Design and Implementation of College Student Information Management System Based on Web Services

TANG Yu-fang, ZHANG Yong-sheng
School of Information Science & Engineering, Shandong Normal University, Jinan, 250014, China
tangyufang2322@126.com; zys@sanlian.com.cn

Abstract
As an example of SIMS (Student Information Management System) developed independently by School of Information Science & Engineering of Shandong Normal University, this paper introduces database design, specific realization of each function module and key technologies used in the system. In the .NET environment, using ASP.NET technology, Visual C# and JavaScript as programming language, this system accesses the database of Microsoft SQL Server 2005 with ADO.NET technology, and could be employed by users with high security following the access control mechanism of RBAC (Role-Based Access Control) on Web Services.

1. Introduction
With the continuous growth of school time, the expanding of school scale and the sharp increasing amount of fresh and previous students in each college and university, the relevant information of students has doubled and redoubled. For each of the colleges and universities, faced with such a large amount of information, in order to accurately and efficiently carry out the students management on the basis of a full grasp of information, the extent of difficult task is unimaginable. Through the SIMS, we can realize the management of students information network-based digitally, track and collect the students information real-timely, accurately, reduce the workload of students information management, promote the scientific and effectiveness of students information management, thus to promote the management of college students to carry out high efficiently.

2. Key technology

2.1. .NET Framework
.NET Framework is the core of .NET, composed by the common language runtime libraries and class libraries, it provides applications support of the intermediate code, and manages the intermediate code at run-time, thus lots of programs written by computer languages can run cross-platform. The most unique attribute of .NET is its multi-language support.

2.2. ADO.NET
.NET platform is a common framework provided by Microsoft, the framework class library, which will step over all of the Windows API functions, in particular, it contains a number of libraries used frequently, and users will find that XML and ADO Object Model are integrated in a tree-like class collection, which is called ADO.NET. ADO.NET is a development approach defined for .NET database application and a set of design environments for database.

2.3. RBAC
RBAC (Role-Based Access Control) introduces the concept of Role in order to isolate User (that is the main action, Subject) from Privilege (an operation on the Object, that is Operation + Object), so that all authorization should be given to the Role rather than directly to the User. Two main characteristics of RBAC are as follows: 1) Because the relationship between the roles and the permissions changes much slower than it between the roles and the users relatively, the complexity of authorization management and
administrative overhead are reduced. 2) The corporate security policies are supported flexibly, and a huge flexibility is given to the changes in the enterprise. Figure 1 demonstrates the relationship among the Users, Roles and Privileges in the RBAC model.

Figure 1. The relationship between Users, Roles and Privileges in the model RBAC

3. System running environment

The student information management system introduced by this paper can run in the following environment.

Server-side:
Operating System: Windows Server 2003
Database: Microsoft SQL Server 2005
Script processing: IIS5.1/6.0

Client-side:
Operating System: Windows 2000/XP/Vista
Browser: IE6.0/7.0/8.0 and other mainstream browser like Maxthon2.0

4. System design

4.1. System function analysis

The system is mainly composed of seven function modules such as the students basic information management module, the score management module, the award management module, the punishment management module, the social practice management module, the subsidization management module and the user privilege management module, through the B/S mode (such as Web browser), users can visit the system. There are two types of users: some are the advanced users (such as faculty counselor) with higher level of operating privilege(such as modification, deletion and other operations), so that they can operate the sensitive data with the high performance of security; others are the general users (such as students) with smaller privilege to operate the system relatively, these users can record their own information (such as contact information of their parents) into the system, and log on to the system at any time to look up personal information (such as examination results).

4.2. Database design

The system builds up a database SIMS.mdf with SQL Server 2005 to store data. In this database, it includes data tables as follows:

1) Basic information table of student: It stores mainly basic information of students, with student ID as a primary key.
2) Information table of students’ scores: It stores mainly the course scores of the first and second semester each academic year, with student ID as a foreign key to associate with the basic information table of student.
3) Information table of obtaining reward: It stores mainly reward of intramural and outside school each academic year, with student ID as a foreign key to associate with the basic information table of student.
4) Information of submitting punishment: It stores mainly punishment of each year, with student ID as a foreign key to associate with the basic information table of student.
5) Information table of receiving assistantships: It stores mainly assistantships of each year, with the student ID as a foreign key to associate with the basic information table of student.
6) Information table of social practice: It stores mainly social practising of each year, with student ID as a foreign key to associate with the basic information table of student.

Figure 2 demonstrates association relations among above six data tables.

Figure 2. Association relations between database 6 data tables in database SIMS

The data tables in database SIMS related to user privilege management are as follows:

1) The user table: It stores mainly personal information of users, such as user number, user name, user password, and the role user-owned, with user number as the primary key, and User_Role as the foreign key.
to associate with the role table.

2) The role table: It stores mainly role information in the system, such as common visitors (students), system administrator (faculty instructor) and system operation privilege assigned, with role number as the primary key, and Role_Privilege as the foreign key to associate with the permission table.

3) Privilege table: It stores mainly operation privilege assigned by system, such as operations on adding, viewing, modifying and deleting personal information, and specific operation privilege and mapping relations between corresponding functional modules, with permission number as the primary key to associate with the role table.

Figure 3 demonstrates association relation among above three data tables.

Figure 3. Association relation between the user table, the role table and permission table related to user privilege management

4.3. Specific realization of each function module

After the SIMS. mdb database created, using of ASP.NET technology, especially the Visual C # Language, it is to make system various functional modules development.

4.3.1. The user login page Login. aspx. This page achieves the Register and Login Authentication functions. With user login authentication as an example, during the login, if both account and password are non-null values, the procedures query the user table to verify the existence of the user account (UserName). If there are no values, the procedures prompt the user to enter a registered account or sign up for a new account; if there is, the procedures determine whether the password user inputted is the same as the password corresponding to the same user account. If it is, the procedures allow users to access the system; otherwise the procedures prompt the user "password error".

Figure 4 demonstrates the user login page Login. aspx.

4.3.2. Six function modules. Through the corresponding pages, the students basic information management module, the score management module, the award management module, the punishment management module, the social practice management module and the subsidization management module can run SQL command on the corresponding data table in the SIMS.Mdf to achieve the operations on the student information (such as names, achievements, rewarded, etc.), such as inserting, query, modification, deletion and other operations. For example, using the system, we can query the school awards information what a given student (StudentNo is 2007021087) gained in the first school year.

4.3.3. The user privilege management module. The module based on the RBAC achieves the authorized management and authentication privilege functions. Authorized management process is divided into two parts: first, to realize the access privileges associated with the roles, and then to realize the roles associated with the users in order to separate the users from the access privileges. The task of the authentication privilege is to verify whether the user own certain privileges to operate certain resources of the system(such as to browse the examination results of all students). Figure 5 demonstrates the data flow of the user privilege management module.

Figure 5. The Data flow of the user privilege management module
The codes of assigning user operation on the system resources are as following:

```csharp
public static RoleActions GetInstance()
{
    if (RoleActions.instance == null)
        RoleActions.instance = new RoleActions();
    return RoleActions.instance;
}

public Object DispatchAndCheck(string UserNo, Func. Actions action, Object Parameter)
{
    string RoleNo = UserNoToRoleNo(UserNo);
    if (string.IsNullOrEmpty(RoleNo))
        return null;
    string[] Privileges = GetRolePrivileges(RoleNo);
    if (Privileges == null) return null;
    bool OkToDispatch = CheckPrivileges(action, Privileges);
    if (OkToDispatch)
    {
        return Func.Dispatch(action, Parameter);
    }
    else
    {
        return null;
    }
}

private bool CheckPrivileges(Func. Actions action, string[] Privileges)
{
    Dictionary<string, string> PrivilegesCache = new Dictionary<string, string>();
    foreach (PDataSet. PrivilegeTableRow PrivilegeRow in this. PrivilegeTable.Rows)
    {
        PrivilegesCache.Add(PrivilegeRow.PrivilegeNo, PrivilegeRow.Privilege_FUNCCLASS);
    }
    bool OkToDispatch = false;
    foreach (string priv in Privileges)
    {
        if (action.ToString().ToLower().Contains(PrivilegesCache[priv.ToLower()])))
        {
            OkToDispatch = true;
        }
    }
    return OkToDispatch;
}

private string[] GetRolePrivileges(string RoleNo)
{
    string[] Privileges = null;
    foreach (PDataSet. RoleTableRow RoleRow in this. RoleTable.Rows)
    {
        if (RoleRow.RoleNo == RoleNo)
        {
            string temp = RoleRow.Role_Privilege;
            Privileges = temp.Split(new char[] { '|' }, StringSplitOptions.RemoveEmptyEntries);
            break;
        }
    }
    return Privileges;
}

private string UserNoToRoleNo(string UserNo)
{
    string RoleNo = string.Empty;
    foreach (PDataSet. UserTableRow UserRow in this. UserTable.Rows)
    {
        if (UserRow.UserNo == UserNo)
        {
            RoleNo = UserRow.User_ROLE;
            break;
        }
    }
    return RoleNo;
}
```
result =
(Object)Delete(Parameter as string);
break;
case Actions.Insert:
    Object[ ] p =
(Object[ ])Parameter;
    result = (Object)Insert(p[0] as string, p[1] as Dictionary<string,string>);
    break;
default:
    break;
}
return result;

private static Dictionary<string,string>
Query(string UserNo)
{
    // code of the query operation
}
private static bool Delete(string UserNo)
{
    // code of the deletion operation
}
private static bool Insert(string UserNo,
Dictionary<string,string> parameters)
{
    // code of the insertion operation
}

The user privilege management module based on the
RBAC, has several advantages such as convenient
coding process, clear logical structure, good system
security, and strong expansibility.

5. Conclusion

The SIMS is developed by the design of database
SIMS.mdf and the realization of each module’s
function. Through the system, students can input
personal information into the database of college
student information management system real-timely, accurately, and inquire personal information timely
and easily. Thus, it reduces information input workload
of administrators, and enables them to focus on information auditing and consulting. It also realizes the
digitalization and networking of information
management for college student, and promotes the
management of college students to carry out high
efficiently.

6. References

[1] DUAN Hai-xin, DUAN Hai-xin, and WU Jian-ping,
"Design and Implementation of a Network Management
System Based on Web and Database", Journal of Software,

[2] CAI Chang-an, WANG Qi, "Design and implementation
of student information management system based on B/S
model", COMPUTER ENGINEERING AND DESIGN,
Beijing, 2006, 27(14), pp. 2585-2587.

[3] ZHA Yi-guo, XU Xiao-yan, and ZHANG Yu-sen,
"IMPLEMENTING RBAC ON THE WEB", JOURNAL OF
COMPUTER RESEARCH AND DEVELOPMENT, Beijing,

[4] LIU Jin-xiao, MA Su-xia, and QI Lin-hai, "Research and
implementation of access-rights control in web systems",
2550-2553.

[5] Chen Yu, Yang Shi-qing, "Key Techniques of Student
Information Management System Based on Internet &
Intranet", Journal of Wuhan University of
Technology(Transportation Science &Engineering), Wuhan,

[6] SA Shi-xuan, WANG Shan, Introduction to Database