser mode of operation, working with the current version of data, etc.), the connected model is preferred when working with relational databases. The .NET Framework uses ADO.NET to access databases.

ADO.NET is a collection of data libraries included in the .NET Framework. These data can be relational, hierarchical (e.g. XML), etc. Libraries include classes, interfaces, structures, and other types and are designed to access various data sources. ADO.NET is entirely based on the .NET Framework and has many of its features: multi-language support, automatic memory management, object-oriented design, a common type system and naming convention. Funds are provided that make it possible to work with the data regardless of the source (Nakov, 2006).

ADO.NET offers a software model for handling data that goes with both data access models - the connected and the disconnected one. In addition, the object model of ADO.NET offers very precise control over the source connection, the command execution, and the data processing. In ADO.NET, a clear distinction is made between data access and data manipulation (ibid.).

The various classes and interfaces provided by ADO.NET are divided into several main namespaces:

- System.Data - this contains the main architectural classes of ADO.NET. It includes, for example, the DataSet, DataTable, and DataRow classes;
- System.Data.Common - this namespace contains classes that are used independently of data sources, such as DataAdapter;
- System.Data.SqlClient - this includes specific SQL Server connection classes that allow to connect to MS SQL Server, to retrieve data, and to execute commands. Some of the classes in this namespace are SqlConnection, SqlCommand, SqlDataReader, etc.;
- System.Data.SqlTypes - this comprises classes that match the types of data embedded in the SQL Server and are a faster and more secure alternative to the other types. It includes SqlInt32, SqlDouble, SqlDateTime, and others ;
- System.Data.OleDb - this provides classes for connecting to an OleDB data source. It includes, for example, the classes of OleDbConnection, OleDbCommand, OleDbDataReader, etc.;
- System.Data.Odbc − these are ODBC connection classes. For example, OdbcConnection, OdbcCommand, etc. are contained;
- System.Xml − this namespace contains classes that support XML data processing and the relationship between the relational model and XML. For example, the XmlDocument and XmlDataDocument classes are often used.

The so-called data providers are used to access the various databases. They are specific to the particular database, but adhere to the programming model of ADO.NET by implementing the interfaces defined in it.

Fig. 1 shows the main components of ADO.NET (ibid.).

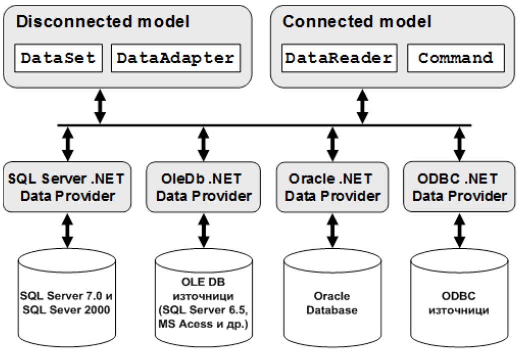


Fig. 1. Components of ADO.NET

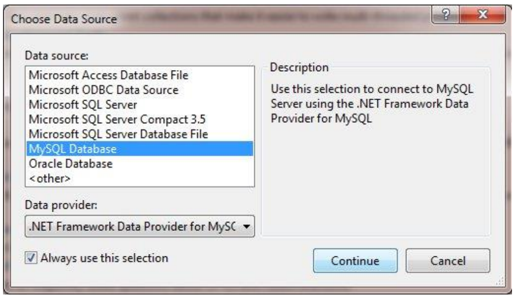
Integrating MySQL to MS VisualStudio

MS VisualStudio does not have an integrated MySQL support; therefore, the installation of the appropriate driver was necessary (MS VisualStudio Data Providers, 2019). The natural choice for such a driver was MySQL Connector/NET, the one officially offered by MySQL.

MySQL Connector/NET is a driver supporting the ADO.NET interfaces. These are required when working with .NET applications that access MySQL (Dubois, 2006).

MySQL Connector / NET allows .NET applications to use MySQL. It is not based on the C client library. It is written in C# and directly implements the client-server communication protocol. Connections can be made via TCP/IP, Unix socket files, named channels, or shared memory (MySQL for Visual Studio, 2019).

Microsoft has provided a convenient graphical method for adding the MySQL library to VisualStudio. In its essence, this is adding References after the successful installation of the driver and after applying it from the Data Connection option (Fig. 2)



Data source:
Microsoft Access Database File
Microsoft ODBC Data Source
Microsoft SQL Server
Microsoft SQL Server Compact 3.5
Microsoft SQL Server Database File
MySQL Database
Oracle Database
<other>

Description
Use this selection to connect to MySQL Server using the .NET Framework Data Provider for MySQL

Data provider:
.NET Framework Data Provider for MySC

☑ Always use this selection

Fig. 2. Choice of database type from a C# medium

The C# classes that are most commonly used for communicating with the MySQL database are the following (MySQL Connector/NET, 2019):

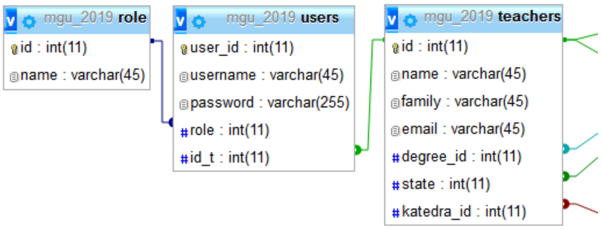
- MySqlConnection: Represents an open connection to a MySQL database;
- MySqlConnectionStringBuilder: Aids in the creation of a connection string by exposing the connection options as properties;
- MySqlCommand: Represents an SQL statement to execute against a MySQL database;
- MySqlCommandBuilder: Automatically generates single-table commands used to reconcile changes made to a DataSet object 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: A helper class that makes it easier to work with the provider;
- MySqlTransaction: Represents an SQL transaction to be made in a MySQL database;
- MySQLMembershipProvider: Manages storage of membership information for an ASP.NET application in a MySQL database;
- MySQLRoleProvider: Manages storage of role membership information for an ASP.NET application in a MySQL database;
- MySqlEFConfiguration: Adds the dependency resolvers for MySQL classes;
- MySqlExecutionStrategy: Enables automatic recovery from transient connection failures.

Data transmission

The existing database required significant changes, both in extending the existing tables and in adding ten new tables (Trifonova, 2019). Therefore, our team decided to rebuild the database in the MySQL environment.

In view of user management, two new tables were added - Users and Role (Fig. 3).



mgu_2019 role
id : int(11)
name : varchar(45)

mgu_2019 users
user_id : int(11)
username : varchar(45)
password : varchar(255)
role : int(11)
id_t : int(11)

mgu_2019 teachers
id : int(11)
name : varchar(45)
family : varchar(45)
email : varchar(45)
degree_id : int(11)
state : int(11)
katedra_id : int(11)

Fig. 3. The “Users” table and the relations associated with it

The “users” table contains information about the users of the system. MySQL function SHA2 (str, hash_length) is used to store the password. The SHA-2 family of hash functions (SHA-224, SHA-256, SHA-384, and SHA-512) is calculated. The first argument is the plaintext string to be hashed. The second argument indicates the desired bit length of the result, which must have a value of 224, 256, 384, 512, or 0 (which is equivalent to 256). If either argument is NULL or the hash length is not one of the permitted values, the return value is NULL. Otherwise, the function result is a hash value containing the desired number of bits. SHA2() can be considered cryptographically more secure than MD5() or SHA1() (MySQL 5.7 Reference Manual, 2019).

The “Role” table describes the role of the user. At this stage, eight roles are envisaged:

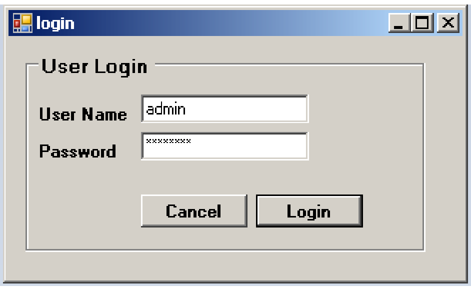
- admin - this user has access to the management of the accounts within the system and of the role definition;
- teacher - the main function of this role is student assessment;
- rector - this role allows the user to provide references to all the available data in the system;
- d_MEMF, d_MTF and d_GPF – reports on the workload of lecturers and departments, as well as the management of curricula, but only of those of the respective faculty: the Faculty of Mining Electromechanics (or MEMF, as is the abbreviation in Bulgarian) the Faculty of Mining Technology (or MTF), or the Faculty of Geology and Exploration (or GFP);
- UO – this role allows the management of the training workload of each lecturer;
- Secretary (or sekretar, if the Bulgarian equivalent is used) – departmental level information provided by secretaries.

Upon successful login, a form opens that corresponds to the role of the respective user. In addition, any user who has successfully connected to the system can change their password.

Development of a graphical interface

Log-in system

The user authentication to the information system is accomplished by entering a user name and password (Fig. 4). Data from the two text controls (User name and Password) are validated in order to avoid database compromise via SQL injection.



User Login
User Name admin
Password ********

Fig. 4. Log-in form

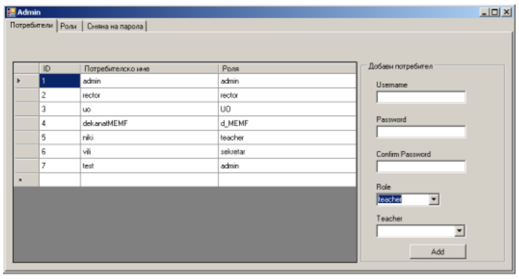
The user account is employed for the successful login to the system, and then a new form is opened containing data to which the user has access.

User data are stored and accessible until the user logs out.

Admin panel

After the “admin” user has successfully logged in the system, a user management form opens. It includes three sub-sections (tabs):

- Users (Fig. 5) – users are added, removed and edited from this tab. The parameters are the name, the password, and the role. If the role is “Teacher”, an additional control opens whose purpose is to allow the selection of a lecturer who is already in the “Teachers” table but who has not been tied to a specific role yet;

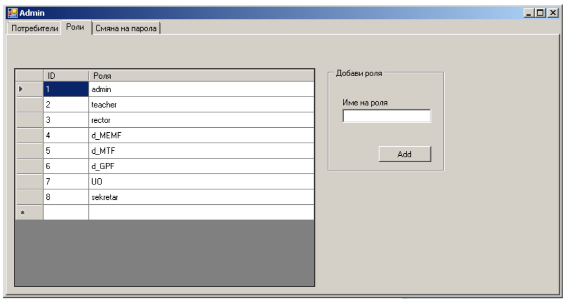


ID	Потребителско име	Роля
1	admin	admin
2	rector	rector
3	uo	UO
4	dekanatMEMF	d_MEMF
5	niki	teacher
6	vili	sekretar
7	test	admin

Role: teacher

Fig. 5. The "Users" tab

- Roles (Fig. 6) - a new role is added from this tab;



ID	Роля
1	admin
2	teacher
3	rector
4	d_MEMF
5	d_MTF
6	d_GPF
7	UO
8	sekretar

Fig. 6. The "Roles" tab

- Password change (Fig. 7) - the change of password is performed by entering the current password and repeating the new password twice. Such a tab is visible to every user who is successfully connected to the system (Trifonova, 2019).

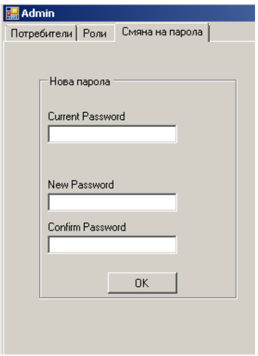


Fig. 7. The "Password Change" tab

Conclusion

The management of the training process is an elaborate task that requires the expertise and capabilities of all employees who are entrusted with its implementation. Building an information system that digitizes, manages, and archives this process would greatly facilitate the work of the respective experts, and would also enable people with lower qualifications to successfully work with the system.

To date, no such system exists at the University of Mining and Geology "St. Ivan Rilski". The information system developed by a team from the Department of Informatics at UMG has been a decisive step towards the solution to this problem.

The transfer of the database to a server-oriented application, like MySQL, has allowed us to take advantage of such benefits of the client-server technology as the centralised data storage, the multi-user access, a higher level of logical and physical independence of the data, etc.

References

DB-Engines Ranking. *https://db-engines.com/ en/ranking*.

Deliyska, D., I. Hristova. 2016. Methodological aspects of training in databases. − *Annual of the University of Mining and Geology "St. Ivan Rilski", 59*, 4, 117−121.

Dubois, P., S. Hintz, K. Pederson. 2006. *MySQL 5.0 Official Certification Guide.* Soft Press.

MS Visual Studio Data Providers. *https://docs.microsoft.com/en-us/dotnet/framework/data/ adonet/data-providers*.

MySQL Connector/NET and the X DevAPI. *https://dev.mysql.com/doc/dev/connector-net/8.0/html/ connector-net-x-devapi-reference.htm*.

MySQL for Visual Studio. https://www. mysql.com/why-mysql/windows/visualstudio.

MySQL 5.7 Reference Manual. https://dev. mysql.com/doc/refman/5.7/en/encryption-functions.html# function_sha2.

Nakov, S. 2006. *Programming of the .NET Framework.* Bulgarian Association of Software Developers.

Toncheva-Pencheva, S., Y. Anastasova. 2018. Protection of personal data in E-learning − methodology and technologies. − *Journal of Mining and Geological Sciences, 61*, 4, 51−53.

Trifonova M., R. Nesheva, M. Kiriakov. 2019. Application for tracking the workload of lecturers and students at universities. − *Journal of Mining and Geological Sciences, 62*.