An Online Course Registration System for the Faculty of Engineering in University of Peradeniya
Department of Computer Engineering, Faculty of Engineering, Peradeniya 20400 Sri Lanka.
Email: *roshanr@ce.pdn.ac.lk

Introduction
Registering for courses of the following semester in a faculty is an activity students undertake towards the end of every semester. The students at the Faculty of Engineering, like all other faculties in the University, manually fill their course registration forms by collecting them from the Assistant Registrar’s (AR) office. These forms are approved by the student advisers before they are returned to the AR’s office again for processing.

Human beings make mistakes: Therefore, there are several potential places where mistakes could be introduced into the data related to registration, starting from the students who fill these forms to the staff members at the AR’s office who enter the details from the forms into a database for processing. These mistakes bring a number of problems, such as clashes in the timetable that affect the smooth functioning of the Faculty [1] and difficulties imparted to the student community. In addition, everyone involved in the registration process are under pressure, as every aspect of course registration are checked manually (most of them are easy to perform automatically, such as pre-requisites of courses) and they have to be perfected for proper running of the system.

This motivated us to develop an electronic course registration system, where most of the aspects are verified automatically and therefore could reduce human mistakes. This paper presents our experience in developing, deploying and running an online course registration system at the Faculty of Engineering in Peradeniya. In this system, students register for their courses online, which are approved by their advisers and then processed automatically to provide a set of reports to the administrative staff at AR’s office, where the information is used. Most aspects of course registration, such as pre-requisites, the maximum number of courses a student can register per semester, etc. are checked automatically reducing the burden (time and pressure) of the advisers and the administrative staff.

Requirement analysis
Before introducing a solution to a problem, it is necessary to gather required information from relevant background. We discussed with the staff responsible for course registration at the AR’s office and collected the necessary information [2] and formalized the main requirements of the system as follows:

- Authentications and Authorizations of users;
- Administrators should be able to decide time period for the registration (before the start of the semester) and time period for the add/drop period (at the beginning of the semester);
- Administrators should be able to enter required data into the system such as courses, students, advisers and examination results;
- Advisers are allowed to view filled registration form of each student and accept/ reject the registration;
- Students should be able to view current courses and previous results, to register or add/drop new semester courses;
- Users should be able to change their passwords and personal information; and
- In the absence of a relevant adviser, the head of the department should be able to accept the online registration forms.

System analysis
Information should be thoroughly analyzed to get a clear understanding of them. Therefore, we analyzed the requirements gathered from relevant parties and decided to categorize the users who use this system into administrators, advisers and students.

All users have their own usernames and passwords to access the system and they have the ability to change their passwords. They will be given separate entry levels to access the system. Figure 1 depicts the use-case diagram of the system. Administrators are the staff officer at the AR Office who is responsible for course registration. They have the authority on deciding time durations, entering required details and finalizing registrations.
Advisors are all the department heads and lecturers who are assigned as advisers for students. They are capable of viewing courses, student details and results and accepting or rejecting registration forms. Student category contains everyone who has registered for a degree programme in the faculty. They are allowed to view available courses, their details and results, and to complete their registration forms and add/drop forms.

System architecture and technologies

The system is designed under the 3-tier architecture model. A 3-tier application is a program which is organized into three major disjunctive tiers and they are: presentation tier (frontend), logical tier (middleware), and data tier (backend). As depicted in Figure 2, each tier/layer can be deployed in geographically separated computers in a network. Some architects divide the logic tier into two sub tiers: business and data-access, in order to increase scalability and transparency. The characteristic of the tier communication is that the tiers will communicate only to their adjacent neighbours. For example, the presentation tier will interact directly with the business tier and not directly with data access tiers.

The advantages and deployment details of such architecture could be found in any software engineering textbook such as [3].

We mainly used Visual Studio 2003 as the development environment and SQL Server 2000 as the database design tool. Technologies which we used to do this project are shown in Table 1.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dream Viewer</td>
<td>GUI Design</td>
</tr>
<tr>
<td>CSS</td>
<td>Additional features in GUI design</td>
</tr>
<tr>
<td>ASP.net</td>
<td>Programming design</td>
</tr>
<tr>
<td>Ajax</td>
<td>Client Script Development</td>
</tr>
<tr>
<td>SQL Server 2000</td>
<td>Database Design (Create data tables and stored procedures)</td>
</tr>
<tr>
<td>IIS</td>
<td>Web server to host the system</td>
</tr>
<tr>
<td>Crystal Report 9.0/MS Excel 2003</td>
<td>Generate reports</td>
</tr>
</tbody>
</table>

Development environment and the procedure

Given the architecture and the technologies, Figure 3 depicts the environment adapted in developing the system. The following are the development steps which we followed to build the system (even though, some of them appear as separate steps, they are concurrent procedures):

- GUI design
- Create Data tables
- Create code behind class
- Create Ajax class

- Create Data Access class
- Create Stored Procedures
- Generate Reports

We generated a solution named as **Online Registration** in the development environment of **Visual Studio 2003** and three projects (as part of the solutions) as followings:
1. Online Registration – this is the start-up project of this system, will depend on the following two projects;
2. Data Access – this project will actuate the data transaction between the client and the server; and
3. Web Security – this project will configure the database and handle the authentication and authorization.

**Implementation**

The implementation of the system consists of the development of a number of web-forms, database entries and store procedures. Every web-form includes one or more data-tables (depicted in Figure 4 – the data diagrams describes the relationship between data tables and how they share data fields) to transact relevant data. According to the relationship between the web-forms, data-tables are created. Stored procedures [4] are used as they are the convenient way of handling SQL queries rather than developing queries within the programming code. Most of the details are omitted for brevity due to the page limit of this extended abstract.

**Testing**

Testing is a main part of the software life cycle. The **Online Course Registration System** was also well tested. The system was deployed on a server at the **AR’s office** and the students were asked to use this system to register for their following semester courses. During this period all the students who should register for their next semester courses used this system and they gave their comments/feedbacks. After they registered for courses, the registrations have to be accepted by their advisors. Therefore, the advisors used the application for accepting registrations. Their comments and feedbacks were taken into consideration as well. Later the administrator’s comments were also considered before a final improvement was made on the system.

**Conclusion**

This paper describes our experience in designing, developing and deploying an online course registration system at the Faculty of Engineering in University of Peradeniya. The system has not only reduced the burden of all parties involved in the course registration process, but also improved the process by reducing errors.

**References**

1. Homepage, Faculty of Engineering, University of Peradeniya, available at [http://www.pdn.ac.lk/eng](http://www.pdn.ac.lk/eng).