Ceylon Journal of Science: Physical Sciences, 15, 27-39 (2009)

CUBRAS: DESIGN AND IMPLEMENTATION OF THE COURSE UNIT BASED RESULTS ANALYSIS SYSTEM

S. R. KODITUWAKKU AND P. CHAMIKARA
Department of Statistics & computer science, University of Peradeniya

ABSTRACT

With the introduction of the course unit system in the state universities, students' evaluation process has been transformed into a highly complicated task. In order to evaluate students' results it is required to check several requirements such as number of credits, compulsory courses, prerequisites and minimum grade point average. Additionally this task needs to be completed quickly and efficiently. This paper presents an information system developed to achieve this task. Salient features of the system include evaluation of students' results to decide whether students are eligible for a particular degree, for special degree programmes and for honours. Additionally, the system facilitates the online registration of students and automatic transcript generation. The system developed was tested with randomly selected student results. The test results indicated that the system functions accurately and efficiently.

1. INTRODUCTION

With the introduction of the course unit system, students are provided with a lot of flexibility in selecting degree programmes. The Faculty of Science, University of Peradeniya offers 21 subject combinations. Each combination consists of three major subjects and each subject consists of several course units. Students who are admitted to the faculty enrol for a subject combination. Before commencing the regular academic programme, students have to enrol for three compulsory foundation courses. During the first year each student enrols for three subjects, a combination, and offers a minimum of 24 compulsory credits. During the second year students could enrol for three major subjects or two major subjects and offer a minimum of 27 credits. This includes both compulsory and optional courses. At the end of the second year students could apply for one or more special degree programmes. Those who have satisfied the requirements of such programmes are selected to follow special degree programmes. The others continue towards a general degree. The general and special degree students have to earn a minimum of 90 credits and 120 credits respectively to obtain their

Corresponding author: E mail salukak@pdn.ac.lk

degrees. Those who have completed these credit requirements are awarded honours degrees subject to the fulfilment of additional requirements. Therefore, it is essential to analyse students' results to make sure that they have fulfilled each and every requirement before awarding respective degrees and honours[1]. This is a time consuming and resource wasting task. Usually ten to fifteen academic staff members work for about ten days to finalise results. As a result, the faculty faces difficulties in releasing results on time. Therefore, an alternative approach is required to expedite this process.

With the advances in Information and Communication Technology, automatic information processing appears to be the best solution for such tasks[2]. This paper proposes a new information system (CUBRAS) to handle the course unit based information processing for the Faculty of Science, University of Peradeniya, which may be applicable to other Universities as well.

2. MATERIALS AND METHODOLOGY

In order to develop an automated system design pattern, modern software process, tools and programming languages are used. The following sections describe these aspects.

2.1 The Model View Controller Pattern

The Model-View-Controller (MVC) architectural pattern separates an application into three main components: model, view, and controller[3, 4]. The MVC architecture is shown in Figure 1.

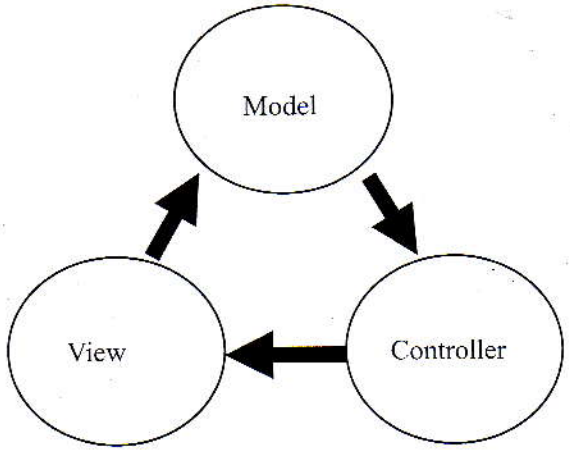

Fig. 1. Architecture of the Model-View-Control pattern

In a MVC application, model represents the data to be processed, controller processes the data in the model and view displays the information. For example, model represents a database, controller handles query-string to retrieve data from the model and view displays the query results.

2.2 Rational Unified Process

A Rational Unified Process (RUP) is a common strategy utilised in the development of software programs. This web-enabled process provides a number of different tools that assist the programmer in fleshing out the basic code for a program, as well as helping to determine the final look of the product[5].

2.3 The Unified Modeling Language

The Unified Modeling Language (UML) is a language for specifying, constructing, visualising, and documenting the artifacts of a software-intensive system. Analogous to the use of architectural blueprints in the construction industry, UML provides a common language for describing software models, and it can be used in conjunction with a wide range of software lifecycles and development processes[6].

2.4 Microsoft Visual Studio

Microsoft Visual Studio is an Integrated Development Environment (IDE). It can be used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight[7].

2.5 SQL Server Management Studio

SQL Server Management Studio is a tool included with Microsoft SQL Server 2005 and later versions for configuring, managing, and administering all components within the Microsoft SQL Server. The tool includes both script editors and graphical tools which work with objects and features of the server[8].

2.6 Microsoft LINQ

Microsoft LINQ defines a set of proprietary query operators that can be used to query, project and filter data in arrays, enumerable classes, XML (XLINQ), relational database, and third party data sources. While it allows any data source to be queried, it requires that the data should be encapsulated as objects. Hence, if the data source does not natively store data as objects, the data must be mapped to the object domain. Queries written using the query operators are executed either by the LINQ query processing engine or, via an extension mechanism. The results of a query are returned as a collection of in-memory objects that can be enumerated using a standard iterator function[9].

2.7 Design of CUBRAS

CUBRAS is designed as a three tier architecture. The user interface of the system is designed as the presentation tier. This tier is designed with web browsers such as Mozilla Fire Fox and Microsoft Internet Explorer. Business logic of the system is designed as the middle tier. It is developed using Internet Information Service (IIS) along with the ASP.Net platform. The bottom tier represents the information storage. In other words bottom tier is the database component of the system. It is constructed by using the MSSQL Server 2005. Three-tier architecture of CUBRAS is depicted in Figure 2.

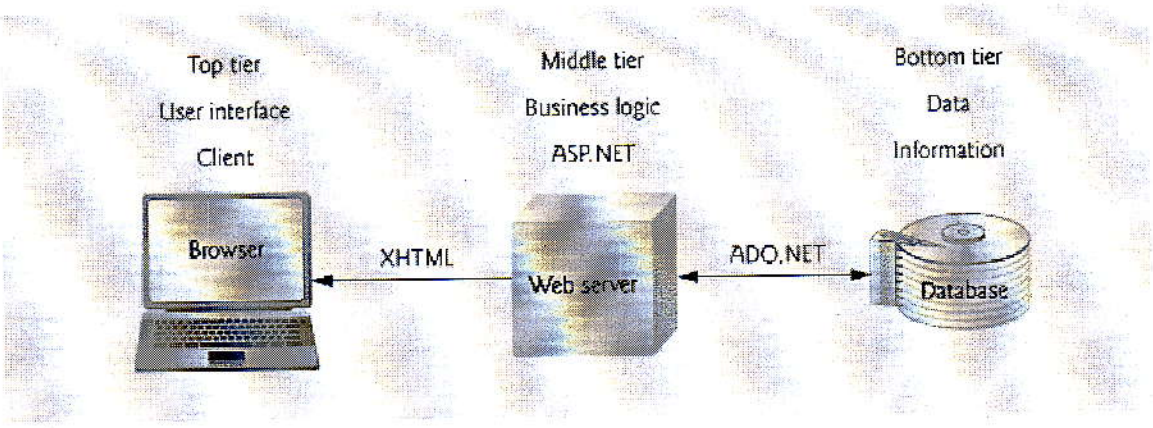

Fig. 2. Three tier architecture of the system

2.7.1 Database Design

As the first step database is modeled as an entity relationship diagram and then transformed into an equivalent relational model. The relational diagram of the CUBRAS database is shown in Figure 3.

2.7.2 Use of MVC (Model-View-Control) pattern

In CUBRAS, the ASP.NET MVC Framework is used to design the routing of URL. When a browser sends a HTTP request to CUBRAS web site, the MVC Framework uses its URL routing engine to map the incoming request onto an action method of a controller class for processing. The Controller class processes the request, handles user input and interactions and executes the application logics such as retrieving and updating the database. After completing the processing the controller class interacts with the view classes to return the output to users. The URL routing mechanism is depicted in Figure 4. For example, user requests for adding or deleting subject combinations are designed through the CUBRAS Controller, Model and Views. When the user requests to add, to delete or to edit by entering data into the user interface, the input is sent to the controller for processing. Then the controller processes the request by interacting with the model. Finally, processed data are sent to the view for displaying.

2.7.3. User interface design

User interface of CUBRAS is designed to provide a simple and an attractive graphical interface to the users. So the graphical user interface (GUI) is designed with three major components: Main interface, administrative interface and view interface. The main interface is designed to allow users to login and to make requests. Administrative interface is designed to provide administrative facilities for secure system management. The view interface is

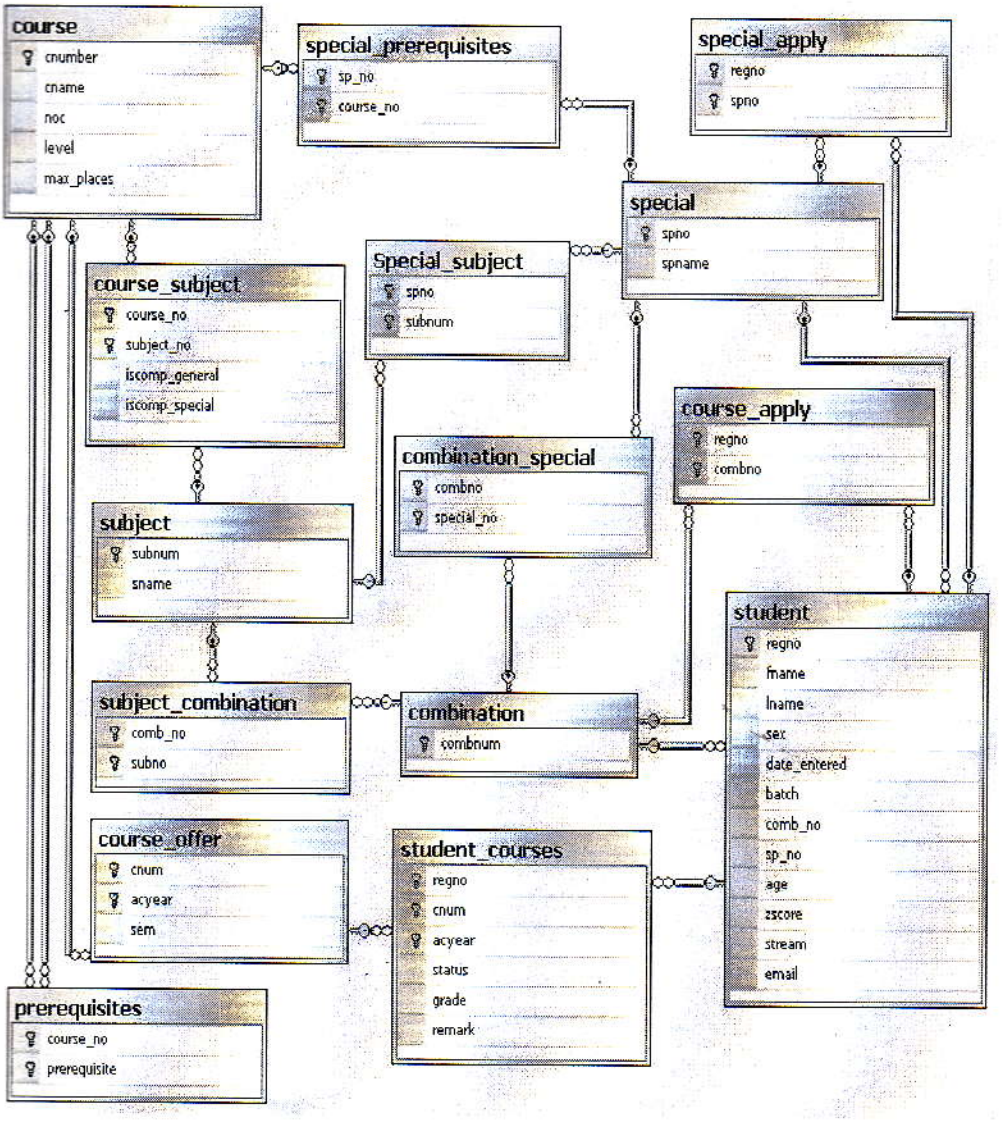

Figure 3. Relational model

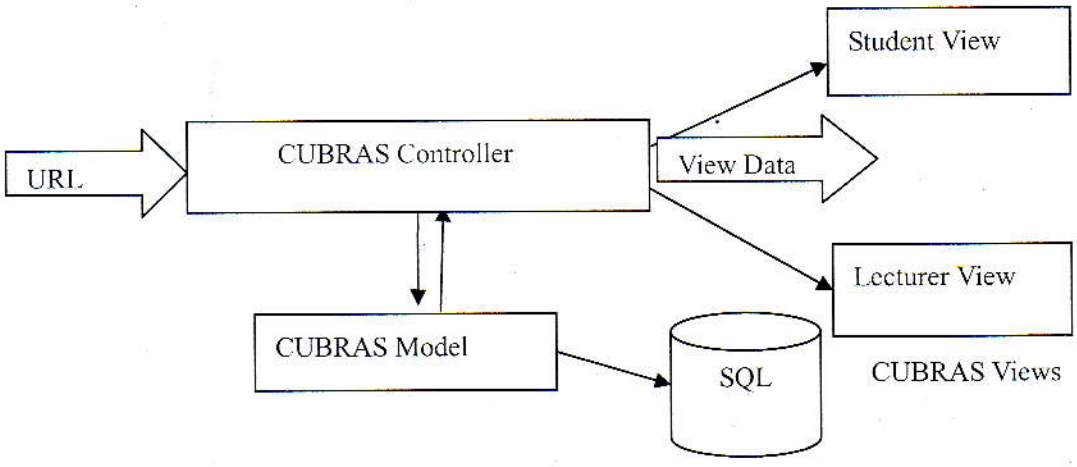

Figure 4. URL Routing mechanism

designed to generate and to print transcripts. The GUI is designed so that the interfaces get dynamically changed according to the user roles. In other words, user access to the system resources is restricted according to the roles of users. For example, facilities provided for students are different from that of an administrator or an academic staff member.

2.8 Functionality of the system

The overall functionality of the system is shown in Figure 5. When a user requests for some information from the CUBRAS, first it checks whether the requested information involves any analysis. If so, the controller performs the analysis by interacting with the model class and returns the results. Otherwise the information is returned by invoking ADO.NET data connection as shown in the flow chart. Since ADO.NET is used in the CUBRAS, the results are either processed directly, or placed in an ADO.NET DataSet object in order to expose them to users.

Vast amount of static data should be filled to the CUBRAS database and they have to be maintained properly. This tedious task is performed by a new actor called data operator.

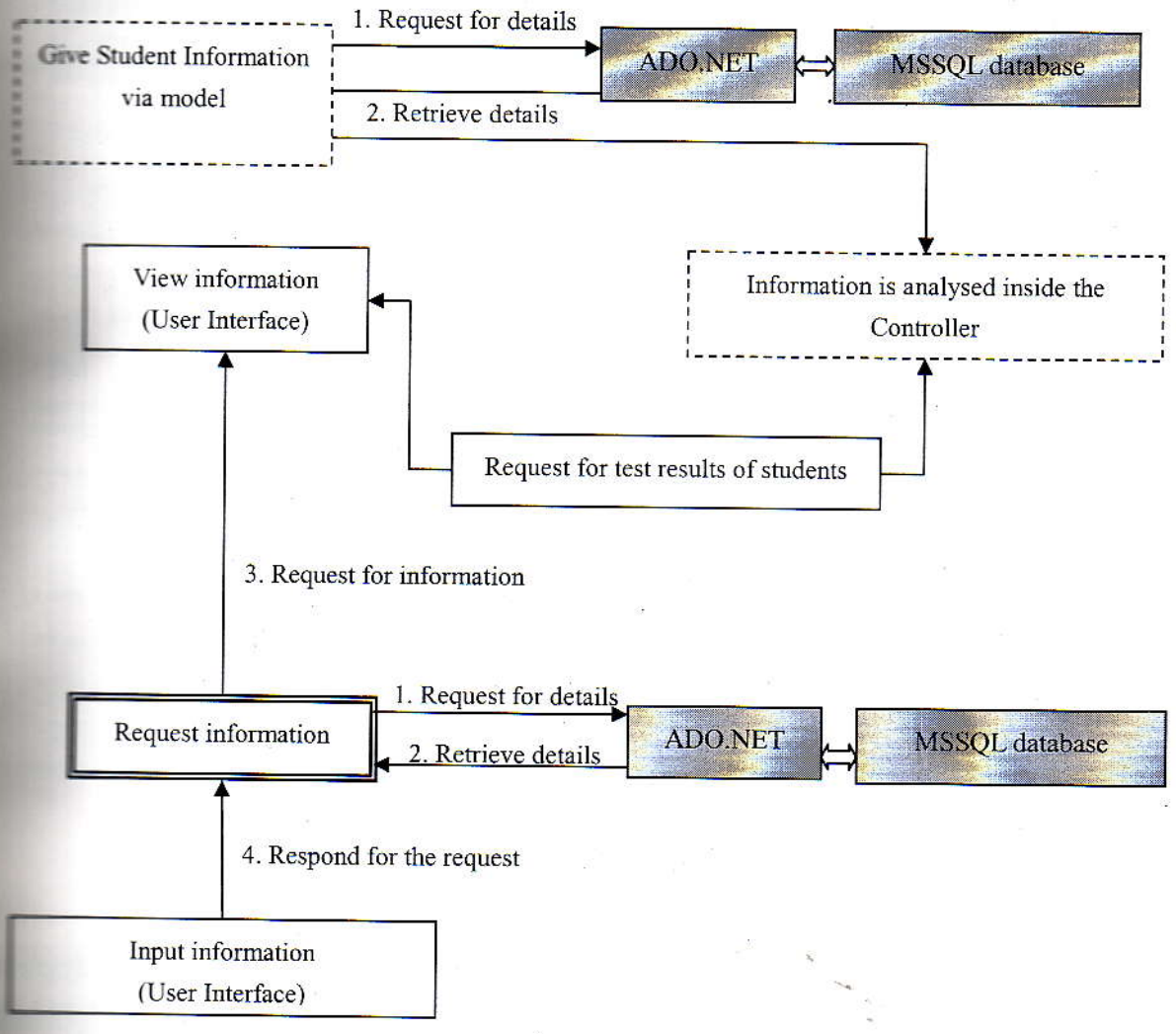

Fig. 5. Workflow of the CUBRAS

3. RESULTS AND DISCUSSION

The system developed was tested with a sample of actual students' grades. The sample grades were derived from students who successfully completed the respective programmes and who could not complete the requirements of the respective programmes. First the grades were stored in the database component of the system and then various result validation activities were performed. According to the test results, student registration and entering of results were consistent. Error messages were returned when users fail to input valid data.

According to the students' subject combinations, students may be eligible to apply for multiple special degree programmes. Therefore the eligibility of each student for special degree programmes was tested at the end of the second year. First the requirements for 100 level and 200 level were validated. The 100 level and 200 level validation results are shown in Figures 6. and 7.

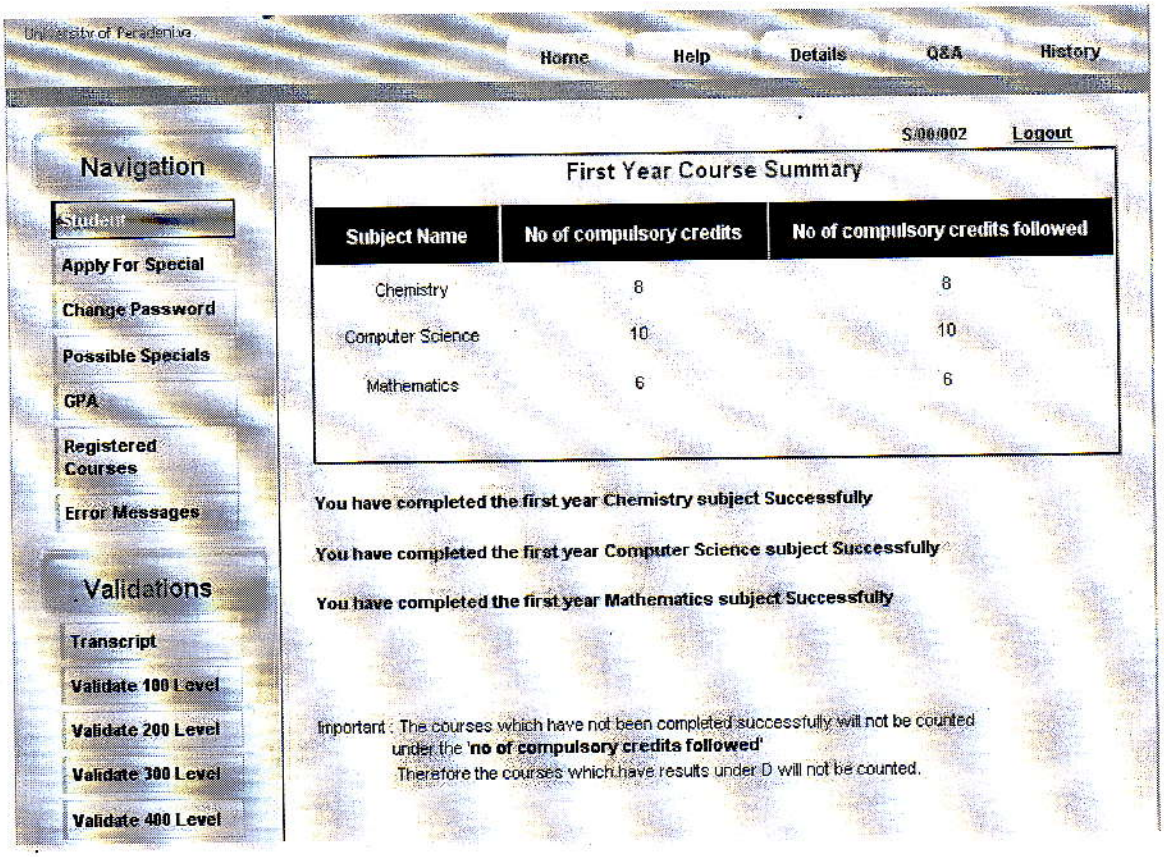

Fig. 6. 100 level results validation

Students who fulfil the 100 level and 200 level requirements can apply for special degrees. Then the system reports the eligible special degree programmes for a given student and the student is allowed to apply for eligible programmes only. Result of an eligibility checking for special degree programmes is depicted in Figure 8.

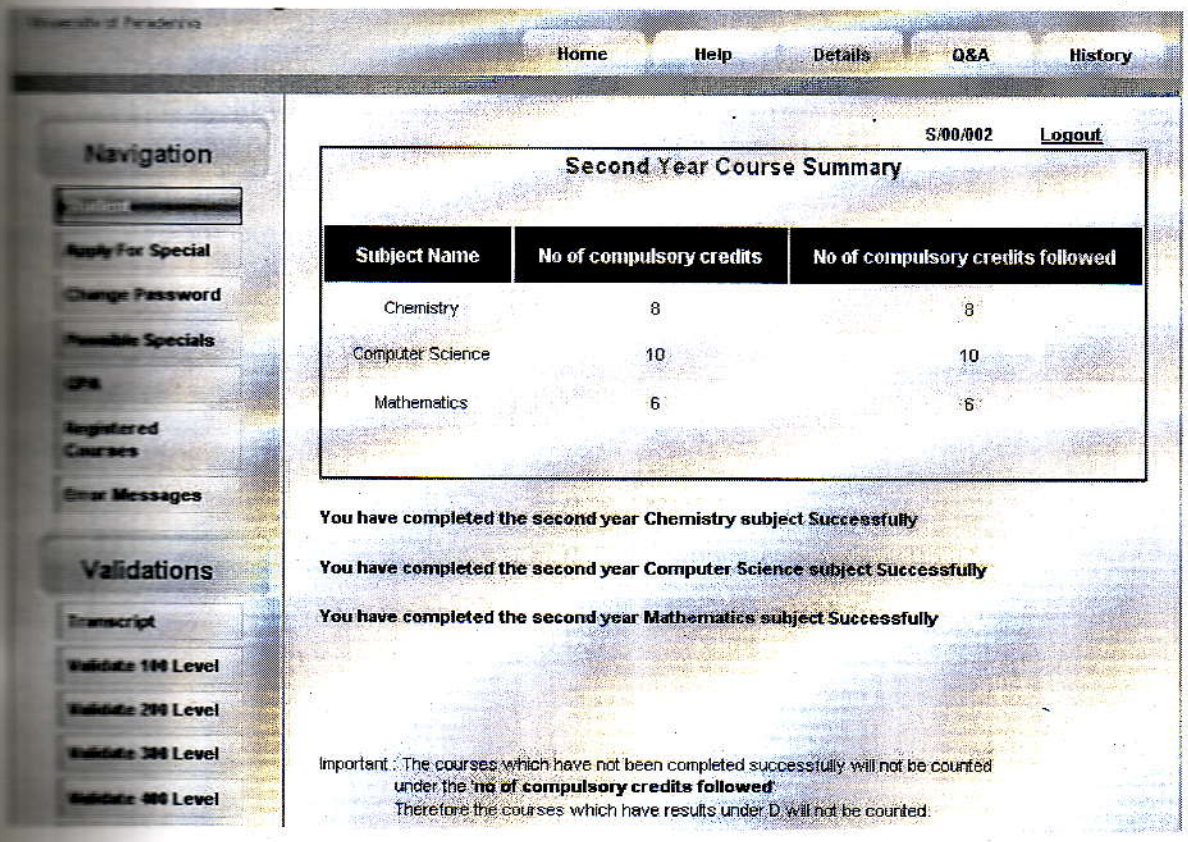

Fig. 7. 200 level results validation

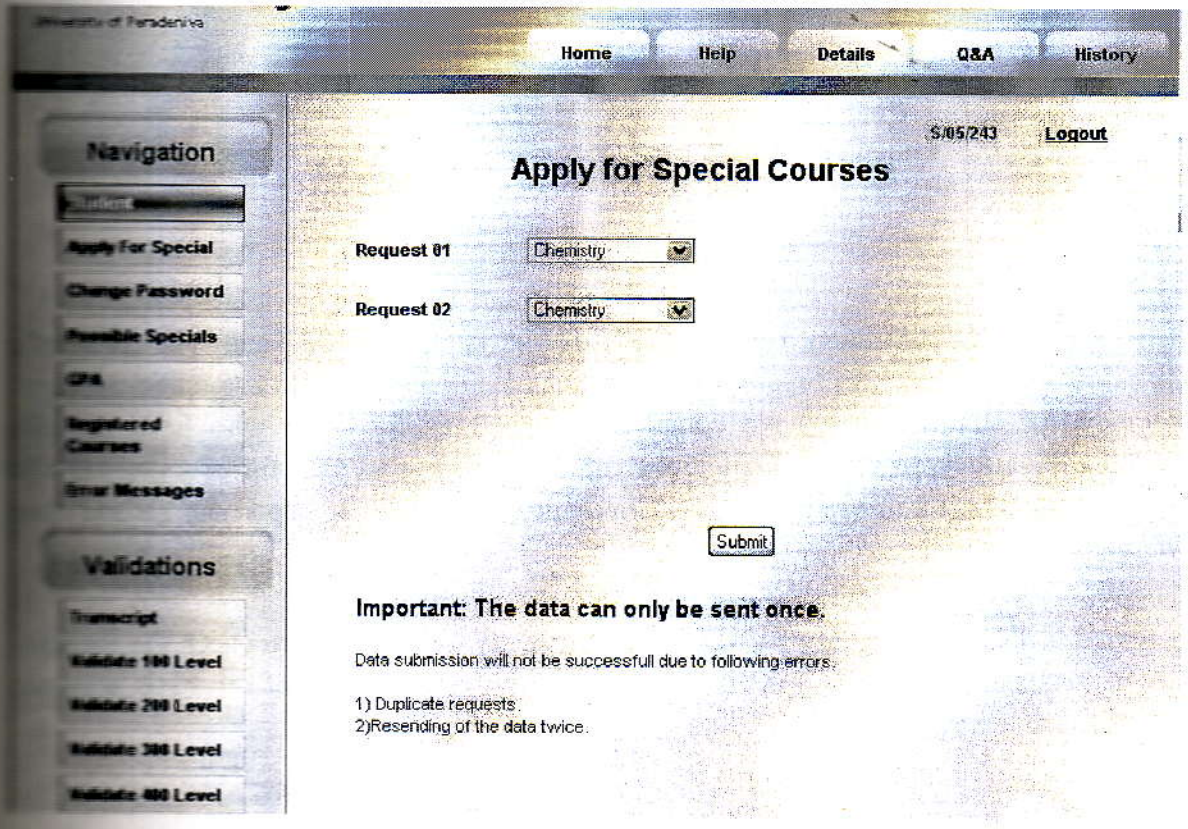

Fig. 8. Eligibility check for special degrees

Once the students complete the duration of their study programme, results were validated in several phases. For each general degree student, level-wise validation was performed to check whether he/she has fulfilled the required credit requirement, and the compulsory and prerequisite requirements. Each student's results were analysed from 100 level to 300 level. 400 level validation was performed only for special degree students. Students who have fulfilled all the requirements at each level were reported to be eligible for awarding the respective degrees. A result of a 400 level validation is shown in Figure 9.

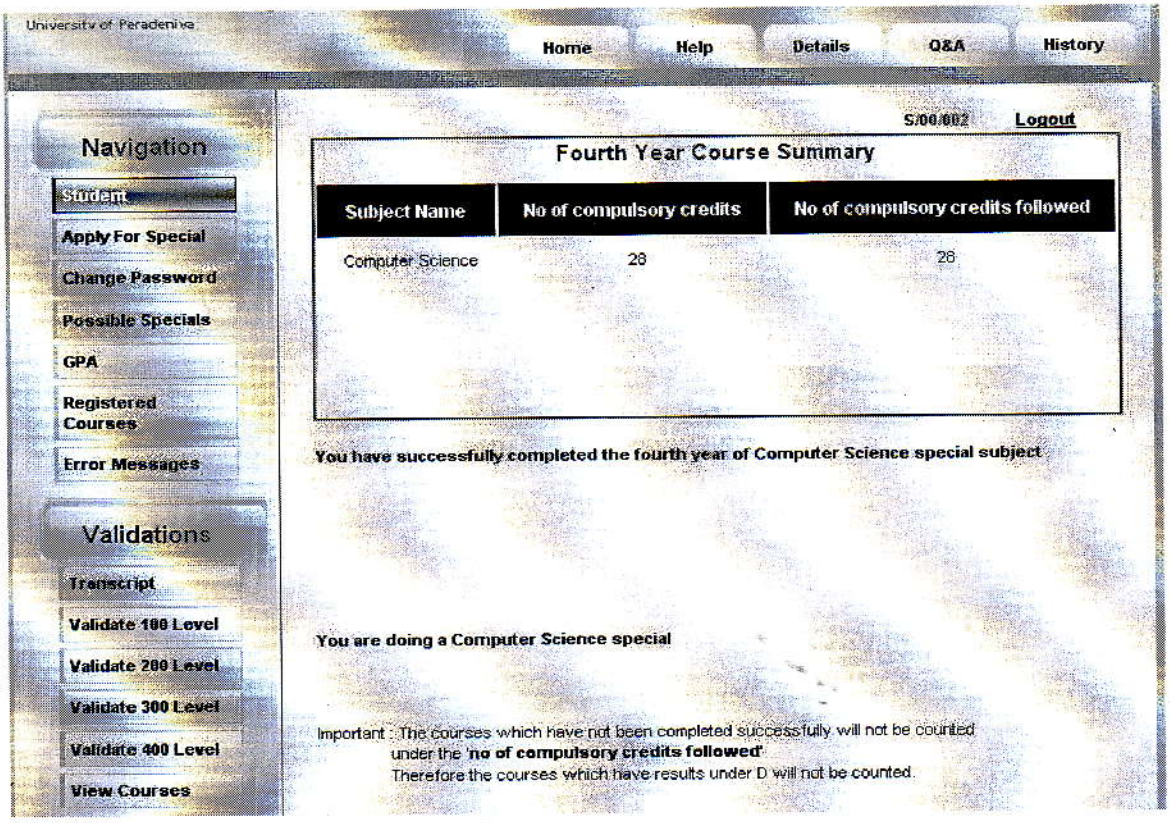

Fig. 9. 400 level results validation

After the level-wise validation, each student's overall GPA was calculated. The result of a GPA calculation is shown in Figure 10. The system also facilitates the automatic transcript generation for students who have completed as well as for those who have partially completed. Figure 11 depicts a sample transcript. After that, eligibility for honours was also tested to determine the respective classes: first class, second class upper and second class lower.

Finally the test results were compared with the manually tested results to check the accuracy of the automated system. The test results indicated that the analysis of results is accurate, efficient and consistent.

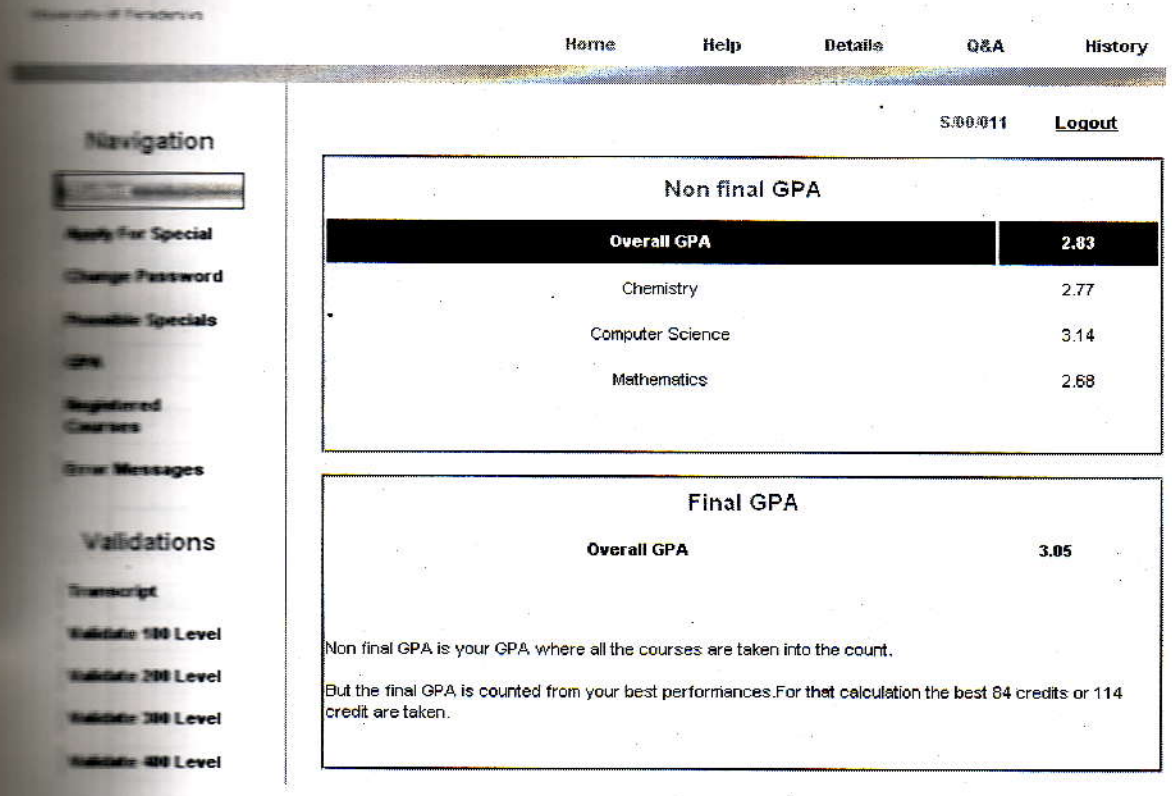

Fig. 10. Calculation of GPA

ACADEMIC RECORD

REGISTRATION NO. : S/05/243
NAME : Pathum Chamikara

Academic Year : 2006/2007

Semester I

Code	Title	Credits	Grade	Status
BL100	Basic Life Science	2	B+	P
CH101	Principles of Chemistry I	3	A	P
CH108	Elementary Chemistry Laboratory	1	A-	P
CS100	Computer Applications	2	B+	P
CS101	Introduction to Computer Science	3	A	P
EN100	Basic English	2	A	P
MT103	Differential Equation	2	A	P
MT104	Abstract Algebra I	3	B-	P

Semester II

Code	Title	Credits	Grade	Status
CH102	Principles of Chemistry II	3	A	P
CH109	Inorganic Chemistry Laboratory	1	A	P
CS102	Programming Techniques	3	A	P
CS103	Programming Laboratory I	2	A	P
MT102	Introduction to Probability Theory	3	B+	P
MT105	Real Analysis I	3	C	P

Academic Year : 2007/2008

Semester I

Code	Title	Credits	Grade	Status
CH211	Inorganic Chemistry I	2	B-	P
CH221	Organic Chemistry I	2	B	P
CH228	Organic Chemistry Laboratory I	1	A-	P
CS201	Data Structures	2	A-	P
CS202	Data Structures Practicals	1	A+	P
CS205	Computer Architecture	2	A	P
ST201	Probability Theory	3	B+	P

Semester II

Code	Title	Credits	Grade	Status
CH212	Inorganic Chemistry II	1	A-	P
CH222	Introductory Organic Synthesis	1	A-	P
CH231	Physical Chemistry I	2	A+	P
CH232	Molecular Properties, Molecular spectroscopy and Spectroscopic Instrumentation	1	A-	P
CH238	Physical Chemistry Laboratory I	1	A-	P
CS203	Database Managemenet Systems	2	A	P
CS204	Programming using Database Management Systems	1	A-	P
CS206	Computer Device Interfacing	2	B+	P
EN200	Englsh for Academic Purpose	2		P
ST203	Theory of Statistics	3	B+	P

Average GPA : 3.54

Final GPA : Your results has not yet been finalized

Fig. 11. System generated academic transcript

4. CONCLUSIONS

An online information system developed can be used to replace the current inefficient and time consuming manual system. The system developed analyses students' results efficiently and accurately. Since the new system is an online system it provides flexibility to both the staff members and students. The fast analysis and validation of results enables the release of students' results quickly. The system developed can be customised to analyse course unit based results at any state university.

REFERENCES

1. Faculty of Science handbook, Faculty of Science, University of Peradeniya, Sri Lanka (2009).
2. Henry C. L., *The Analysis, Design and Implementation of Information Systems*, Mc-Graw Hill, 3rd Editions, New York, USA (1982).
3. Rohnert H., Sommerland P. and Michael S., *Pattern-Oriented Software Architecture: A System of Patterns,* Wiley, New York, USA (1996).
4. MVC pattern (http://weblogs.asp.net/scottgu/), Accessed in June 2008.
5. Cruchten P., *The Rational Unified Process: An Introduction*, 3rd edition, Addison-Wesley, New York, USA (2002).
6. Timothy L., Robert L., and Laganiere R, *Object-Oriented Software Engineering: Practical Software Development using UML and Java,* 1st Edition, McGraw-Hill New York, USA, (2002)
7. Walter S., *ASP.NET Unleashed (2nd Edition),* Sams publication, USA, (2001),
8. MSSQL (http://msdn.microsoft.com) , Accessed in July 2008.
9. MS LINQ (http://www.microsoft.com/sqlserver/2008/en/us/linq.aspx) , Accessed in July 2008.