DESIGN AND IMPLEMENTATION OF WEB BASED ACADEMICS CURRICULUM EVALUATION SYSTEM FOR NUST

By

Rabia Altaf Malik

Aatika Sultana

Sarah Qayyum

A report submitted to the faculty of Computer Science Department Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfillment of the requirements for the degree of BESE in Computer Sciences

March 2008

DESIGN AND IMPLEMENTATION OF WEB BASED ACADEMICS CURRICULUM EVALUATION SYSTEM FOR NUST

by

Rabia Altaf Malik

Aatika Sultana

Sarah Qayyum

A report submitted to the faculty of Computer Science Department Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfillment of the requirements for the degree of BESE in Computer Sciences

March 2008

DEDICATION

In the Name of Allah, The Most Beneficent, The Most Merciful

To our parents

To our supervisor, Mr. Athar Mohsin Zaidi

ACKNOWLEDGEMENTS

After the grace of Almighty Allah, the prayers & support of our parents, we are deeply beholden to our supervisor Mr. Athar Mohsin Zaidi for his continuous assistance, inspiration, patience and unconditional support. We are highly gratified to him for his continuous and valuable suggestions, guidance, and commitment towards provision of undue support throughout our project work.

We wish to acknowledge the continuous support of the Commandant MCS, Mr. Khalid Rao, Chief Instructor Engineering Division, Mr. Raja Iqbal, Head of Computer Science Department, Mr. Naveed Sarfraz Khattak and the faculty and administration of MCS. We are highly thankful to all of our professors who had been guiding and supporting us through out our course and research work. Their knowledge, guidance and training enabled us to carry out this research work.

We would like to offer our admiration to all our classmates, and our seniors who had been supporting, helping and encouraging us throughout our project. We are also indebted to the MCS system administration for their help and support. We would like to offer appreciation to our parents for their vision and commitment to make us learn, and other family members for their encouragement, support and prayers.

ABSTRACT

Evaluation is defined as the process of analysis and assimilation based on merits and real quantification, actual worth of any system, procedure evoked, system in practice, process being adhered for completion or achievement of any desired goals and objectives. Interpreting these definitions in the realm of institute it implies basically to the assessment of faculty, programs, departments and students acquired professional acumen at the terminal ends of any training program/degree. The pyramid analytically focus on strength and weaknesses of the curriculum there by enabling the trainer to pinpoint the weak links may it be syllabi, students intake, their factual knowledge, method of instructions, lack of modern training aids and their subsequent induction, faculty broad based knowledge, imparting of practical knowledge based on latest market trend, research and development process, enhancement of existing teaching facilities and culminating all these identified weak areas into a dynamic, progressive and mission oriented curriculum within the scope and parameter of a defined program. The evaluation must steer towards goals and objectives based on facts and figures relating to the ground realities. It is mandatory to periodically review the system during the semester, at the end of each semester and at the culmination of academic section.

TABLE OF CONTENTS

Chapter # **Description** **Page #**

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Introduction

There is always a need for continuous improvement in any organization especially in education sector the quality can be ensured by continual monitoring of Academics’ and Students’ faculty performance evaluation. The Progressive

Universities requires to maintain a historical academic database and customized information system for survey, research, assessments and critical, trend and comparative analysis. The utility and advantage of such system can only be best achieved by an integrated and automated IT solution. The idea for this project has been derived from this concept. The proposed project will be a web based application which will institute broad based evaluation and assessment system. This system will have multi dimensional usage, students can use it for their result / progress assessment, teaching faculty can monitor and evaluate overall course / class / individual student's performance and senior management, can evaluate instructional performance. The evaluation criteria and spectrum of analysis will be determined by the university defined parameters. The system will intelligently implement them and it will be flexible to adopt any future changes and with positive additions. This system will be customizable therefore it can be implemented in any NUST institute or campus.

1.2 Background

In the back drop of lack of any comprehensive and workable evaluation system, a need has been felt to introduce an analytical, spear headed, goal oriented practicable evaluation system, to be developed and customized to the existing curriculum of the various programs in Military College of Signals.

1.3 Problem Statement

Design and develop an integrated web based comprehensive solution for academics credential, records and performance evaluation. The functional scope of the system is to facilitate students, faculty and management in academics credentials evaluation.

1.4 Aim

The system aims to maintain, archive and retrieve academic data of students and faculty. It is meant to provide accurate, reliable and purposeful analysis.

1.5 Objectives

The system database gradually builds a data warehouse of historic data of students, faculty and academic curriculum. It is a web portal which is versatile to easily adopt additional and future requirements. The system is very user friendly and is easy to use, simple to manage and quick to implement and has modular functional components and structured database. The system design and development is governed by stringent testing and quality standards. It has detail technical, user and reference documentation. The system is secure with role based access control and customized security features.

1.6 Organization of project report

This report is organized into seven chapters. The first chapter is the introduction to the project. It is followed by Literature Review which shows a brief summary of the cited literature. The third chapter constitutes Requirement Analysis. The fourth and fifth chapter give Design and Implementation details respectively. The sixth chapter describes testing and verification methods of the system. Finally the last chapter covers limitations of the system and future work.

Chapter 2

Literature Review

2.1 Introduction

As ACES is a web based system, therefore it is developed using standard web based architecture. For this purpose study related to web based applications was carried out which included research papers and articles. Also the possible languages for implementation were studied in detail.

For the decision of performance evaluation criteria different universities were surveyed and their system of evaluation was studied in detail. Other than the above mentioned areas different statistical testing techniques were also explored. These all areas of study are explained in the sections ahead:

2.2 Motivation

The motivation to design and implement ACES is to facilitate the administration, faculty and students in academic evaluation and also to increase the organizing capability, and monitoring of the smooth functioning of the institution.

2.3 Literature Survey

The literature survey was carried out of the technologies that can be used for development of web based applications. A brief description of the cited literature is elaborated.

2.3.1 Possible Approaches for Developing Web Based Applications

The possible approaches that can be used to develop web-based applications are highlighted in proceeding sections:

2.3.1.1 Java Servlets

Servlets are the Java platform technology that can be used for extending and enhancing Web servers. Servlets provide a component based and platform independent method for building web based applications, without the performance limitations of CGI programs. Unlike proprietary server extension mechanisms (such as the Netscape Server API or Apache modules), servlets are server and platform independent. Servlets have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. Servlets can also access a library of HTTP-specific calls and receive all the benefits of the mature Java language, including portability, performance, reusability, and crash protection. They are a popular choice for building interactive Web applications.

2.3.1.2 Java Server Pages(JSP)

Java Server Pages (JSP) is a Java technology that allows software developers to dynamically generate HTML, XML or other types of documents in response to a web client request. The technology allows Java code and certain pre-defined actions to be embedded into static content. The JSP syntax adds additional XML-like tags, called JSP actions, to be used to invoke built-in functionality. Additionally, the technology allows for the creation of JSP tag libraries that act as extensions to the standard HTML or XML tags. Tag libraries provide a platform independent way of extending the capabilities of a Web server. JSPs are compiled into Java Servlets by a JSP compiler. A JSP compiler may generate a servlet in Java code that is then compiled by the Java compiler, or it may generate byte code for the servlet directly. JSPs can also be interpreted reducing the time taken to reload changes [1].

2.3.1.3 Active Server Pages(ASP.Net)

ASP.NET is a web application framework marketed by Microsoft that programmers can use to build dynamic web sites, web applications and XML web services. It is part of Microsoft's .NET platform and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime, allowing programmers to write ASP.NET code using any Microsoft .NET language.

ASP.NET aims for performance benefits over other script-based technologies (including ASP Classic) by compiling the server-side code to one or more DLL files on the web server. This compilation happens automatically the first time a page is requested that is the developer does not need to perform a separate compilation step for pages. This feature provides the ease of development offered by scripting languages with the performance benefits of a compiled binary. However, the compilation might cause a noticeable delay to the web user when the newly-edited page is first requested from the web server.

The ASPX and other resource files are placed in a virtual host on an Internet Information Services (or other compatible ASP.NET servers). The first time a client requests a page, the .NET framework parses and compiles the file(s) into a .NET assembly and sends the response; subsequent requests are served from the DLL files.

By default ASP.NET will compile the entire site in batches of 1000 files upon first request. If the compilation delay is causing problems, the batch size or the compilation strategy may be changed. Developers can also choose to pre-compile their code before deployment, eliminating the need for just-in-time compilation in a production environment [2].

2.3.1.4 Visual Basic.Net(VB.Net)

Visual Basic .NET provides the features that are most important to programmers, such as object-oriented programming, strings, graphics, graphical-user-interface (GUI) components, exception handling, multithreading, multimedia (audio, images, animation and video), file processing, database processing, Internet and World-Wide-Web-based client/server networking and distributed computing.

The language is appropriate for implementing Internet-based and World-Wide-Web-based applications that can be integrated with PC-based applications. Visual Basic .NET is the next phase in the evolution of Visual Basic.

The .NET platform offers powerful capabilities for software development and deployment, including independence from a specific language or platform. In addition to providing language independence, .NET extends program portability by enabling .NET applications to reside on, and communicate across, multiple platforms—thus facilitating the delivery of Web services over the Internet. .NET enables Web-based applications to be distributed to consumer-electronic devices, such as cell phones and personal digital assistants, as well as to desktop computers.

The capabilities that Microsoft has incorporated into the .NET platform create a new software-development paradigm that will increase programmer productivity and decrease development time.

2.3.1.5 Microsoft SQL Server

Microsoft SQL Server is an application used to create computer databases for the Microsoft Windows family of server operating systems. It provides an environment used to generate databases that can be accessed from workstations, the web, or other media such as a personal digital assistant (PDA).

It is a relational database management system (RDBMS) produced by Microsoft. Its primary query language is Transact-SQL, an implementation of the ANSI/ISO standard Structured Query Language (SQL). The architecture of Microsoft SQL Server is broadly divided into three components: SQLOS which implements the basic services required by SQL Server, including thread scheduling, memory management and I/O management; the Relational Engine, which implements the relational database components including support for databases, tables, queries and stored procedures as well as implementing the type system; and the Protocol Layer which exposes the SQL Server functionality.

2.3.1.6 Oracle Server

The Oracle Server is a relational database management system that provides an open, comprehensive, and integrated approach to information management. An Oracle Server consists of an Oracle database and an Oracle instance. An Oracle database has both a physical and a logical structure. Because the physical and logical server structures are separate, the physical storage of data can be managed without affecting the access to logical storage structures. An Oracle database's physical structure is determined by the operating system files that constitute the database. Each Oracle database is made of three types of files: one or more data files, two or more redo log files, and one or more control files. The files of an Oracle database provide the actual physical storage for database information.

An Oracle database's logical structure is determined by one or more table spaces (A logical area of storage) and the database's schema objects. A schema is a collection of objects. Schema objects are the logical structures that directly refer to the database's data. Schema objects include such structures as tables, views, sequences, stored procedures, synonyms, indexes, clusters, and database links. The logical storage structures, including table spaces, segments, and extents, dictate how the physical space of a database is used. The schema objects and the relationships among them form the relational design of a database.

2.3.2 Statistical Analysis Techniques

Following are some statistical analysis techniques that can be used for verification of the system.

2.3.2.1 Hypothesis Testing

The Hypothesis Testing is a technique used for statistical analysis. A statistical hypothesis is an assumption about a population parameter. This assumption may or may not be true. The best way to determine whether a statistical hypothesis is true would be to examine the entire population. Since that is often impractical, a random sample from the population is examined. If sample data are consistent with the statistical hypothesis, the hypothesis is accepted; if not, it is rejected. There are two types of statistical hypotheses. The null hypothesis, denoted by H_0, is usually the hypothesis that sample observations result purely from chance.

The alternative hypothesis, denoted by H_1 or H_a, is the hypothesis that sample observations are influenced by some non-random cause. The process to determine whether to accept or reject a null hypothesis is based on sample data.

2.3.2.1.1 Steps of Hypothesis Testing

This process, called hypothesis testing, consists of four steps. The first step is to state the hypotheses. This involves stating the null and alternative hypotheses. The hypotheses are stated in such a way that they are mutually exclusive. That is, if one is true, the other must be false.

Secondly we formulate an analysis plan. The analysis plan describes how to use sample data to accept or reject the null hypothesis. The accept/reject decision often focuses around a single test statistic. Then we analyze sample data i.e. find the value of the test statistic (mean score, proportion etc.) described in the analysis plan and complete other computations, as required by the plan.

The last step is to interpret results and apply the decision rule described in the analysis plan. If the test statistic supports the null hypothesis, we accept the null hypothesis; otherwise, reject the null hypothesis.

2.3.2.1.2 Decision Errors

Two types of errors can result from a hypothesis test.

A Type I error occurs when we reject a null hypothesis when it is true. The probability of committing a Type I error is called the significance level. This probability is also called alpha, and is often denoted by α.

A Type II error occurs when we accept a null hypothesis that is false. The probability of committing a Type II error is called Beta, and is often denoted by β. The probability of *not* committing a Type II error is called the Power of the test.

2.3.2.1.3 Decision Rules

The analysis plan includes decision rules for accepting or rejecting the null hypothesis. These decision rules are described in two ways - with reference to a P-value or with reference to a region of acceptance.

The strength of evidence in support of a null hypothesis is measured by the P-value. Suppose the test statistic is equal to *S*. The P-value is the probability of observing a test statistic as extreme as *S*, assuming the null hypothesis is true. If the P-value is less than the significance level, we reject the null hypothesis.

The region of acceptance is a range of values. If the test statistic falls within the region of acceptance, the null hypothesis is accepted. The region of acceptance is defined so that the chance of making a Type I error is equal to the significance level.

The set of values outside the region of acceptance is called the region of rejection. If the test statistic falls within the region of rejection, the null hypothesis is rejected. In such cases, we say that the hypothesis has been rejected at the α level of significance.

2.3.2.1.4 One-Tailed and Two-Tailed Tests

A test of a statistical hypothesis, where the region of rejection is on only one side of the sampling distribution, is called a one-tailed test. For example, if the null hypothesis states that the mean is less than or equal to 10.

The alternative hypothesis would be that the mean is greater than 10. The region of rejection would consist of a range of numbers located on the right side of sampling distribution; that is, a set of numbers greater than 10.

A test of a statistical hypothesis, where the region of rejection is on both sides of the sampling distribution, is called a two-tailed test. For example, if the null hypothesis states that the mean is equal to 10.

The alternative hypothesis would be that the mean is less than 10 or greater than 10. The region of rejection would consist of a range of numbers located on both sides of sampling distribution; that is, the region of rejection would consist partly of numbers that were less than 10 and partly of numbers that were greater than 10 [3].

2.3.2.2 The Chi-Square Test

Chi-square is a statistical test commonly used to compare observed data with data we would expect to obtain according to a specific hypothesis. The chi-square test is always testing what scientists call the null hypothesis**,** which states that there is no significant difference between the expected and observed result.

The formula for calculating chi-square (chi^2) is:

$$\text{chi}^2 = \ (o\text{-}e)^2/e$$

That is, chi-square is the sum of the squared difference between observed (*o*) and the expected (*e*) data (or the deviation, *d*), divided by the expected data in all possible categories. Chi-square requires that numerical values are used instead of percentages or ratios.

2.3.2.2.1 Step by Step Procedure

First of all state the hypothesis being tested and the predicted results. Gather the data by conducting the proper experiment. Then determine the expected numbers for each observational class. Chi-square should not be calculated if the expected value in any category is less than 5. The next step is to calculate chi^2 using the formula. All calculations are completed to three significant digits. The answer is rounded off to two significant digits. The chi-square distribution table is used to determine significance of the value.

Determine degrees of freedom and locate the value in the appropriate column. Locate the value closest to the calculated chi^2 on that degrees of freedom *df* row. Move up the column to determine the p value, and state the conclusion in terms of hypothesis.

If the *p* value for the calculated chi^2 is $p > 0.05$, accept the hypothesis. 'The deviation is small enough that chance alone accounts for it. A *p* value of 0.6, for example, means that there is a 60% probability that any deviation from expected is due to chance only. This is within the range of acceptable deviation.

If the p value for the calculated chi^2 is $p < 0.05$, reject the hypothesis, and conclude that some factor other than chance is operating for the deviation to be so great. For example, a p value of 0.01 means that there is only a 1% chance that this deviation is due to chance alone. Therefore, other factors must be involved [4].

2.3.3 Statistical Analysis Tools

Among the statistical Analysis tools two important tools were studied and explored mentioned in the proceeding sections.

2.3.3.1 SPSS

SPSS, formerly "Statistical Package for the Social Sciences," is quite popular not only because it is a valuable analysis tool, but also because it's user-friendly and visually oriented. SPSS originally emphasized its programming language, but soon became "pull down" inclined via its windowing environment. But the SPSS programming language and syntax is still where the real power of this program resides. SPSS can read and write data from ASCII text files (including hierarchical files), other statistics packages, spreadsheets and databases. SPSS can read and write to external relational database tables via ODBC and SQL.

SPSS places constraints on internal file structure, data types, data processing and matching files, which together considerably simplify programming. SPSS datasets have a 2-dimensional table structure where the rows typically represent cases (such as individuals or households) and the columns represent measurements (such as age or household income). Only 2 data types are defined: numeric and text (or "string"). All data processing occurs sequentially case-by-case through the file. Files can be matched one-to-one and one-to-many, but not many-to-many.

SPSS, like other statistical programs, has evolved into a set of modules. Most of SPSS's modules are available in its main installation media. Some modules that are SPSS's versions of particular algorithms have been packaged as stand-alone products. The installation CD for SPSS contains the following modules: Windows Base, Professional Statistics, Advanced Statistics, Tables, Trends, Categories, Conjoint, Exact Tests, Missing Value, Smart Viewer, and Maps [5].

2.3.3.2 Stata

Stata is a complete, integrated statistical package that provides everything needed for data analysis, data management, and graphics. With a point-and-click interface, an intuitive command syntax, and online help, Stata is easy to use, fast, and accurate. All analyses can be reproduced and documented for publication and review.

Stata’s data-management commands give us complete control of all types of data: we can combine and reshape datasets, manage variables, and collect statistics across groups or replicates. We can work with byte, integer, long, floats, double, and string variables. Stata also has advanced tools for managing specialized data such as time-series data, categorical data, and survey data.

Stata makes it easy to generate publication-quality, distinctly styled graphs, including regression fit graphs, distributional plots, time-series graphs, and survival plots. With the integrated Graph Editor we can click to change anything about the graph or to add titles, notes, lines, arrows, and text [6].

2.4 Conclusion

This chapter summarizes the technologies and development languages that were studied for the development of ACES. The study included books and articles on the related topics. This gave a proper start to the project after this phase of reviewing the existing literature.

Chapter 3

Requirement Analysis

3.1 Introduction

This chapter deals with the development of Software Requirements that specify what an Academic Curriculum Evaluation System should and should not do.

3.1.1 System Context

This section clearly depicts the environment and boundaries of the Academic Curriculum Evaluation System and the entities with which it interacts. It helps us see how the system fits into the existing scheme of things. What the system will do by itself and what it expects other entities to do is clearly delineated.

3.1.2 Functional Requirements

This section is the bulk of the chapter and precisely states the functions of the system – what it should do and what it should not. This section is split into subsections. Freedom from ambiguity and navigability were kept in mind while documentation. A consistent terminology has been followed throughout.

3.1.3 Non-functional Requirements

These are quality requirements that stipulate the performance levels required of the system for various kinds of activities. Numerical lower and upper limits set conditions on the response times, access times etc of the system. Sometimes, tradeoffs are necessary among various non-functional requirements.

3.2 System Context

The Academic Curriculum Evaluation System will provide the easy-to-use, interactive, and intuitive graphical user interfaces. The primary users of the system will be students, faculty members and the administrators. The access rights will be different for all types of users. The web pages that will be available to a user will contain only those services that are allowed for that user. The users of the system will be authenticated users.

3.3 Functional Requirements

The functional requirements of Academics Curriculum Evaluation System cover following areas.

3.3.1 Logging Facility

The system shall provide a facility for its users to log onto the system through a web page that requires user’s login name and password as input.

3.3.2 Create Profile

All types of users should be able to create and edit their profiles to keep their personal information available to the system.

3.3.3 Students Perspective

There are two sub perspectives for students as View Semester-Wise Result and View Course-Wise Result.

3.3.3.1 View Semester-Wise Result

The students should be able to view the result of all the previous semesters they have studied. The selection of the semester is made through a dropdown list that contains all previous semesters. This result will contain the Grade Point Average

(GPA) of that particular semester, the grades achieved in the courses in that semester and cumulative GPA (CGPA) up till the selected semester.

3.3.3.2 View Course-Wise Result

The students will also be able to view the detailed result for every course they have taken that includes the marks for sessionals, finals, labs and the grade.

3.3.4 Faculty perspective

There are three sub perspectives for faculty as View Personal Assessment Result, View Student's Result and Add/Update/Delete Student's Results.

3.3.4.1 View Personal Assessment Result

The faculty members will be able to view their assessment results when they log into the system.

He/she will be restricted to view other faculty members assessment only authorized to view own assessment.

3.3.4.2 View Student's Result

Faculty members will be able to view student's results for the courses they have taught.

3.3.4.3 Add/Update/Delete Student's Results

The faculty members will add delete and update the student's results after compilation of results.

3.3.5 Administration Perspective

There are three sub perspectives in administration as View Assessment Result of Faculty, View Student's Result and Upload Information.

3.3.5.1 View Assessment Result of Faculty

The administration will be able to view the assessment result of all faculty members.

3.3.5.2 View Student’s Result

The administration will be able to view with details the results of all the students whenever required.

3.3.5.3 Upload Information

The administration will have the writing access to the system to make updates and uploading of information.

3.3.6 Examination Branch Perspective

The two sub perspectives of examination branch are Upload Result and Review Result.

3.3.6.1 Upload Result

The final result of any semester i.e. the grades will be uploaded by the Examination Branch.

3.3.6.2 Review Results

The Examination Branch will be able to review the results in case of any mistake within a defined period of time after the declaration of results.

3.4 Non-functional Requirements

The non-functional requirements of Academics Curriculum Evaluation System cover following areas.

3.4.1 Performance

Response time of the Academic Curriculum Evaluation System should be less than 5 seconds most of the time. Response time refers to the waiting time while the system accesses, queries and retrieves the information from the database.The system shall be able to handle at least 500 transactions/inquiries per second and shall show no visible deterioration in response time as the number of users or data increases

3.4.2 Reliability

Academic Curriculum Evaluation System shall be available 24 hours a day, 7 days a week and shall always provide real time information. The system shall be robust enough to have a high degree of fault tolerance. For example, in case of invalid inputs, the system should not crash and shall identify the invalid input and produce a suitable error message.

Academic Curriculum Evaluation System shall be able to recover from hardware failures, power failures and other natural catastrophes and rollback the databases to their most recent valid state.

3.4.3 Usability

Academic Curriculum Evaluation System shall provide a easy-to-use graphical user interface. The web interface should be intuitive and easily navigable. Users should be able to understand the menu and options provided by Academic Curriculum Evaluation System. Any notification or error messages generated by Academic Curriculum Evaluation System shall be clear, succinct, polite and free of jargon.

3.4.4 Integrity

Only system administrator has the right to change system parameters, such as access rights etc. Users need to be authenticated before having access to any confidential data.

3.4.5 Interoperability

ACES shall minimize the effort required to couple it to another system.

3.6 Conclusion

The chapter encompasses the functional and non functional requirements of Academic Curriculum Evaluation System. It provides an insight and tool both to the students and faculty to feed and know their performance. Yet access to all time monitoring facility to the management for timely injections of improvement measures at different stages of programs being taught in college.

Chapter 4

System Design

4.1 Introduction

This chapter incorporates all software engineering concepts required for designing any system, covering architectural, DFD, activity models highlighting their uses. The state transition diagram, logical and sequential models thereby culminating at mathematical model designed for faculty evaluation.

4.2 Proposed System Architecture

A Web application is an application that is accessed via web over a network such as the Internet or an intranet. Web applications are popular due to the ubiquity of a client, sometimes called a thin client. The ability to update and maintain Web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity.

A significant advantage of building Web applications to support standard browser features is that they should perform as specified regardless of the operating system or OS version installed on a given client. Though many variations are possible, a Web application is commonly structured as a three-tiered application.

In its most common form, a Web browser is the first tier, an engine using some dynamic Web content technology is the middle tier, and a database is the third tier as depicted in Figure 4.1. The Web browser sends requests to the middle tier, which services them by making queries and updates against the database and generates a user interface.

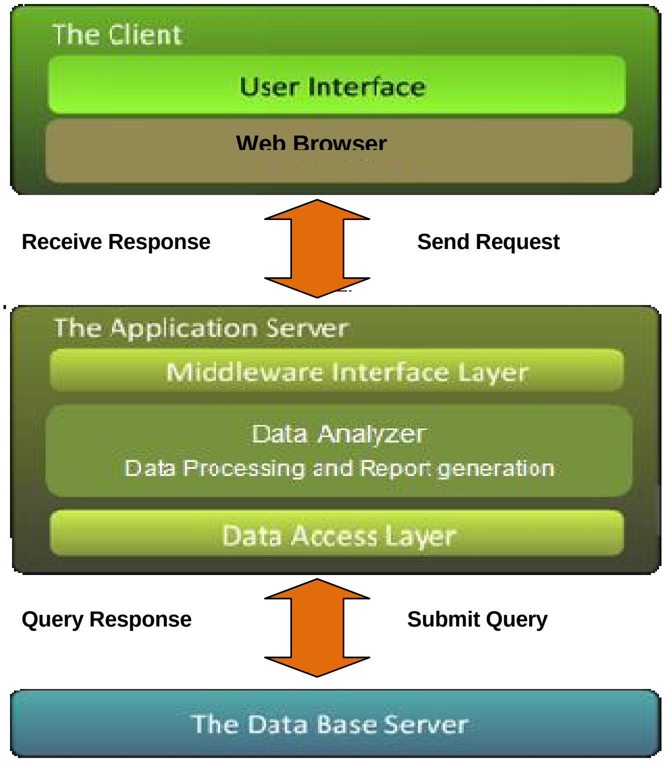


Figure 4.1 Proposed Architecture

4.3 System Development Methodology of ACES

The system is developed in accordance with a well defined System Development life Cycle (SDLC). The methodology is incremental water fall model with its distinctive elaborate software engineering processes depicted as five phases which are requirements definition, system design, development and implementation, testing and verification and maintenance as shown in Figure 4.2.

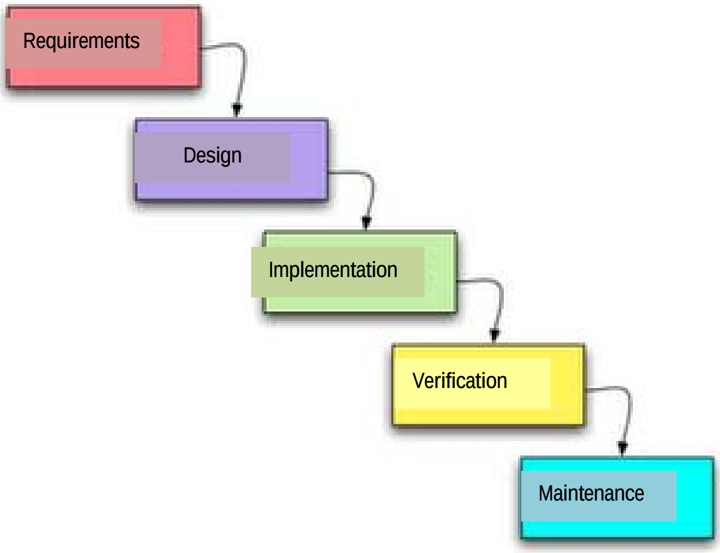


Figure 4.2 Incremental Waterfall Model

Following is the main overview of the phases along with its artifacts:

4.3.1 Requirement Definition

Formatted: Bullets and Numbering

In depth the requirements study and analysis is carried out. All manual / existing processes are studied and evaluated data requirements / data flows ascertained and business rules recorded. Detailed Requirement Specification (RS) Document is produced.

The RS provides the inputs for process flow diagrams of as is system and use cases are developed for proposed to be system these details are documented in Functional Specifications (FS).

4.3.2 System Design

FS is transformed into detail database (DB) schema design and system / software specifications (SRS). All GUIs are designed and system integration structure is created.

4.3.3 System Development

Formatted: Bullets and Numbering

The modules development and unit testing is completed at this stage. The system / sub system integration and functional testing is conducted.

4.3.4 Implementation

The system implementation is according to an elaborate implementation plan which is defined in design phase. This includes data migration, application installation and system configuration.

4.4 Design and Architecture

This section incorporates architecture and software design model of ACES.

4.4.1 Features

ACES provide an environment which is very useful for students in general and faculty members in specific. The real advantage of ACES is the improvement of academic performance and overall quality assurance by the senior management of NUST and its constituent colleges / campuses.

The explicit advantages of ACES are listed in Table 4.1.

Table 4.1 Advantages of ACES

ADVANTAGES OF ACES	
1	Self assessment by students
2	Performance assessment by instructors
3	Academics monitoring by the management
4	Comparative analysis on multi dimensional parameters
5	Trend analysis based on various variable / factors
6	Comprehensive reporting
7	Data security and access control
8	User friendly and adaptive
9	Flexible and scalable
10	Easy to customize and implement

4.4.2 Architectural Diagrams

Three layer architecture has been used as presentation layer (involve aspx pages), business layer (BLL) and the data layer (DAL). This three layer modeling is also referred to as 3-tier architecture as in Figure 4.3.

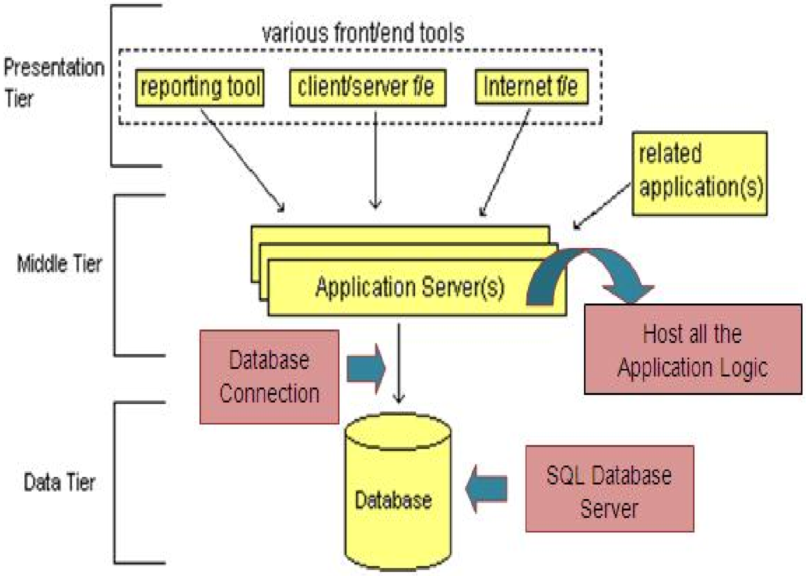


Figure 4.3 3-Tier Architecture

The architectural flow in Figure 4.4 depicts the request and response sequences over the web. The remote user's computer usually the client, send request via internet in the form of http request to the web server. Web server further communicates with the server program or application server. It sends data request as query to the database server. The database server retrieves data and forwards it as query response to application server. The application server receives requested data, analyses and processes data and generates reports. These reports are send to client as html responses.

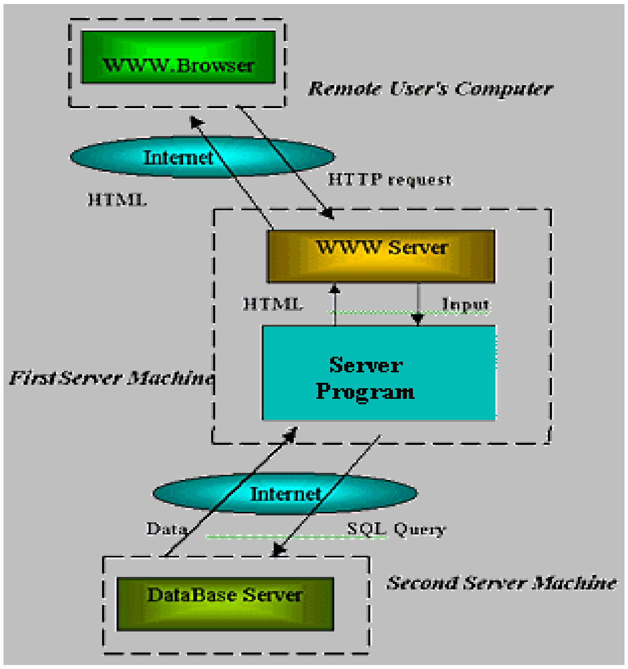


Figure 4.4 Architectural Flow

4.4.3 Data Flow Diagram

The data flow of different system scenarios is depicted in the proceeding sub sections.

4.4.3.1 User Login Data Flow Diagram

The user is asked for login ID and password and after authentication the user is logged into the system.

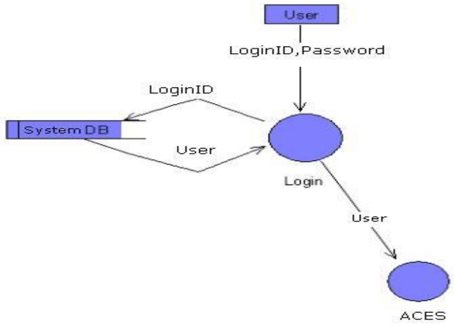


Figure 4.5 User Login DFD

4.4.3.2 Change Password Data Flow Diagram

The user is required to enter old password, new password and then reconfirm new password in order to change password.

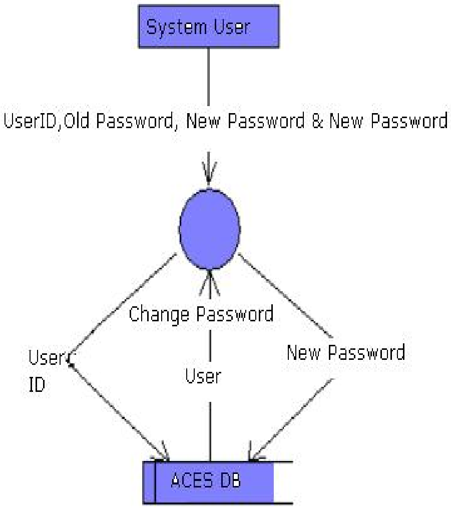


Figure 4.6 Change Password DFD

4.4.3.3 Student User Data Flow Diagram

When user is logged in as a student he/she is able to perform functions specified in Figure 4.7.

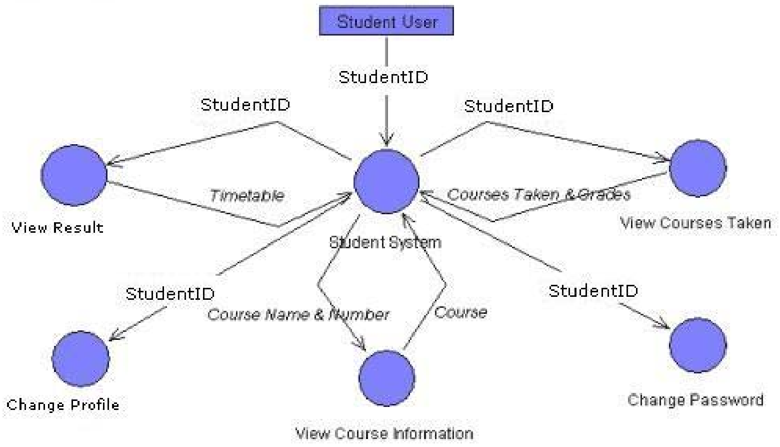


Figure 4.7 Student DFD

4.4.3.4 Faculty User Data Flow Diagram

When user is logged in as a faculty member he/she is able to perform functions specified in Figure 4.8.

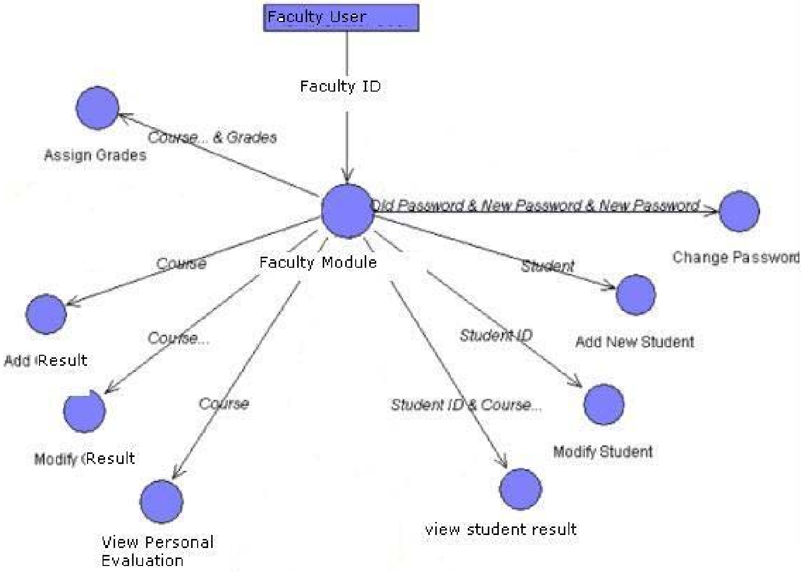


Figure 4.8 Faculty DFD

4.4.3.5 Exam-Branch User Data Flow Diagram

When user is logged in as a exam branch representative he is able to perform functions specified in Figure 4.9.

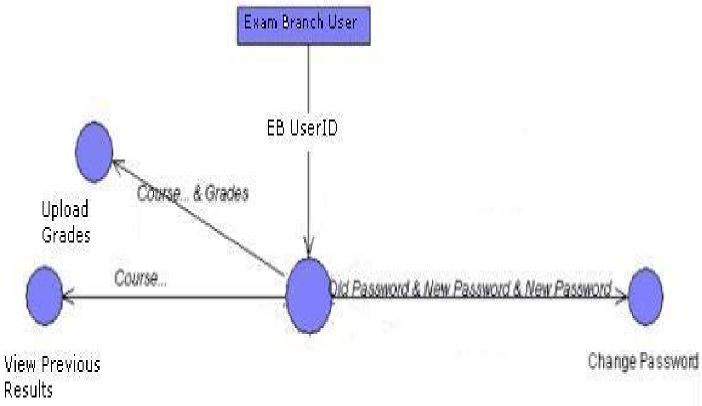


Figure 4.9 Exam-Branch DFD

4.4.3.6 Administration User Data Flow Diagram

When user is logged in as an administration body he/she is able to perform functions specified in Figure 4.10.

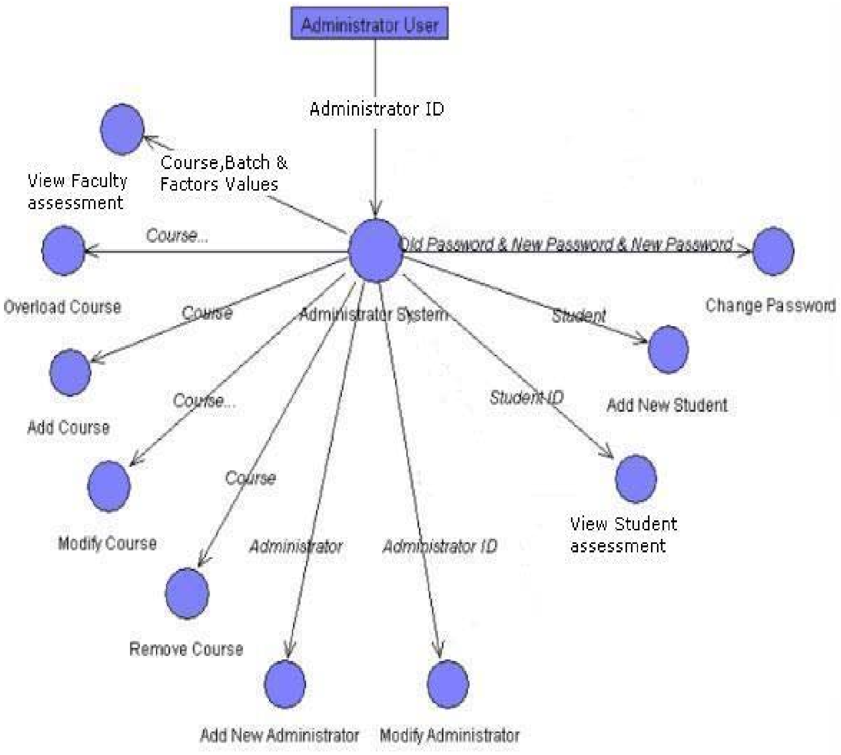


Figure 4.10 Administration DFD

4.4.4 Activity Diagram of ACES

Following are the system activity diagrams. Activity diagrams have been used for business process modeling, for modeling the logic captured by a single use case or usage scenario. They are used to highlight the detailed logic of a business rule or to model the internal logic.

4.4.4.1 Activity Diagram for Login

Figure 4.11 shows the activity diagram for the user to log in to the system.

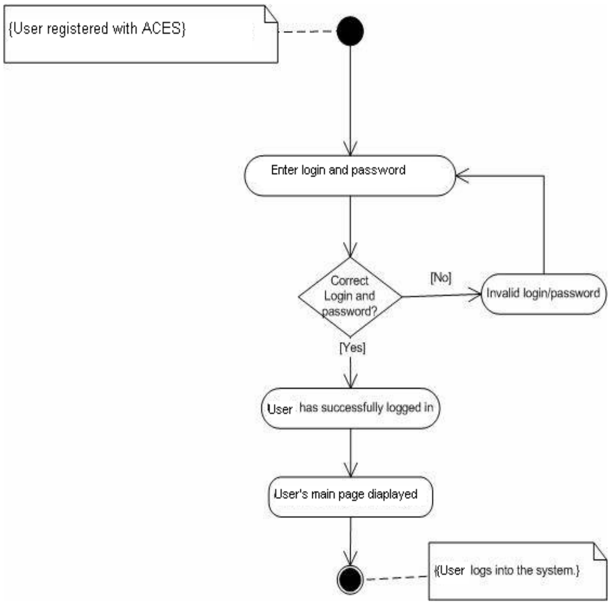

{User registered with ACES}
Enter login and password
Correct Login and password?
[No]
Invalid login/password
[Yes]
User has successfully logged in
User's main page diaplayed
{User logs into the system.}

Figure 4.11 Login Activity Diagram

4.4.4.2 Activity Diagram for Logout

Figure 4.12 shows the activity diagram for the user to log out from the system.

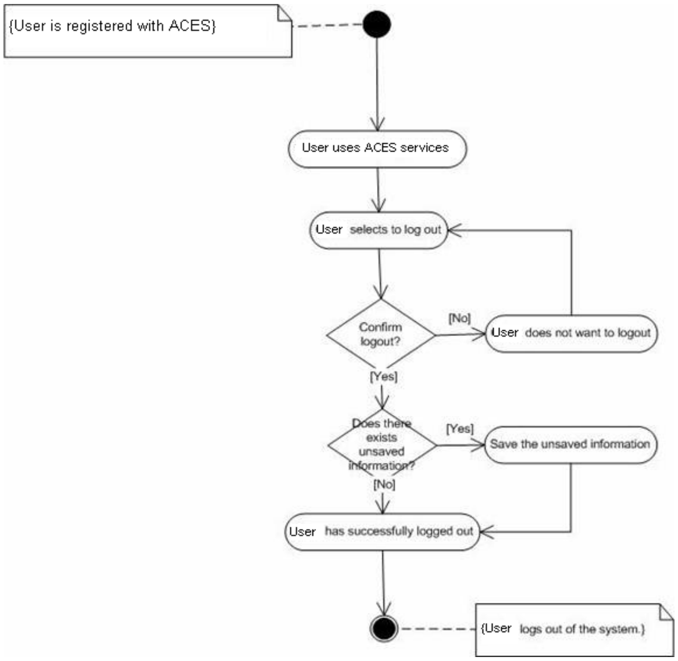

{User is registered with ACES}
User uses ACES services
User selects to log out
Confirm logout?
[No]
User does not want to logout
[Yes]
Does there exists unsaved information?
[Yes]
Save the unsaved information
[No]
User has successfully logged out
{User logs out of the system.}

Figure 4.12 Logout Activity Diagram

4.4.4.3 Activity Diagram for New User Registration

Figure 4.13 shows the activity diagram for the New User Registration to the system.

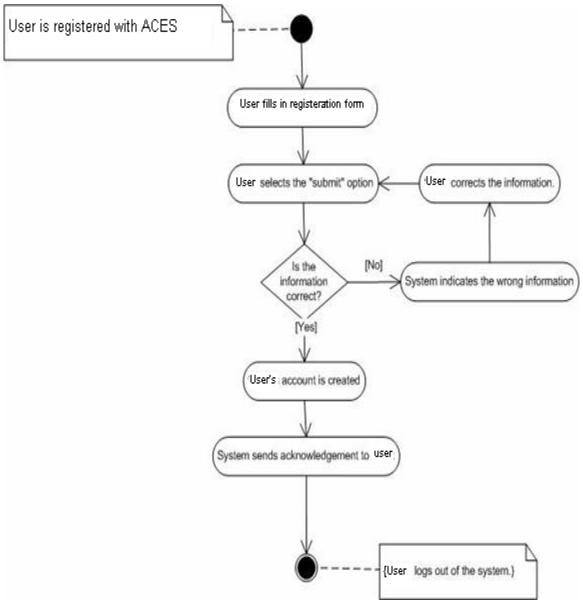

User is registered with ACES
User fills in registeration form
User selects the "submit" option
User corrects the information.
Is the information correct?
[No]
System indicates the wrong information
[Yes]
User's account is created
System sends acknowledgement to user.
{User logs out of the system.}

Figure 4.13 New User Registration Activity Diagram

4.4.4.4 Activity Diagram to View Account

Figure 4.14 shows the activity diagram for the View Account in the system.

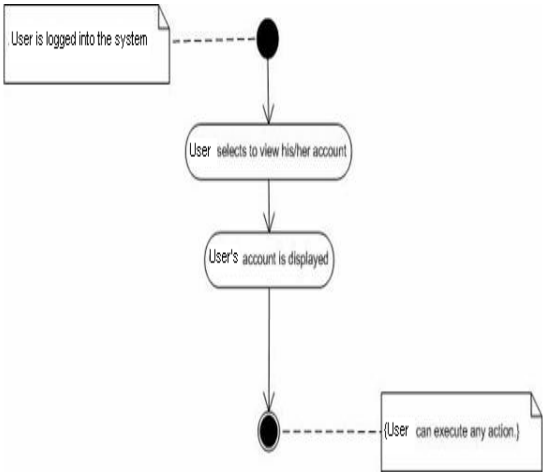

User is logged into the system
User selects to view his/her account
User's account is displayed
{User can execute any action.}

Figure 4.14 View Account Activity Diagram

4.4.4.5 Activity Diagram to Change Account Information

Figure 4.15 shows the activity diagram for the Change Account in the system.

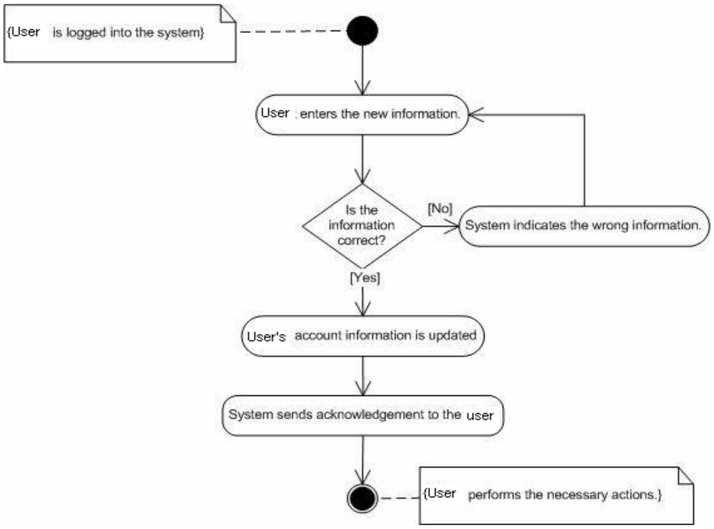

{User is logged into the system}
User : enters the new information.
Is the information correct?
[No]
System indicates the wrong information.
[Yes]
User's account information is updated
System sends acknowledgement to the user
{User performs the necessary actions.}

Figure 4.15 Change Account Information Activity Diagram

4.4.4.6 Activity Diagram to Retrieve Information

Figure 4.16 shows the activity diagram for the Retrieve Information from the system.

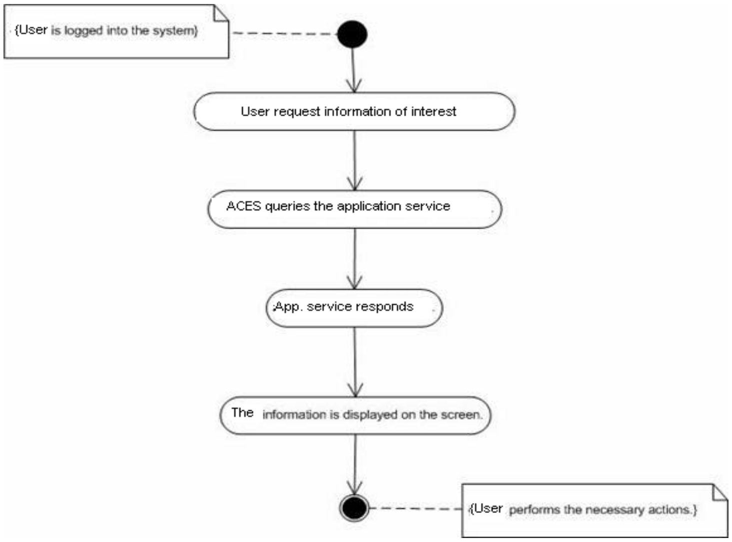

{User is logged into the system}
User request information of interest
ACES queries the application service
App. service responds
The information is displayed on the screen.
{User performs the necessary actions.}

Figure 4.16 Retrieve Information Activity Diagram

4.4.5 Use Cases of ACES

The proceeding sections are the system use cases.

4.4.5.1 System Level Use-Case

The Figure 4.17 highlights the system level use case of ACES.

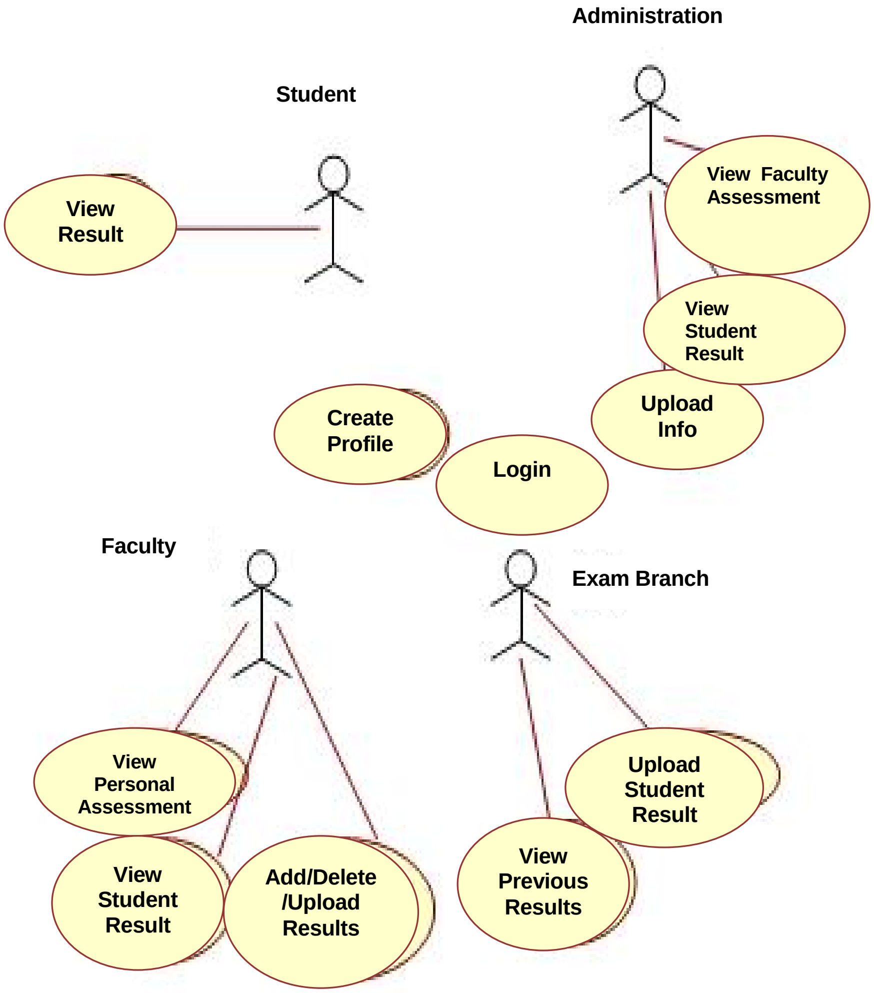


Figure 4.17 System Use-Case

4.4.5.2 Functional Use-Cases

The functional use cases are described in the sub sections ahead.

4.4.5.2.1 Login

Table 4.2 Login Description

Primary Actor	User (student, faculty member, administration representative, examination branch representative)
Preconditions	The user has opened the homepage of the system
Post Conditions	The user is authenticated and logged on to the system
Basic flow	The user enters his username and password. System authenticates the user and finally the user is logged in
Extensions	Either the username or the password entered is incorrect and user is not authenticated and error message is displayed

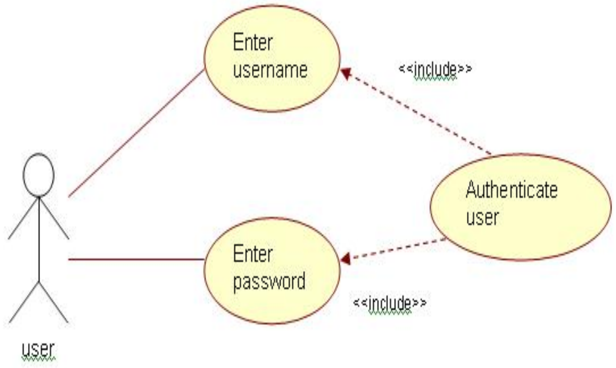


Figure 4.18 Login Use-Case

4.4.5.2.2 Create Profile

Table 4.3 Create Profile Description

Primary Actor	User (student, faculty member, administration, representative, examination branch representative)
Preconditions	The user has logged in to the system

Post Conditions	The user profile is created
Basic flow	The user enters his personal, professional and account information through a form. System saves the information in the database
Extensions	The user information is not saved properly

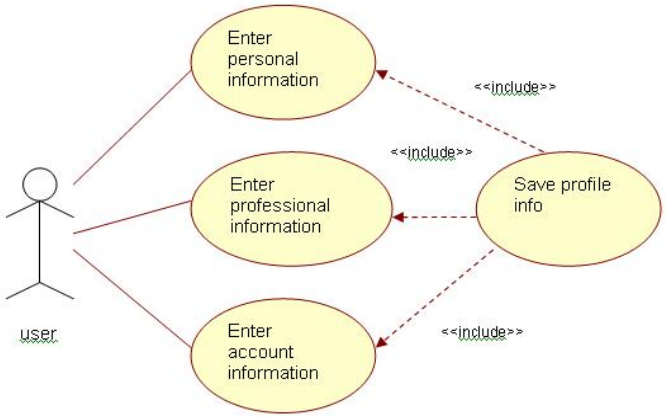

Enter personal information
<<include>>
<<include>>
Enter professional information
Save profile info
user
Enter account information
<<include>>

Figure 4.19 Create Profile Use-Case

4.4.5.2.3 Student-View Result

Table 4.4 Student-View Result Description

Primary Actor	Student
Preconditions	The student has logged in to the system.
Post Conditions	The student result is displayed
Basic flow	The student selects the semester for which the result is required and then selects the course to view result. The detailed result is then displayed for the selected course
Extensions	The required result is not displayed

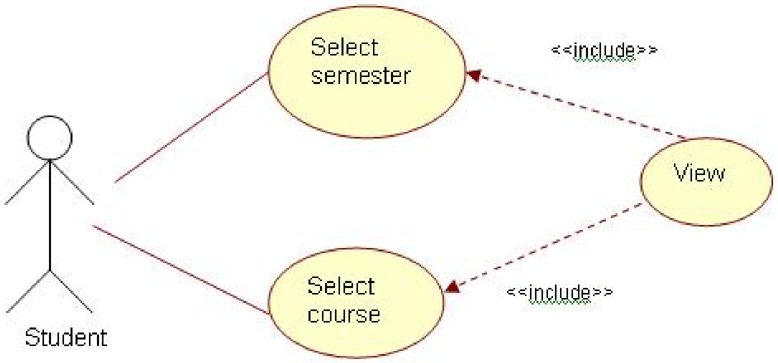


Figure 4.20 Student-View Result Use-Case

4.4.5.2.4 Faculty-View Personal Assessment

Table 4.5 Faculty-View Personal Assessment Description

Primary Actor	Faculty member
Preconditions	The faculty has logged in to the system
Post Conditions	The faculty’s personal assessment report is displayed.
Basic flow	The faculty selects the course for which the assessment is required and then selects the batch to view report
Extensions	The required report is not displayed

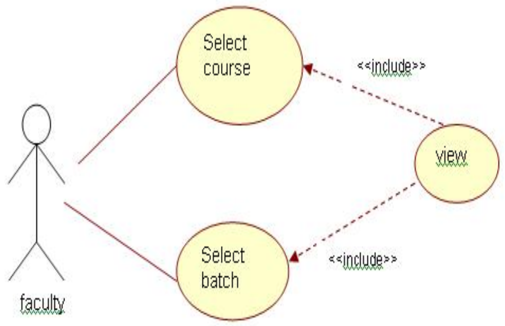


Figure 4.21 Faculty-View Personal Assessment Use-Case

4.4.5.2.5 Faculty-View Student Result

Table 4.6 Faculty-View Student Result Description

Primary Actor	Faculty member
Preconditions	The faculty has logged in to the system
Post Conditions	The student result is displayed
Basic flow	The faculty selects the student whose result is required by name or by batch and then selects the course to view result. The detailed result is then displayed for the selected course.
Extensions	The required result is not displayed

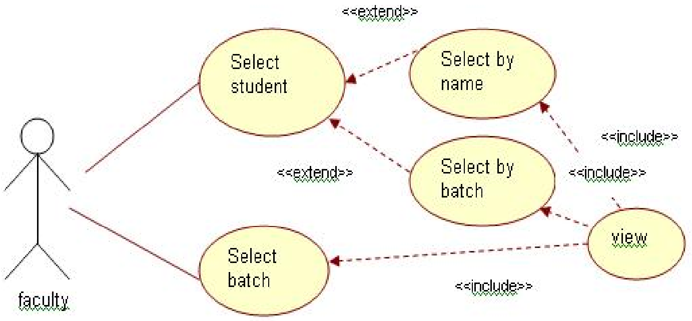


Figure 4.22 Faculty-View Student Result Use-Case

4.4.5.2.6 Faculty - Add/Delete/Update Student Result

Table 4.7 Faculty - Add/Delete/Update Student Result Description

Primary Actor	Faculty member
Preconditions	The faculty has logged in to the system and has opened the student result in editable mode.
Post Conditions	The student result is added, deleted or updated
Basic flow	The faculty selects the course and batch. The detailed result is then displayed for the selected course. Then the result can be updated.
Extensions	The required result is not added, deleted or updated due to system failure.

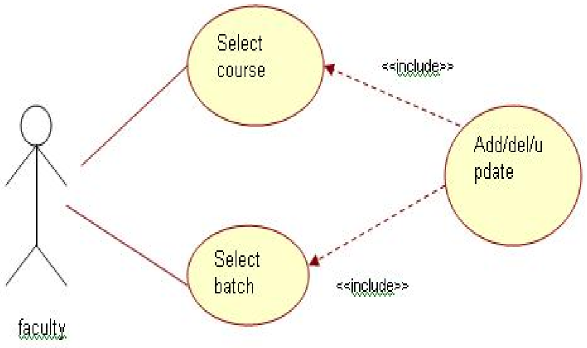


Figure 4.23 Faculty - Add/Delete/Update Student Result Use-Case

4.4.5.2.7 ADM-Upload Information

Table 4.8 ADM-Upload Information Description

Primary Actor	Administration representative
Preconditions	The user has logged in to the system
Post Conditions	The information is displayed on the main page

Basic flow	The administration representative types the information in the text area provided and then uploads it.
Extensions	The information is not displayed on the required page.

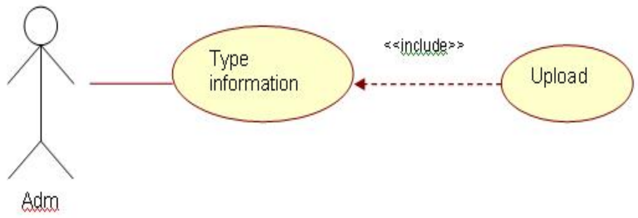


Figure 4.24 ADM-Upload Information Use-Case

4.4.5.2.8 ADM-View Faculty Assessment

Table 4.9 ADM-View Faculty Assessment Description

Primary Actor	Administration representative
Preconditions	The user has logged in to the system
Post Conditions	The required faculty assessment report is displayed
Basic flow	The user selects the faculty member's name for which the assessment report is required and then selects the course to view result. Any batch can also be selected to view faculty's assessment who has taught that batch. The required report is then displayed for the selected course.
Extensions	The required report is not displayed

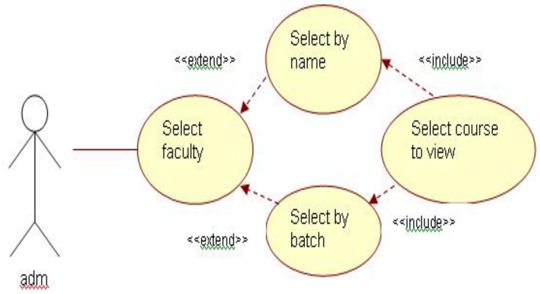


Figure 4.25 ADM-View Faculty Assessment Use-Case

4.4.5.2.9 ADM-View Student Assessment

Table 4.10 ADM-View Student Assessment Description

Primary Actor	Administration representative
Preconditions	The user has logged in to the system
Post Conditions	The required student assessment report is displayed
Basic flow	The user selects the student's name for which the assessment report is required and then selects the course to view result. Any batch can also be selected to view student's assessment. The required report is then displayed for the selected course.
Extensions	The required report is not displayed

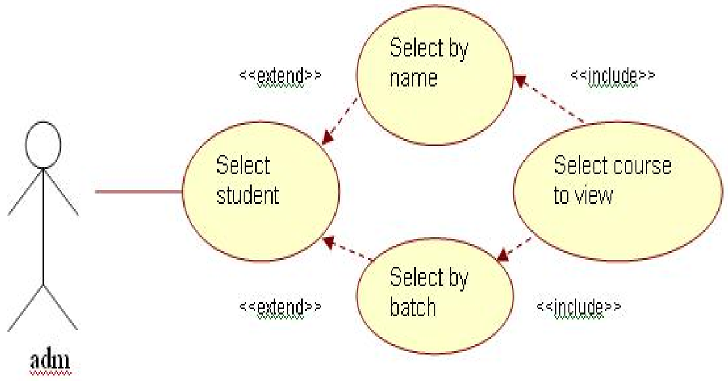


Figure 4.26 ADM-View Student Assessment Use-Case

4.4.5.2.10 Exam Branch-Preview Student Result

Table 4.11 Exam Branch-Preview Student Result Description

Primary Actor	Examination branch representative
Preconditions	The user has logged in to the system.
Post Conditions	The student result is displayed
Basic flow	The user selects the course and batch to preview the results
Extensions	The required result is not displayed due to system failure

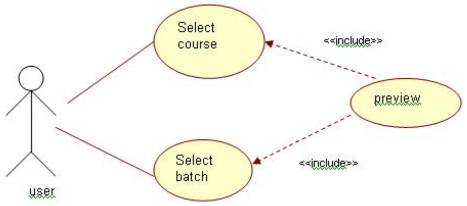


Figure 4.27 Exam Branch-Preview Student Result Use-Case

4.4.5.2.11 Exam Branch-Upload Student Result

Table 4.12 Exam Branch-Upload Student Result Description

Primary Actor	Examination branch representative
Preconditions	The user has logged in to the system.
Post Conditions	The student result is uploaded
Basic flow	The user selects the course and batch to upload the results
Extensions	The required result is not upload due to system failure

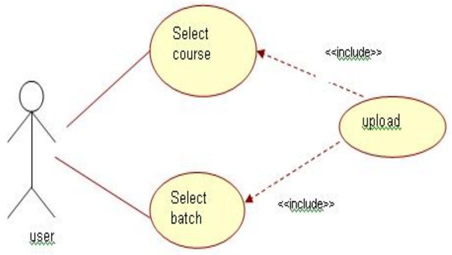


Figure 4.28 Exam Branch-Upload Student Result Use-Case

4.4.6 Sequence Diagram of ACES

The sequence diagrams of ACES are shown as general functions and request information sequence diagram.

4.4.6.1 General Functions Sequence Diagram of ACES

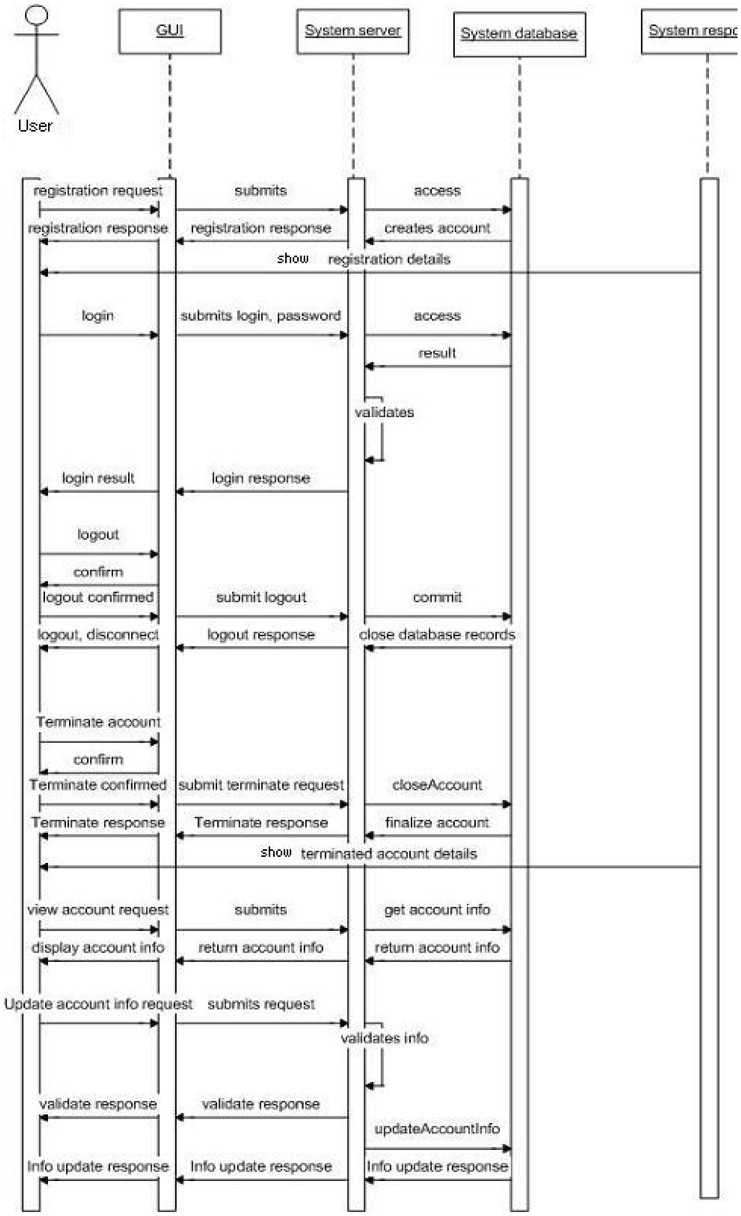


Figure 4.29 General Sequence Diagram

4.4.6.2 Request Information Sequence Diagram of ACES

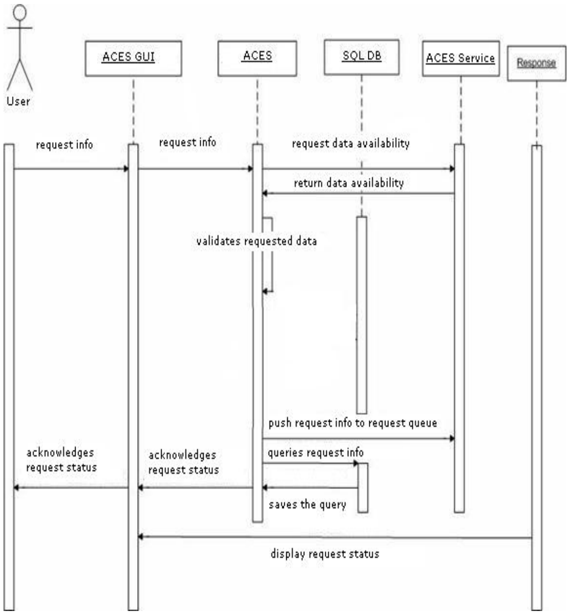


Figure 4.30 Request Information Sequence Diagram

4.4.7 Logical Design of ACES

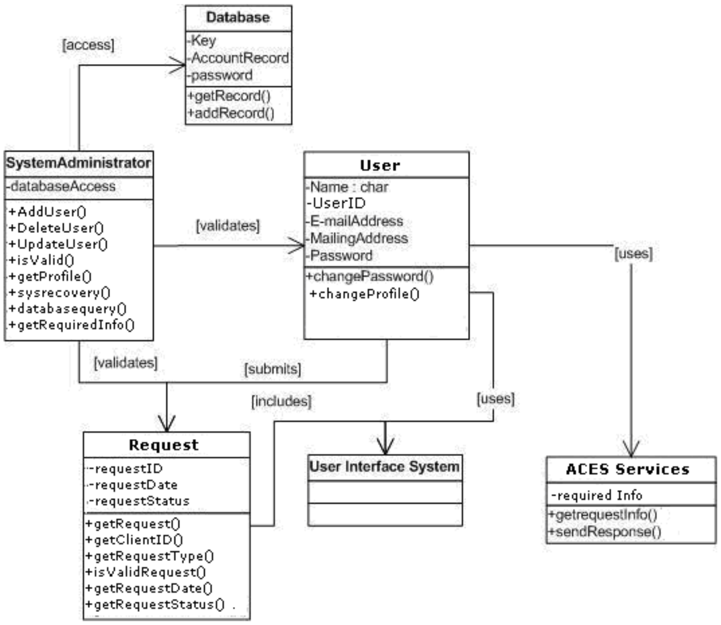


Figure 4.31 Logical Design

4.4.9 Database Design of ACES

Table 4.13 Entities and Attributes

Entity	Attributes
Aces_News	News_id, News_Heading, News_Details, Image_Name, docURL
Course_Eval	Rec_Id, Course_Code, Batch, Practical, Theory, Books, Course contents, Credit_hrs
Courses	Dept_Code, Course_ID, Course, Credit_hourse_Theory, Credit_Hourse_Pract
Courses_Degree_Sem	Rec, Degree, Semester, Course_Code
Departments	Dept_Code, Department
Discipline	Discipline_ID, Discipline_Desc
Instructor_Courses	rec_id, Instructor, Theory_Practival, Degree_Code, Course_Code, Semester, Attendance
Instructor_Factors_Values	recID, Instructor_ID, Course_id, Factor_id, Factor_Value
St_Prev_Qual	Record_No, Qual_ID, Qual, Institution, Marks_Obtained, Total_Marks, Percent, Grade, Div
Student_GPA	RecID
Students_Results	Rec_id, Student_Id, Course, Degree, Semester, Assignment_1, Assignment_2, Assignment_3, Assignment_4, Assignment_5, Assignment_6,Quiz_1, Quiz_2, Quiz_3, Quiz_4, Quiz_5, Quiz_6, Mids, Sessionals_50, Finals, Totals, Sessional_70, Final_30, Total100, Grand_Total, Grade
tCast	CastId, Name
tCategory	CategoryId, Name
tCountry	Country_Code, Country
tDegree	DegreeId, Name
tDiscipline	DisciplineId, Name
tDocuments	DocumentId, Name,

tDomicile	DomicileId, Name
Teachers	Teacher_ID,
Teacher_Course	Rid, Degree, Semester, Teacher_ID, Course_ID, Theory_Practical, Dept_Code
tFactors	Recid,
tGroups	GroupId, Name, MenuContent
tProfessions	Profession_Code, Professions
tQualifications	StudentId, DegreeId, InstituteName, From_To, Obtained Marks, Percentage, Div_Grade
tRanks	RankId, Name
tReligion	ReligionId, Name
tRoles	GroupId, DocumentId
tSect	SectId, Name
tStudents	StudentId, Rank_No, Name, CategoryId, BloodGp, DisciplineId,
tUsers	UserId

4.4.10 Class Diagram of ACES

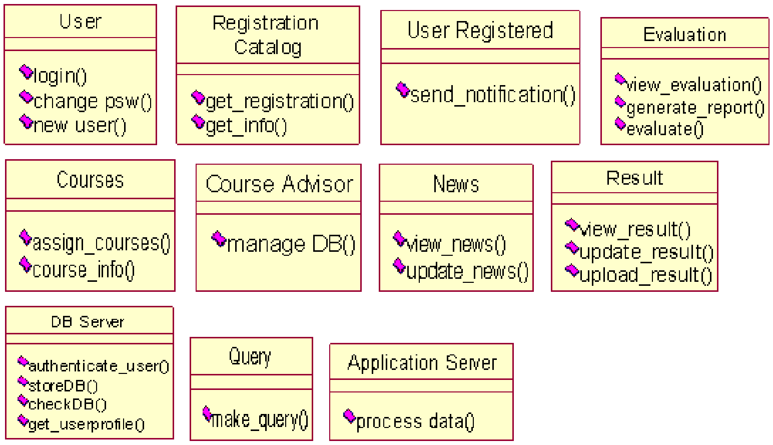


Figure 4.32 Class Diagram

4.5 Mathematical Model for Faculty Evaluation

The core mathematical processing for faculty evaluation is depicted in the sub sections ahead.

4.5.1 Basis of Faculty Evaluation Decisions

Faculty evaluation decisions may require deliberate analysis of student and faculty performance. It is necessary to have quantifying factors to make appropriate faculty evaluation decisions on one hand; while on the other hand, there is a need to see the results of faculty evaluation decisions by taking measurements.

Data sitting in databases are of little use unless they are processed, converted and expressed in intelligent ways to present decision-makers with valuable information. It is time consuming to make statistics gathering or detailed analysis of the available data.

Academic administrators and departments need timely and accurate information on student performance and faculty educational base for decision making. Academic decisions may have far reaching consequences on student and faculty performance.

4.5.2 Materials and Methods

The assessment of faculty members is based on different factors of evaluation criteria. The three major factors are teaching, research and service. These are further divided into sub-factors which are listed in Table 4.13, 4.14, 4.15 [7].

Table 4.14 Teaching Factors

Teaching

a)	Depth of subject Knowledge.
b)	Ability to design and organize the course.
c)	Effective delegating skills.
d)	Ability to stimulate thought and enquiry among the students.
e)	Interaction with the students both inside and outside classrooms.
f)	Punctuality and regularity.
g)	Adherence to high standards of professionalism.
h)	Average class result.
i)	Course content covered.
j)	Effective communication skills.
k)	Class strength
l)	Student attendance

Table 4.15 Research Factors

Research	
a)	Quality and quantity of published research articles and books
b)	Number of papers presented at professional meetings
c)	The number and nature of awards/honors and professional appointments
d)	Qualifications

Table 4.16 Service Factors

Service	
a)	Participation in departmental and university governance through committees, boards, councils etc.
b)	Involvement in student research.

4.5.3 Weightings Distribution

The weightings assigned to teaching, research and service are different for permanent and visiting faculty members. These are shown in Table 4.16 [7]:

Table 4.17 Weightings

Faculty	Permanent	Visiting

Status	faculty (%)	faculty (%)
Factors		
Teaching	40	45
Research	40	45
service	20	10

Further the weightings are assigned to the sub-factors which are separate for faculty members on the basis of qualification i.e. MS or PhD and the average attendance (A) of the class for which the faculty is being evaluated.

The weightings for sub-factors in Teaching, Research and Service respectively are shown in Table 4.17, 4.18, 4.19 whose percentages values highlighted are out off 100.

Table 4.18 Teaching Sub-Factors Weightings

Check	MS,	MS,	PhD,	PhD,
Factors	A<75%	A>75%	A<75%	A>75%
a)	8	8	15	15
b)	10	10	5	5
c)	15	13	15	12
d)	5	2	5	3
e)	7	7	9	9
f)	10	10	8	8
g)	7	7	5	5
h)	10	10	10	10
i)	10	10	10	10
j)	10	10	10	10
k)	3	3	3	3
l)	5	10	5	10

Table 4.19 Research Sub-Factors Weightings

Check	MS	PhD
Factors		

a)	25	35
b)	15	10
c)	30	30
d)	30	25

Table 4.20 Service Sub-Factors Weightings

Check	MS	PhD
Factors		
a)	35	25
b)	65	75

On the basis of above factors and their weightings formulae are derived for calculating an absolute value that ranges from 0 to 1 as a representation of faculty's evaluation. These formulae are formulated separately for permanent and visiting faculty. The factor t in the set of formulae depicts the value of evaluation factors taken from data. They are:

4.5.4 Permanent Faculty

For Qualification = MS and Class Average Attendance<75%

$$F(t)=3.2(t)+4(t)+6(t)+2(t)+2.8(t)+4(t)+2.8(t)+4(t)+4(t)+4(t)+1.2(t)+2(t)+10(t)+6(t)+12(t)+12(t)+7(t)+13(t) \tag{1}$$

For Qualification = MS and Class Average Attendance>75%

$$F(t)=3.2(t)+4(t)+5.2(t)+0.8(t)+2.8(t)+4(t)+2.8(t)+4(t)+4(t)+4(t)+1.2(t)+4(t)+10(t)+6(t)+12(t)+12(t)+7(t)+13(t) \tag{2}$$

For Qualification = PhD and Class Average Attendance<75%

$$F(t)=6(t)+2(t)+6(t)+2(t)+3.6(t)+3.2(t)+2(t)+4(t)+4(t)+4(t)+1.2(t)+2(t)+14(t)+4(t)+12(t)+10(t)+5(t)+15(t) \tag{3}$$

For Qualification = PhD and Class Average Attendance>75%

$$F(t)=6(t)+2(t)+4.8(t)+1.2(t)+3.6(t)+3.2(t)+2(t)+4(t)+4(t)+4(t)+1.2(t)+4(t)+14(t)+4(t)+12(t)+10(t)+5(t)+15(t) \qquad (4)$$

4.5.5 Visiting Faculty

For Qualification = MS and Class Average Attendance<75%

$$F(t)=3.6(t)+4.5(t)+6.75(t)+2.25(t)+3.15(t)+4.5(t)+3.15(t)+4.5(t)+4.5(t)+4.5(t)+1.35(t)+2.25(t)+11.25(t)+6.75(t)+13.5(t)+13.5(t)+3.5(t)+6.5(t) \qquad (5)$$

For Qualification = MS and Class Average Attendance>75%

$$F(t)=3.6(t)+4.5(t)+5.85(t)+0.9(t)+3.15(t)+4.5(t)+3.15(t)+4.5(t)+4.5(t)+4.5(t)+1.35(t)+4.5(t)+11.25(t)+6.75(t)+13.5(t)+13.5(t)+3.5(t)+6.5(t) \qquad (6)$$

For Qualification = PhD and Class Average Attendance<75%

$$F(t)=6.75(t)+2.25(t)+6.75(t)+2.25(t)+4.05(t)+3.6(t)+2.25(t)+4.5(t)+4.5(t)+4.5(t)+1.35(t)+2.25(t)+15.75(t)+4.5(t)+13.5(t)+11.25(t)+2.5(t)+7.5(t) \qquad (7)$$

For Qualification = PhD and Class Average Attendance>75%

$$F(t)=6.75(t)+2.25(t)+5.4(t)+1.35(t)+4.05(t)+3.6(t)+2.25(t)+4.5(t)+4.5(t)+4.5(t)+1.35(t)+4.5(t)+15.75(t)+4.5(t)+13.5(t)+11.25(t)+2.5(t)+7.5(t) \qquad (8)$$

The final value calculated from these formulas will range from 0 to 1.This is the normalized value for the faculty's evaluation which will be taken in one of the four ranges which are illustrated in Table 4.20.

Table 4.21 Normalized Value's Range

Normalized Value's Range	Result
F(t)>=0 && F(t)<=0.25	Poor performance
F(t)>=0.25 && F(t)<=0.50	Average performance
F(t)>=0.50 &&	Good

F(t)<=0.75	performance
F(t)>=0.75 && F(t)<=1.00	Excellent performance

On the basis of this final value the faculty member is rated and his evaluation report is generated.

4.6 Conclusion

The chapter elaborated critical and very important ingredients of software development system, related to software model and architecture. It includes the basic activity diagram with the DFD and entity relationship diagram. The state transition diagram, linking it with the logical design which hinges on sequential, class diagram and mathematical model.

Chapter 5

Implementation of ACES

5.1 Introduction

This chapter basically deals with the elaboration of different technologies used in the development of proposed system. The technologies and the supporting softwares involved in the development of Academics Curriculum Evaluation System ranges from server to client applications.

5.2 Development Technologies Used for ACES

The development technologies used and the strategies being followed are:

5.2.1 Internet Information Services (IIS)

Microsoft Internet Information Services (IIS) is generally known as Internet Information Server and has been used for web hosting. It is a set of Internet-based services for servers using Microsoft Windows.

For Windows XP Professional IIS 5.1 version has been used. An overview of IIS is in Figure 4.35.

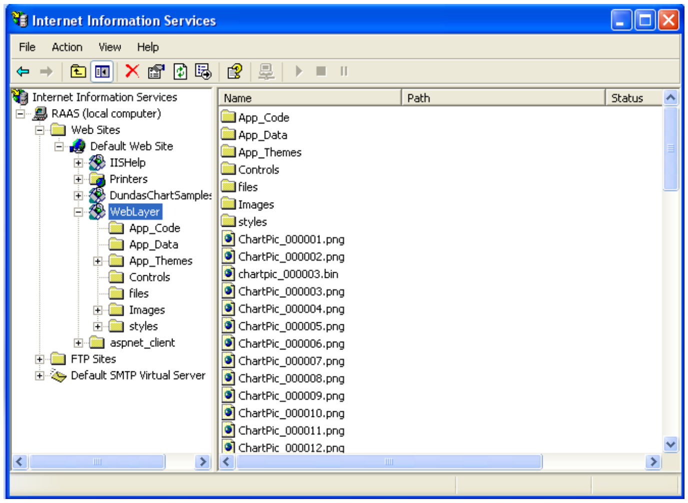


Figure 5.1 IIS

5.2.2 Microsoft SQL Server 2000

Mainly four major features of SQL server have been used which are tables, views, stored procedures and users. SQL server is native backend for visual studio and ASP.Net, as visual studio and ASP.Net are native front end of SQL server. The generalized types of SQL language queries have been used as data retrieval (select statement), data definition (create statement) and data manipulation (update statement).

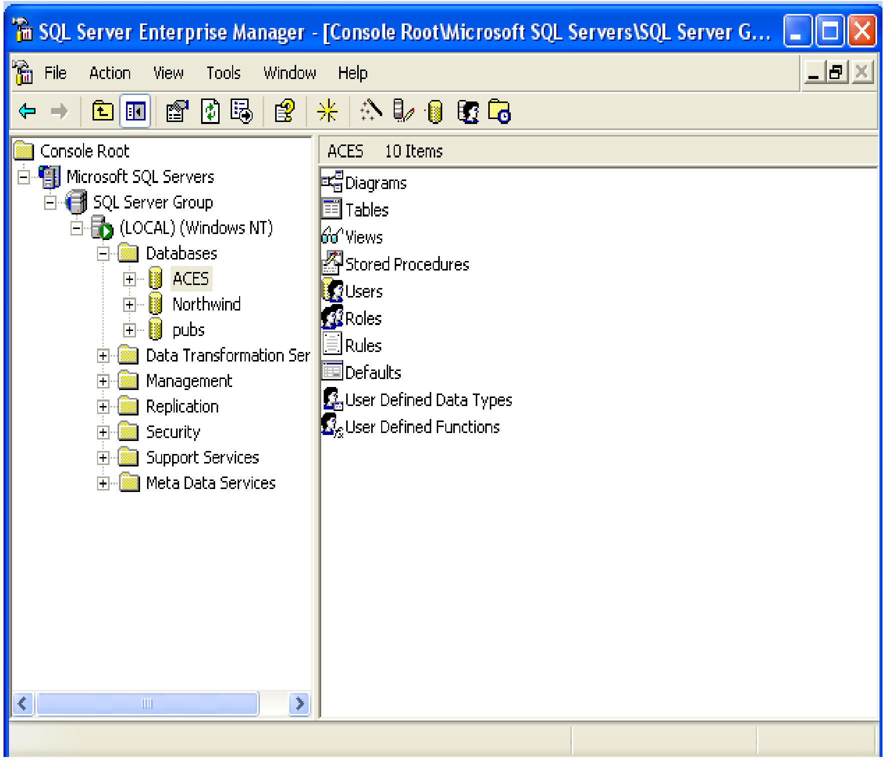
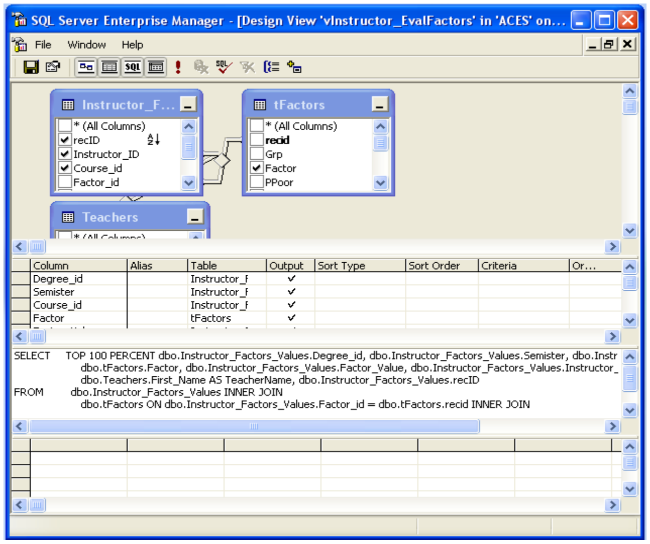


Figure 5.2 SQL Enterprise Manager

5.2.2.1 Views (Database)

Stored queries as views in which select statement have been used. It served as a virtual or logical table containing the result set of a query.



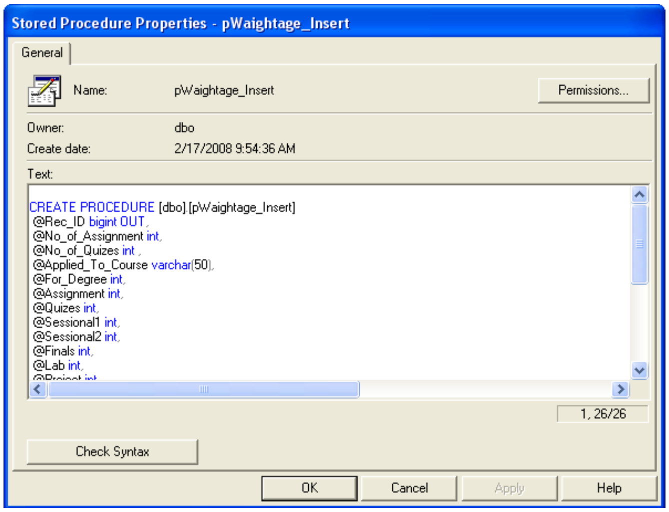
Column	Alias	Table	Output	Sort Type	Sort Order	Criteria	Or...
Degree_id		Instructor_F	✓				
Semister		Instructor_F	✓				
Course_id		Instructor_F	✓				
Factor		tFactors	✓				

```
SELECT    TOP 100 PERCENT dbo.Instructor_Factors_Values.Degree_id, dbo.Instructor_Factors_Values.Semister, dbo.Instr
              dbo.tFactors.Factor, dbo.Instructor_Factors_Values.Factor_Value, dbo.Instructor_Factors_Values.Instructor_
              dbo.Teachers.First_Name AS TeacherName, dbo.Instructor_Factors_Values.recID
FROM        dbo.Instructor_Factors_Values INNER JOIN
              dbo.tFactors ON dbo.Instructor_Factors_Values.Factor_id = dbo.tFactors.recid INNER JOIN
```

Figure 5.3 vInstructor_EvalFactors View

5.2.2.2 Stored Procedures

They are used to provide data manipulation methodologies. We call them as functions and pass parameters to them which are being defined in code. A stored procedure served as a subroutine available to applications accessing a relational database system.



Stored Procedure Properties - pWaightage_Insert

Name: pWaightage_Insert

Owner: dbo

Create date: 2/17/2008 9:54:36 AM

Text:

```
CREATE PROCEDURE [dbo].[pWaightage_Insert]
 @Rec_ID bigint OUT ,
 @No_of_Assignment int,
 @No_of_Quizes int ,
 @Applied_To_Course varchar(50),
 @For_Degree int,
 @Assignment int,
 @Quizes int,
 @Sessional1 int,
 @Sessional2 int,
 @Finals int,
 @Lab int,
 @Project int
```

1, 26/26

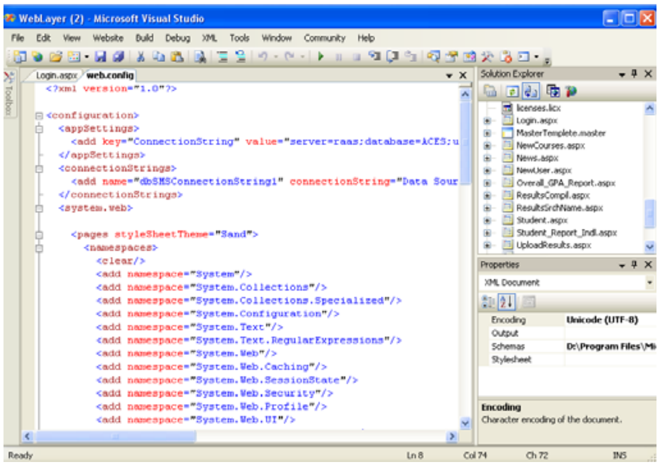
Figure 5.4 pWeightage_Insert Stored Procedure

5.2.3 .Net Framework 2.0

.NET Framework 2.0 has been used to provide Interoperability, Common Runtime Engine, Language Independence, Base Class Library, Simplified Deployment, Security and Portability.

5.2.4 Visual Studio.Net 2005

Microsoft Visual Studio as the main Integrated Development Environment (IDE) has been used. It has been used to develop console and GUI applications along with web sites, web applications, and web services.



```
<?xml version="1.0"?>

<configuration>
  <appSettings>
    <add key="ConnectionString" value="server=raas;database=ACES;u
  </appSettings>
  <connectionStrings>
    <add name="dbSMSConnectionString1" connectionString="Data Sour
  </connectionStrings>
  <system.web>

    <pages styleSheetTheme="Sand">
      <namespaces>
        <clear/>
        <add namespace="System"/>
        <add namespace="System.Collections"/>
        <add namespace="System.Collections.Specialized"/>
        <add namespace="System.Configuration"/>
        <add namespace="System.Text"/>
        <add namespace="System.Text.RegularExpressions"/>
        <add namespace="System.Web"/>
        <add namespace="System.Web.Caching"/>
        <add namespace="System.Web.SessionState"/>
        <add namespace="System.Web.Security"/>
        <add namespace="System.Web.Profile"/>
        <add namespace="System.Web.UI"/>
```

Encoding	Unicode (UTF-8)
Output	
Schemas	D:\Program Files\Mi
Stylesheet	

Encoding
Character encoding of the document.

Figure 5.5 Visual Studio 2005

5.2.4.1 ASP.Net

ASP.NET is a web application framework developed by Microsoft that facilitates programmers to build dynamic web sites, web applications and XML web services. It is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime, allowing programmers to write ASP.NET code using any Microsoft .NET language. The ASPX (ASP.Net files) and other resource files are placed in a virtual host on an Internet Information Services. Mostly applications are developed using an event-driven GUI paradigm (event-driven GUI model.

5.2.4.2 VB.Net

VB.Net was introduced with Visual Studio .NET (2002). It is meant for Rapid Application Development. Visual Basic can be used to develop both console applications as well as GUI applications. It supports the Visual Studio Class designer, Forms designer, and Data designer. VB.NET compiler also is available as a part of .NET Framework but the language services that let VB.NET projects be developed with Visual Studio.

5.2.4.3 Views and Multi-Views

The MultiView and View controls are new addition to ASP.NET 2.0. They are replacement of Tab Control in ASP.NET. A View control works as a container of common controls and a MutliView control works as a container of several View controls.

5.2.4.4 Master Page Technology

In master page technology there is one layout for all other pages. Child pages inherit attributes of master page. This technology has been used to maintain overall symmetry.

5.2.5 Additional Controls

These are basically third party controls being used to provide additional functionality.

5.2.5.1 Dundas Charts

Dundas Chart for ASP.NET is the perfect solution for developers looking to add advanced, feature rich, visually appealing charts to their ASP.NET applications. It consists of the Chart web control, numerous sample applications and solutions, and extensive documentation. The Chart control is an ASP.NET web-server control that lets .NET developers add robust charting abilities to their applications with a minimum amount of effort. It is a fully-managed .NET component, and has been specifically designed for use with Visual Studio .NET. The Dundas Chart comes with a wizard that allows for the quick and easy creation of charts. The Dundas Chart is used to quickly retrieve and bind your database data to your Chart without the need to write code. This tool is being used to achieve the charting solution of evaluation and assessment.

5.2.5.2 Obout Suite

It is being used to provide control for news and events module developed in project. It is also used to provide menu styling option as well as the main login page slide show is also created using this tool.

5.2.6 Coding Strategy

To maintain enhanced coding style two separate files is maintained for each web page, one with aspx extension containing the design part of the web pages and the other one with aspx.vb extension containing the code behind the design.

Event driven programming is being used mainly adopted for user interface development. For database data access layer (DAL) is used to link front end and stored procedures.

5.2.7 Role Based Access

Role based access is implemented in our system. There are two things authentication and authorization. For authentication form based authentication is used and for authorization there are TRoles and Tdoc tables.

5.2.8 Connection Method

In order to connect front end to back end Active X Data Object (ADO.Net) is used. It contains different object with connection from database as command object, data set, data adapter and data connection.

5.3 Conclusion

This chapter walks through the development strategies and coding methodologies used for Academics Curriculum Evaluation System. It highlights the role of internet information services, .Net framework, SQL server, visual studio ASP.Net and VB.Net with focusing on additional controls. Hence covering the complete range of software technology in vogue in the present day world of IT technologies.

Chapter 6

Testing

6.1 Introduction

Testing is extremely important, both to ensure that the system meets requirements, and to ensure that it is free of errors. That is why it is performed throughout the development process at every level.

6.2 Unit Testing

Unit testing ensures that each unit works as expected before the units are integrated. Black box testing provides us with the best method of testing because it enables to ensure that the inputs and outputs to each unit are properly handled.

Some key functions which must undergo unit testing are login, creating profile, viewing of result by students and faculty and updating and uploading of result.

6.2.1 Login

Table 6.1 shows inputs and expected outputs for login.

Table 6.1 Login

Inputs:	Username and password
Expected Outputs:	1) Main page appears 2) Error message is displayed and login screen reappears

6.2.2 Create Profile

Table 6.2 shows inputs and expected outputs for Create Profile.

Table 6.2 Create Profile

Inputs:	Values of all fields e.g. name, batch, department, date of birth etc.
Expected Outputs:	1) The profile is created successfully 2) Appropriate error messages are displayed

6.2.3 Student-View Result

Table 6.3 shows inputs and expected outputs for Student-View Result.

Table 6.3 Student-View Result

Inputs:	Student name, batch, course and semester are selected
Expected Outputs:	1) Detailed student result for selected course is displayed

6.2.4 Faculty-View Personal Assessment

Table 6.4 shows inputs and expected outputs for Faculty-View Personal Assessment.

Table 6.4 Faculty-View Personal Assessment

Inputs:	Faculty name, course taught, semester and batch
Expected Outputs:	1) Faculty assessment report

6.2.5 Faculty-View Student Result

Table 6.5 shows inputs and expected outputs for Faculty-View Student Result.

Table 6.5 Faculty-View Student Result

Inputs:	Student name, batch and course
Expected Outputs:	1) Selected student's detailed result sheet

6.2.6 Faculty-Add/Delete/Update Student Result

Table 6.6 shows inputs and expected outputs for Faculty-Add/Delete/Update Student Result.

Table 6.6 Faculty-Add/Delete/Update Student Result

Inputs:	Student name, course and batch
Expected Outputs:	1) Student result added successfully 2) Student result deleted successfully 3) Student result updated successfully

6.2.7 Administration-View Faculty Assessment

Table 6.7 shows inputs and expected outputs for Administration-View Faculty Assessment.

Table 6.7 Administration-View Faculty Assessment

Inputs:	Faculty name, course and batch taught
Expected Outputs:	1) Selected faculty's detailed assessment report

6.2.8 Administration-View Student Result

Table 6.8 shows inputs and expected outputs for Administration-View Student Result.

Table 6.8 Administration-View Student Result

Inputs:	Student name, batch and course
Expected Outputs:	1) Selected student's detailed result

6.2.9 Administration-Upload Info

Table 6.9 shows inputs and expected outputs for Administration-Upload Info.

Table 6.9 Administration-Upload Info

Inputs:	Text to be uploaded
Expected Outputs:	1) Information uploaded successfully

6.2.10 Examination Branch-Upload Student Result

Table 6.10 shows inputs and expected outputs for Examination Branch-Upload Student Result.

Table 6.10 Examination Branch-Upload Student Result

Inputs:	Batch and course
Expected Outputs:	1) Result uploaded successfully

6.2.11 Examination Branch-Preview Student Result

Table 6.11 shows inputs and expected outputs for Examination Branch-Preview Student Result.

Table 6.11 Examination Branch-Preview Student Result

Inputs:	Student name, batch and course
Expected Outputs:	1) Detailed student result displayed

6.3 Conclusion

This testing process helps to verify that the system is working properly and is according to the requirements. Due to its great significance testing is performed in a separate phase in the development of software systems. In order to make the system error proof detailed testing should be applied so that no portion of the system is left unverified.

Chapter 7

Limitations & Future Work

7.1 Introduction

In any system there are some limitations and boundaries of its performance. The system is confined in terms of its efficiency such as number of users, portability etc. Some features which add to the usefulness of the system may not be included in the current system but are mentioned as the future work and incorporated in the later versions of the system.

7.2 Limitations to the Project

Limitations to Academics Curriculum Evaluation System include its limited functionality as it does not cover all areas of management activities of an institute. It also does not allow the students to enter feedback.

7.3 Future Work

These are the specifications which are not provided for now in the current version of Academics Curriculum Evaluation System but which could be incorporated into future versions. Some of these need advanced technologies and interfaces with other systems.

The Academics Curriculum Evaluation System could be designed in future to enhance the existing capabilities or add entirely new ones. Future work may include enhancements such as the system shall be integrated with the current system MIMS which is working in Military College of Signals.

Additional modules as examination system evaluation can be developed and integrated with Academics Curriculum Evaluation System.

7.4 Conclusion

This chapter is a brief description of the features which are not currently implemented in Academics Curriculum Evaluation System but can be included in future to improve its capacity and make the system more efficient. Further other academics management and curriculum monitoring aspects can also be incorporated as well.

REFERENCES

[1] Java Server Pages, http://en.wikipedia.org/wiki/JavaServer_Pages

[2] ASP.Net, http://en.wikipedia.org/wiki/ASP.NET

[3] *Hypothesis Testing* Statistical Analysis for Decision Making, Morris Hamberg, ISBN: 0-15-583760-5.

[4] *Chi-Square Test,* http://www2.lv.psu.edu/jxm57/irp/chisquar.html

[5] *SPSS*, http://www.accc.uic.edu/Statistical Tools of the Trade.htm

[6] *Stata,* http://www.stata.com/whystata/

[7] Criteria And Procedures For Faculty Evaluation In The Department of Zoology, University Of Oklahoma.

BIBLIOGRAPHY

Evaluation of Faculty at Chadron State College. Joyce Phillips Hardy, Senior Vice President for Academic and Student Affairs Presented to President Thomas L. Krepel on 15 January 2003.

Developing a Comprehensive Faculty Evaluation System. Anker Publishing Company, Inc. Bolton MA.

Practical Decisions in Developing and Operating a Faculty Evaluation System. IN: Michael Theall and Jennifer Franklin, eds. Student Ratings of Instruction: Issues for Improving Practice. Jossey-Bass. San Francisco.

Faculty Evaluation Criteria. Approved FS 04/22/05; Signed off by President, VPAA, VPSS 05-03-05.

Evaluating Faculty Performance. Jossey-Bass. San Francisco. New Directions for Institutional Research.

An Academic Decision-Support System Based on Academic Performance Evaluation for Student and Program Assessment, Dervis Z. Deniz and Ibrahim Ersan.

Criteria And Procedures For Faculty Evaluation In The Department of Zoology, University Of Oklahoma.

Statistical Analysis for Decision Making, Morris Hamberg, ISBN: 0-15-583760-5.

Designing Microsoft ASP.NET Applications By Douglas J. Reilly, ISBN: 0-7356-1348-6.

Deitel – VB.NET How to Program, 2nd Edition*.*

ASP.NET Web Developer’s Guide, PUBLISHED BY Syngress Publishing, Inc. 800 Hingham Street Rockland, MA 02370

ISBN: 1-928994-51-2

Syngress - VB.NET Developer's Guide, PUBLISHED BY Syngress Publishing, Inc. 800 Hingham Street Rockland, MA 02370

ISBN: 1-928994-48-2

Wrox - Professional ADO.NET Programming

An Introduction to Java Servlets by Hans Bergsten