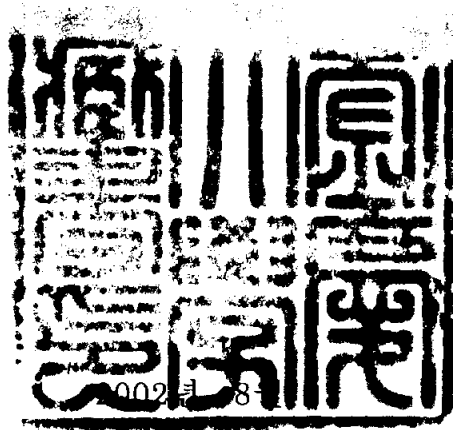


경영학석사 학위논문

.NET을 이용한 학습평가 시스템의 개발에 관한 연구

지도교수 염 창 선

이 논문을 경영학석사 학위논문으로 제출함



부경대학교 경영대학원

경영학과(정보관리전공)

정 수 현

정수현의 경영학석사 학위논문을 인준함

2002년 6월

주심 공학박사 한재호



위원 경영학박사 이현규



위원 경영정보공학박사 염창선



목 차

제 1 장 서론.....	1
제 1 절 연구의 배경 및 목적.....	1
제 2 절 연구의 방법 및 구성.....	2
제 2 장 온라인 교육 및 학습평가 시스템에 대한 이론적 고찰.....	4
제 1 절 온라인 교육의 개념.....	4
제 2 절 학습평가 시스템의 개념.....	5
제 3 절 학습평가 시스템에 관련된 기존 연구.....	8
제 3 장 .NET을 이용한 학습평가 시스템의 구현을 위한 요소기술.....	13
제 1 절 .NET의 개념.....	13
제 2 절 .NET의 동작원리.....	15
제 3 절 .NET의 특성.....	24
제 4 절 SQL 서버.....	25
제 4 장 .NET을 이용한 학습평가 시스템의 설계.....	29
제 1 절 시스템 개발 환경.....	29
제 2 절 학습평가 시스템의 구성 및 기능.....	30
제 3 절 .NET을 이용한 학습평가 시스템의 모듈별 구성.....	35
제 4 절 .NET을 이용한 학습평가 시스템의 흐름도.....	38
제 5 절 .NET을 이용한 학습평가 시스템의 데이터베이스 설계.....	42

제 5 장 .NET을 이용한 학습평가 시스템의 구현	45
제 1 절 관리자 메뉴	46
제 2 절 문제 출제자 메뉴	52
제 3 절 학생 메뉴	58
제 4 절 .NET과 ASP를 이용한 학습평가 시스템의 비교	64
 제 6 장 결 론	 67
 참고문헌	 69

표 목 차

[표 3-1] J2EE와 .NET의 사용자 응답 시간	24
[표 3-2] J2EE와 .NET의 CPU사용량 비교	25
[표 4-1] 관리자 모듈 구성표	36
[표 4-2] 학생 모듈 구성표	36
[표 4-3] 출제자 모듈 구성표	37
[표 4-4] 학습평가 시스템의 테이블 목록	42
[표 5-1] ASP와 .NET의 CPU사용량 비교	66
[표 5-2] ASP와 .NET의 사용자 응답 시간	66

그 립 목 차

[그림 3-1] .NET의 구조.....	14
[그림 3-2] CLR의 구성.....	17
[그림 3-3] .NET 코드의 실행 구조.....	18
[그림 3-4] 구체적인 실행 모델.....	19
[그림 3-5] ASP.NET의 실행 구조.....	20
[그림 3-6] Cache에 저장된 aspx 실행 구조.....	21
[그림 3-7] ASP.NET의 개발 흐름도.....	23
[그림 3-8] SQL 서버 관계형 엔진 개요.....	27
[그림 3-9] 데이터베이스 저장소 엔진 개요.....	28
[그림 4-1] 학습평가 시스템의 하드웨어 구성	32
[그림 4-2] 학습평가 시스템의 구조.....	33
[그림 4-3] .NET을 이용한 평가시스템의 계층 구조.....	34
[그림 4-4] 관리자 모듈 흐름도.....	39
[그림 4-5] 학생 모듈 흐름도	40
[그림 4-6] 출제자 모듈 흐름도.....	41
[그림 4-7] .NET을 이용한 학습평가 시스템의 ERD.....	43
[그림 5-1] 학습평가 시스템의 메인 화면.....	45
[그림 5-2] 출제자 관리 화면	46
[그림 5-3] 학생 리스트 화면	47
[그림 5-4] 성적관리 화면.....	48

[그림 5-5] Login 하는 C#, ASP.NET 코드.....	48
[그림 5-6] Login 후 다른 메뉴 생성을 위한 C#, ASP.NET 코드.....	50
[그림 5-7] 과목등록 화면.....	53
[그림 5-8] 학생등록 화면.....	53
[그림 5-9] 시험 출제시 설정사항 등록 화면.....	54
[그림 5-10] 시험 출제를 위한 C#, ASP.NET 코드.....	55
[그림 5-11] 성적처리를 위한 C#, ASP.NET 코드.....	57
[그림 5-12] 수강과목 열람 화면.....	59
[그림 5-13] 시험응시 화면.....	60
[그림 5-14] 시험응시를 위한 C#, ASP.NET 코드.....	60
[그림 5-15] 채점을 위한 C#, ASP.NET 코드.....	62
[그림 5-16] 채점결과를 DB에 저장하기 위한 C#, ASP.NET 코드.....	63
[그림 5-17] .NET과 ASP 학습평가시스템 비교를 위한 테스트 환경 구성도.....	65

A Study on the Development of Learning Test System Using .NET

Jeong Su-Hyun

*Department of Information Management,
Graduate School of Business Administration,
Pukyong National University*

Abstract

The purpose of this study is to improve the performance of existing Learning Test System. Existing Learning Test Systems are unstable and have a slow response time when many users use the systems simultaneously.

To overcome these problems, this study aims to design and implement the Learning Test System using .NET which is developed by MicroSoft. The Learning Test System is based on the Web, consists of administrator module, questioner module and student module. The functions of the system, i.e., providing test questions, performing test and evaluating result of test are achieving on the web in real time.

To performance test, .NET system and ASP system were compared

on the same server platform environment. The result showed that .NET system has better response time and stability than ASP system.

제 1 장 서 론

제 1 절 연구의 배경 및 목적

온라인 교육(e-Learning)이란 인터넷과 테크놀로지를 이용하여 학습을 수행하는 것이라고 정의될 수 있다.¹ 기존의 교육이 교사 중심으로 지식과 정보가 주로 학생들에게 획일적, 일방적으로 전달되고 의사소통의 창구가 있더라도 대부분 제한적인 데 반해, 온라인 교육은 학습자-내용간, 학습자-교수자간, 학습자-학습자간의 다차원적이며 양방향적인 상호작용을 가능하게 한다.² 온라인 교육은 e러닝, 혹은 원격 교육 등의 다양한 말로도 표현된다. 현재 많은 대학과 교육기관, 그리고 기업의 교육부서에서는 각종 교육시스템을 온라인으로 구축하여 운영 중에 있다. 현재 가상대학의 매출은 연간 120 억원 정도이며 또한 일반 오프라인 대학들의 온라인 교육서비스 규모는 200억원 정도이다. 기업체 시장은 더욱 커서 올해 기업 온라인교육시장의 규모는 600억원 정도로 예상되고 있다.

지금까지의 온라인 교육분야에 대한 연구는 부분 학습내용의 전달을 위한 시스템 개발에 치우쳐 왔다. 그러나 최근에 온라인 교육 시스템에서의 학습을 어떻게 평가할 것인가도 중요한 문제로 대두되고 있다. 온라인 교육 시스템은 교육전달 시스템과 학습평가 시스템으로 구분할 수 있다. 현재 운용되고 있는 학습평가 시스템은 주

¹ 김덕중,김연주 “e-learning 기획실무 스타일가이드”,2002

² 남윤희, “원격 교육을 위한 평가시스템 설계 및 구현”, 석사학위논문, 신라대학교, 2001.8.

로 Web을 기반으로 하여 객관식이나 주관식을 나열하여 응시자의 시험 결과를 평가하는 방식이다. 학습평가 시스템은 교육전달 시스템보다 안정적이고 뛰어난 반응 속도를 제공하여야 한다. 교육전달 시스템은 일반적으로 웹을 이용한 실시간 스트리밍 시스템과 교육 콘텐츠를 파일에 저장한 다음 이 파일을 학습자가 다운로드하여 특정한 프로그램에서 구동한 후 학습하는 시스템이 있다. 다운로드하여 사용하는 방법은 콘텐츠 도용의 우려가 높기 때문에 요즘의 추세는 웹을 이용하여 실시간 스트리밍으로 사용하는 교육전달 시스템이 널리 사용되고 있다. 학습평가 시스템은 실시간 평가와 비실시간 평가 방법이 있는데, 실시간 평가 방법은 시스템의 안정성이 중요시 된다. 최근의 학습평가 시스템은 실시간 평가방법으로 구축되고 있다. 실시간 학습평가 시스템은 다수의 학습자가 사용할 때 시스템에 과부하가 발생하면서 속도가 느려지거나, 접속이 끊기는 등의 문제가 발생하고 있다.

그러므로 본 연구는 위에서 언급한 기존의 실시간 학습평가 시스템의 문제를 해결하기 위한 방안으로 .NET을 이용한 학습평가 시스템을 개발하고자 한다. .NET을 이용하여 개발되는 학습평가 시스템은 다수 학습자에게 빠른 속도와 안정적인 사용을 보장하여 줄 것이다.

제 2 절 연구의 방법 및 구성

본 연구는 .NET을 이용한 학습평가 시스템을 구축하는 연구로서 마이크로소프트사의 최신 웹서비스 구현 기술인 C# 언어와

ASP.NET으로 구현하여 웹에서의 빠른 응답률과 안정성을 가능하게 한다. 또한 완전 한글화가 되어 한글과 완벽한 호환성을 가진 마이크로소프트사의 SQL 서버 2000을 DBMS로 사용한다. 시스템의 사용 주체는 시스템을 관리하는 관리자, 문제를 출제하는 출제자, 시험에 응시하는 학생 등으로 구분할 수 있다.

본 연구에서 이루어진 학습평가 시스템을 이용하면 관리자는 출제자(교수)에게 아이디 및 암호를 부여하여 등록하게 된다. 그리고 출제자는 자신에게 부여된 아이디와 암호를 이용하여 시스템에 로그인하고 자신이 강의하는 과목을 등록한다. 과목을 등록할 때는 여러 가지 기본적인 과목명, 설명, 강의 일정등과 함께 출석률, 리포트 성적 비율, 중간고사 비율, 기말고사 비율을 설정할 수 있다. 그리고 자신의 과목을 수강할 수 있는 학생을 등록한다. 학생은 자신에게 부여된 아이디와 암호를 이용하여 로그인하게 되고 정해진 시험일시가 되면 출제자가 등록한 시험을 응시하여 평가 받게 된다.

본 연구의 제2장에서는 온라인 교육 및 학습평가시스템에 대한 이론적 고찰을 수행한다. 제3장에서는 학습평가 시스템을 구현하는 데 필요한 요소 기술들에 대해 언급한다. 제4장에서는 .NET을 이용한 학습평가 시스템의 전체구성과 모듈별 구성, 그리고 데이터베이스 구성 환경을 설명한다. 그리고, 제5장에서는 실제 구현에 들어가서 시스템의 핵심코드와 실행화면을 제시한다. 마지막 제6장에서는 본 연구의 결론과 향후에 해결하여야 할 문제를 기술한다.

제 2 장 온라인 교육 및 학습평가 시스템에 대한 이론적 고찰

제 1 절 온라인 교육(e-Learning)의 개념

온라인 교육은 1870년 미국의 일리노이의 한 대학에서 시작된 통신교육이 시초라고 할 수 있다.¹ 통신교육은 당시 교육의 수혜를 입기 어려웠던 사람들에게 교육의 기회를 제공해 주었다. 그러다가 기술의 발달로 라디오, 텔레비전, 인터넷에 이르기까지 제공하는 플랫폼을 바꾸어가며 현재의 인터넷을 이용한 온라인 교육으로 발전해 왔다. 기업들이 온라인 교육을 선호하는 이유를 살펴보면 다음과 같다.

첫째, 기업의 규모가 커짐에 따라 늘어나는 학습수요를 처리하여야 하는데 이러한 많은 학습 수요를 강의실에서 감당하려면 엄청나게 많은 시간이 소요된다.

둘째, 온라인 교육을 이용하면 언제 어디서든 시간과 장소에 구애받지 않고 교육을 받을 수 있다.

셋째, 학습자당 교육 비용으로 계산할 때 온라인 교육은 강의실 교육보다 비용이 적게 든다. 이러한 비용절감은 강사와 학습자를 한 공간에 모아놓기 위한 비용이 절감되기 때문이다.

넷째, 온라인 교육은 그때 그때 서버에 있는 내용만 새롭게 바뀌면 신속하게 새로운 내용을 학습자가 접할 수 있다.

다섯째, 온라인 교육은 학습자에게 진도 관리의 기능을 부여한다.

¹ 김덕중, 김연주 “e-learning 기획실무 스타일가이드”, 2002

즉, 학습자는 자기 스타일에 맞게 진도를 관리할 수 있고, 또한 미진한 부분은 몇 번씩 반복할 수 있게 한다.

여섯째, 보통 학습자들은 질문은 꺼리는 경향이 있는데 그것은 자신의 생각이나 질문이 잘못될 경우를 두려워하기 때문이다. 그러나 온라인 교육은 이러한 문제를 배제하여 주기 때문에 학습자들은 활발하게 질문할 수 있다.

일곱째, 온라인 교육은 피교육자가 원하는 과목을 학습할 수 있게 하기 때문에 피교육자가 중심이 된 개인화된 교육을 가능케 한다.¹

제 2 절 학습평가 시스템의 개념

전기한 것처럼 온라인 교육의 효용성이 갈수록 커짐에 따라 온라인을 이용한 교육뿐만 아니라 이의 효과를 평가하는 온라인 평가(e-test)의 중요성도 갈수록 커지고 있다. 실제로 연구조사 보고에 의하면 온라인 평가의 비중은 전체 온라인 교육시장에서 약 18% 가량을 차지하는 것으로 나타나서 온라인 평가의 비중이 만만치 않음을 나타내고 있다.² 또한 일반이 생각하는 대학이나 초.중.고교와 같은 학교기관 보다는 기업체 재교육시장, 일반인 대상 자격증 및 외국어 시장이 오히려 시장성이 큰 것으로 나타나 온라인 평가에 관한 연구가 앞으로 활성화 될 것임을 암시하고 있다.

온라인 교육에서 학습평가 시스템은 PC를 이용하여 제출자가 입

¹ ㈜아이비즈넷, "인터넷 비즈니스 @i-biznet.com", 2000

² 한국소프트웨어진흥원, "디지털 콘텐츠 산업 조사 연구 사업 연구 보고서", 2000.12

력하고 나면 이를 학생들이 응시하여 평가를 하는 것을 기본으로 하고 있다. 이러한 평가방법에는 비실시간 평가방법과 실시간 평가방법이 있다.

비실시간 평가방법은 E-mail을 이용하거나 FTP를 이용하는 방법이 대표적이다. 우선 E-mail을 이용하는 방법은 학생들이 게시된 과제물이나 문제에 대하여 답을 작성한 후에 담당 교수에게 E-mail을 제출하는 방법으로 이 경우는 담당 교수가 지속적으로 E-mail을 확인하여 첨부물을 확인하는 방식으로 평가해야 하고, 또한 결과물이 정형화되지 않고 제 각각으로 나올 확률이 있다. 결국에는 평가하는 방법이 기존의 오프라인 방식과 크게 다르지 않을 수 있다. FTP를 이용하는 방법으로는 우선 제출자가 FTP 서버를 자체적으로 운영할 수 있는 기술정도를 알아야 하고, 또한 보안권한상의 문제로 다른 사람의 계정에 있는 결과물을 도용할 수 있는 문제가 발생할 수 있다. 물론 일일이 계정을 다르게 발급하면 되겠지만 학생이 많아지면 계정발급상의 부하가 발생할 수도 있다.

실시간 평가방법으로는 웹을 이용하는 방법이 가장 대표적이다. 요즘의 학습평가 시스템은 Web 기반으로 개발되는 것이 많이 연구되고 있다. 그 이유는 웹이 가진 몇 가지 특징점에 기인한다. 그것들은 다음과 같다.

첫째, 대중적인 접근 용이성이다. Web 기반 시스템은 웹 브라우저를 통해 모든 시험응시와 관리가 이루어진다. 따라서 프로그램 사용법을 따로 교육할 필요가 없다.

둘째, 프로그램 설치가 불필요하다는 점이다. 기존의 C/S 환경에서는 반드시 해당 프로그램을 설치하여 구동해야 했지만 Web 기반 환경에서는 시스템에 미리 설치되어 있는 웹 브라우저만 있으면 된

다.

셋째, 업데이트의 용이성이다. 새로운 기능이 추가될 경우 서버에 서만 업데이트 하면 되기 때문에 일일이 클라이언트 프로그램을 업데이트해야 하는 불편함이 없다.

넷째, 데이터보존이 용이하다는 점이다. DBMS를 이용하기 때문에 관련한 DB 데이터만 백업하면 모든 시험 관련 데이터를 보존할 수 있다.

이런 장점을 가진 웹을 이용하는 방법은 또한 문제의 게시만 웹을 이용하고, 문제의 답은 E-mail 이나 FTP를 이용하는 방법과 스크립트를 이용하여 답을 웹 브라우저로 업로드 할 수 있는 방식이 있다. 웹을 이용하여 문제를 게시하고 E-mail 이나 FTP를 이용하여 답을 받는 방식은 앞에서 말한 비실시간 평가방법이 가지는 단점을 그대로 가지게 된다. 그러나 스크립트를 이용하여 웹 브라우저로 답을 올리게 되면 실시간으로 답을 평가할 수가 있어서 문제 출제자의 부담을 많이 줄일 수 있고, 응시자도 원한다면 바로 자신의 평가점수를 확인할 수도 있다. 그러나 스크립트를 개발하는데 비용과 시간이 들고 관리모듈을 따로 제작하여야 하며, 이를 DBMS로 관리해야 하기 때문에 전문지식을 갖춘 개발사에 의뢰하여야 한다는 점을 가지고 있다. 하지만 한번 개발해 놓으면 다른 방식의 단점들을 모두 극복하고 모든 학생들의 시험평가와 모든 제출자들의 제출자료를 한곳에서 일목요연하게 관리하고 보관할 수 있어 이를 피드백 하거나 시간이 지난 후에 다시 볼 수 있는 등의 많은 장점이 있다.

제 3 절 학습평가 시스템에 관련된 기존 연구

3.1 기존의 연구방향

학습평가 시스템에 대한 연구는 Web 기반 학습 시스템이나 e-learning에 비하여 늦게 시작되었다고 볼 수 있다. 온라인 평가는 제 2절에서 기술한 바와 같이 E-mail 이나 FTP를 사용하였던 방식이었지만 웹의 도입에 따라서 웹 기반 학습평가 시스템의 연구가 시작되었다.

학습평가 시스템의 연구가 이루어지기 시작하던 초기에는 웹으로 학습평가를 할 수 있는 자체에 대한 연구가 이루어졌다. 배상현(1998)은 출제자가 웹 브라우저를 통해 문제를 제출하고 이를 학습자가 웹 브라우저로 응시하게 되면 결과를 분석하여 응시자에게 보여주는 연구를 시도하였다. 이것은 Illustra ORDBMS를 이용하여 실시간 평가시스템을 구축하여 강의와 평가를 나누어서 설계한 것이었다. 강의처리부에서는 출제자가 강의계획과 노트를 작성하여 저장하고 평가처리부에서는 학생이 문제를 풀고 평가할 수 있는 식이다. 평가는 실시간 처리 방식이다. 실시간으로 처리하기 때문에 학습자가 빠른 피드백을 받는다는 장점은 있으나, 실제로 행해지는 대학에서의 많은 평가가 주관식도 있다는 점으로 볼 때 주관식에 대한 연구가 미흡하다라고 평가할 수 있다.

그러다가 학습평가가 정규 시험으로만 이루어지는 것이 아니라는 데 주목하면서 정규적인 시험이외의 평가도 할 수 있는 시스템의 연구가 이루어졌다. 노일순(1999)은 학습자의 수업 참여도를 측정하는 퀴즈 형태의 문제를 내는 학습평가 시스템을 연구하였다. 여기서는 시스템을 논리적으로 학습자 기능, 교수 기능, 관리자 기능으로

나누어서 연구하였다. 이 연구에서는 이전에 이루어지던 학습평가 시스템이 주로 CGI나 ASP를 기반으로 했던 것에 비해, 당시로서는 새로운 개념이던 Servlet과 JDBC를 이용하였던 점에 주목할 수 있다. 즉, 평가시스템에 대한 연구가 대규모 동시접속자가 일으키는 시스템의 부하를 줄일 수 있는 프로그래밍쪽으로도 이루어졌다는 점이다. 그러나 과부하를 줄이고 서버의 CPU 점유율을 낮추는 데에 연구를 집중하다 보니, 평가시스템 자체의 확장된 기능이라든지 평가처리부분에는 이전의 연구보다 발전한 점이 별로 없다고 볼 수 있다.

시험문제의 축적과 재활용에 대한 중요성이 커지면서 출제자가 이미 제출했던 문제를 재활용하여 시험을 구성할 수 있게 하는 문제은행식을 활용한 시스템의 연구가 이루어지기 시작했다. 이진경(2000)은 이미 응시했던 문제나 출제자가 자신이 이미 출제했던 문제를 검색, 재활용할 수 있게 피드백에 중점을 둔 연구를 시도하였다. 이 연구에서는 교사는 웹 상에서 문제 출제를 체계적으로 할 수 있으며 학생들이 학습하는 과정에 평가 문제를 해결하고 즉각적인 피드백을 제공받을 수 있게 하였다. 또한 교사가 출제한 문제는 데이터베이스에 문제은행화하여 문제의 출제 및 검색, 관리가 효율적으로 이루어지도록 하였다. 그러나 시스템 설계시 학습자 모듈과 출제자 모듈만을 두어 전체적으로 관리를 할 수 있는 관리자 기능이 미흡했다.

학습평가 시스템이 어느 정도 연구되자 시험자체를 관리, 감독하는 것을 강화하는 연구가 이루어지기 시작했다. 신경아(2001)는 사용자의 키보드 동작을 감시하고 이에 따라 경고

메시지와 시험을 제어할수 있는 기능에 관한 연구를 수행하였다¹. 또 시험 중 응시자와 감독관 사이의 실시간 문답 기능을 지원하도록 하였다. 즉, 평가방법 중에서 일종의 구두 평가를 웹에 적용할 수 있는 방법을 제시하였다.

최근에는 학습평가 시스템의 주관식 문항의 평가에 대한 주안점을 둔 연구도 이루어졌다. 남윤희(2001)는 학습평가 시스템의 평가 영역을 주관식, 객관식 문항을 별도로 관리하여 랜덤출제 할 수 있게 하고 평가 문제를 데이터베이스화 하여 문제은행식으로 출제할 수 있게 하였다. 또한 객관식뿐 아니라 주관식 문항도 평가할 수 있게 하였다. 주관식 문항의 경우 시스템에서 자동으로 처리하는 것이 아니라 주관식에 한해 출제자가 직접 채점을 하고 이를 자동 채점처리된 객관식 성적과 같이 합산할 수 있게 설계하였다. 주관식 문항을 완전 자동처리하자면 한글 자연어 처리 등 많은 문제점이 있을 뿐 아니라 출제자가 직접 채점하는 방식을 선호하기 때문에 주관식의 완전 자동처리는 앞으로 많은 난관이 있을 것으로 보인다.

이제까지 살펴본 연구를 토대로 앞으로의 연구방향을 예측하여 보면 학습평가 시스템의 연구는 웹을 기반으로 하여 발전하고 있으며, 점차 많은 수의 사용자가 Web 기반 학습평가 시스템을 쓰게 됨에 따라 평가시스템의 기능적 연구와 더불어 다수의 사용자가 동시에 접속하여 시스템을 사용하더라도 이상 없이 서비스 할 수 있는 기능적인 면에도 많은 연구가 이루어 질 것으로 보인다.

¹ 신경아, “원격평가시스템의 설계 및 구현” 한국교원대학교 교육대학원 석사학위 논문, 2001.2

3.2 기존의 웹 서버 프로그래밍

웹의 발전과 함께 정보를 웹상에 나타나게 해 주는 웹 프로그래밍도 발전되어 왔다. 웹 프로그래밍은 웹 브라우저 프로그래밍과 웹 서버 프로그래밍으로 나눌 수 있다. 웹 브라우저 프로그래밍은 JavaScript, VB Script, Java Applet 등이 사용되고 있으며, 웹 서버 프로그래밍은 CGI, ASP, Servlet, PHP 등이 사용되고 있다.

CGI는 Unix 기반의 웹 서버에서 많이 사용되고 있다. 이것은 C 언어 등을 이용하여 간편하게 사용할 수 있는 반면에, 실행될 때마다 새로운 프로세스를 형성해야 한다는 단점이 있다. 이 경우에는 동시에 많은 수의 사용자가 접속할 때에 시스템에 과부하를 초래하여 성능이 저하될 수도 있다.

ASP는 서버에서 실행되는 스크립트 언어로서 코딩 할 때는 HTML 언어 사이에 섞어서 코딩을 하게 되지만, 이를 사용자가 웹 브라우저로서 요청하게 되면 웹 서버에서 스크립트를 해석하여 클라이언트에는 HTML 코드로만 반환한다. 이렇게 되면 사용자는 HTML만을 보기 때문에 소스가 드러나는 보안상의 문제를 해결할 수 있다. 또한 Session 개념을 사용하여 CGI처럼 과도하게 프로세스가 형성되는 것을 방지하였다. 그러나 ASP는 언어가 아니라 스크립트이기 때문에 일단 서버에서 인터프리트(해석)되어야만 하고, 다양한 기능을 부여할 수 없어서 부가적인 기능을 원할 경우에는 일일이 다른 언어로 컴포넌트를 제작하여야 한다.

PHP의 경우도 ASP와 마찬가지로 서버에서 실행되는 언어이기 때문에 장단점은 흡사하다. 그러나 Windows 플랫폼에서만 사용될 수 있는 ASP에 비해 Linux, Unix, Windows 등 다양한 플랫폼에서

사용할 수 있다는 점을 가지고 있다.

3.3 .NET의 장점

.NET은 마이크로소프트가 최근 발표한 개발 언어들을 지칭하는 말이자, PC 뿐만 아니라 PDA나 태블릿 PC, 휴대폰 등을 아우르는 멀티 플랫폼의 서비스 전략이라고 할 수 있다. .NET을 사용하게 되면 다음과 같은 장점을 얻을 수 있다.

첫째, 운영시스템과 독립적으로 소프트웨어를 개발하고 실행할 수 있다. Windows 시스템뿐만 아니라 PDA같은 모바일 디바이스에까지 하나의 .NET 어플리케이션을 개발하여 다양한 디바이스에서 실행할 수 있다.

둘째, 프로그래밍 언어에 구애 받지 않는다. 따라서 개발자는 자신이 선호하는 프로그래밍 언어를 이용하여 코딩하고 이를 혼용하여 .NET 어플리케이션으로 만들 수도 있다. 심지어, 자바까지도 2002년 하반기에 나올 예정인 J#을 이용하여 .NET 어플리케이션으로 만들 수 있다.

셋째, 기존의 ASP 방식은 스크립트 언어를 사용하여 인터프리트 하기 때문에 실행 속도가 떨어지고 전체적으로 서버 성능을 떨어뜨릴 수 있었다. 그러나 .NET에서는 시스템에 코드가 컴파일되어 있다가 들어오는 요구에 반응하기 때문에 속도가 훨씬 빠르다.

넷째, 한번 컴파일된 코드는 캐쉬에 저장해서 같은 요구가 들어오면 바로 캐쉬에서 실행되기 때문에 처음 실행될 때는 약간 느릴 수 있지만 두 번째부터는 빠른 속도로 반응하게 된다.

제 3 장 학습평가 시스템의 구현을 위한 요소기술

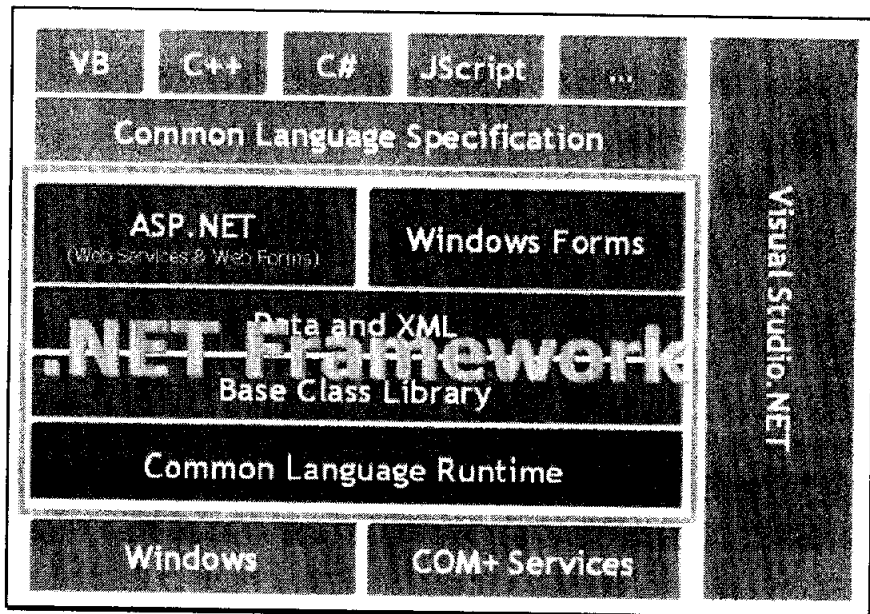
본 연구에서의 온라인 학습평가 시스템은 DBMS로서 Microsoft사의 SQL Server 2000을 사용하고 또한 이를 위한 프로그램 개발 언어로서는 Microsoft사의 최신 웹 서비스 구현 기술인 C#과 ASP.NET을 이용하였다. 또한 문제 입력은 웹을 이용하여 실시간으로 문제 출제자가 입력하면 될 수 있게 하였으며 결과는 곧바로 볼 수도 있고 아니면 출제자의 설정 사항에 따라 일정 시간이 흐른 후에 결과를 응시자가 확인할 수도 있다. 그러므로 이 장에서는 .NET을 이용한 학습평가 시스템 개발을 위해 필요한 .NET의 요소 기술에 대해 살펴본다.

제 1 절 .NET의 개념

.NET이라는 개념은 단순한 프로그래밍 상의 개발툴의 이름이나 언어의 이름이 아닌 상당히 포괄적인 개념이다. 개념적인 면에서 살펴보자면 Microsoft의 새로운 네트워크 전략이자 제 3세대 인터넷에 대한 마이크로소프트의 비전이며 인터넷 환경을 혁신적으로 바꿀 MS의 제품군과 기술의 총칭이라고 볼 수 있다. 또한 이전까지의 소프트웨어 개념을 웹 서비스라는 이름으로 바꿈으로써 서비스로서의 소프트웨어를 활용하는 개념이라고 볼 수 있다. 기술적인 면에서 보자면 .NET은 Microsoft의 다양한 기술들을 설명하는 포괄적인 용어이다. 16비트 개발플랫폼에서 32비트로 전환한 이래, 마이크로

소프트 개발 플랫폼 역사상 가장 중대한 변화중의 하나이며 .NET은 윈도우즈 프로그래밍을 위한 전혀 새로운 개발환경이며, 새로운 API(Application Programming Interface)라고 볼 수 있다. 또한, .NET은 이름에서 알 수 있듯이 서버와 클라이언트 사이에서 분산 처리, 즉 네트워크를 통한 처리를 강조하고 있으며, 더 나아가 시간과 장소, 장치에 구애 받지 않고 언제 어디서든 정보를 이용하는 것을 그 목적으로 하고 있다. .NET은 .NET 언어로 개발된 프로그램을 Windows뿐만 아니라 다른 환경에서도 작동할 수 있도록 설계되었으며, C#을 이용한다면 다이내믹 웹 페이지(Dynamic Web Page)로부터 분산 컴포넌트, 데이터베이스 액세스 컴포넌트 그리고 기존의 Windows 애플리케이션 등 가능한 거의 모든 프로그램을 작성할 수가 있다.

[그림 3-1] .NET의 구조



[그림 3-1]에서 보면 가장 하단에는 Windows 라는 OS와 Windows의 핵심 서비스인 COM+ Services가 자리하고 있으며, .NET 프레임워크라는 가장 핵심적인 .NET의 기술은 그러한 OS와 서비스의 기반 위에 자리하고 있는 것을 볼 수가 있다. 이는 .NET 프레임워크가 독립적으로 존재하는 OS가 아닌, 플랫폼(Windows OS)기반 위에서 제공되는 프레임워크(골격), 구조라는 것이다.

.NET 프레임워크는 내부적으로 몇 가지의 요소를 가지고 있다. [그림 3-1]에서 알 수 있지만 .NET 프레임워크를 구성하는 것으로는 Common Language Runtime, Base Class Library, Data & XML, ASP.NET & Web Services, Windows Forms가 있다.

제 2 절 .NET 의 동작원리

2.1 .NET 프레임워크

웹 서비스 및 기타 응용프로그램의 작성, 배포 및 실행을 위한 환경을 의미한다. 공통 언어 런타임 (Common Language Runtime) , Framework Class 및 ASP.NET의 세가지 주요 부분으로 구성되어 있는 환경을 의미한다. 공통 언어 런타임 모듈을 통해 .NET 은 모든 언어를 사용 가능하게 하는 언어 중립적인 플랫폼으로서 기능한다. 또한 .NET 프레임워크는 단순히 ASP.NET을 이용하여 웹 어플리케이션을 작성할 수 있을 뿐 아니라 윈도우즈 어플리케이션의 개발까지도 모두 관여하는 기본적인 시스템 서비스들을 제공한다. 즉, .NET 프레임워크는 OS에 add-on 되어져서 .NET 기술에

대한 기본 지원 시스템 서비스들을 제공해 준다. 기술적으로 살펴보면 .NET 프레임워크는 공통 언어 런타임(Common Language Runtime)과 .NET 프레임워크 클래스 라이브러리(.NET Framework class library) 의 두부분으로 크게 나눌 수 있다. [그림 3-1]에서 .NET 프레임워크 부분을 살펴보면, Common Language Runtime과 Base Class library 그리고, Data & XML과 ASP.NET & Web Services, Windows Forms으로 구성되어져 있음을 확인할 수도 있다.

여기서 Base Class Library 는 I/O, 문자열, 네트워크 보안, 스레딩, 텍스트, 리플렉션, 컬렉션 등의 기본적인 클래스를 제공하는 기본 클래스 라이브러리를 말하고, DATA & XML 은 데이터 액세스와 관계한 ADO.NET과 XML 부분 및 ASP.NET과 웹 서비스는 분산 웹 어플리케이션을 구축하는 근간을 제공한다. 그리고 Windows Forms은 말 그대로 윈도우즈 어플리케이션을 구축하는 기반이 되어준다.

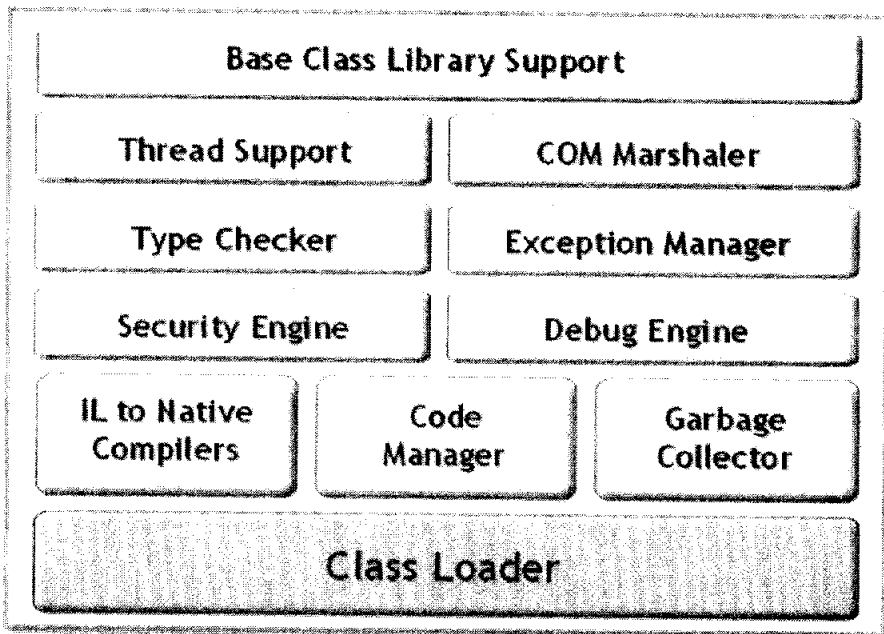
2.2 Common Language Runtime

CLR은 .NET타입들의 위치를 지정, Load, 관리하는 역할을 담당한다. 자동으로 메모리를 관리해 주고 언어를 통합할 수 있도록 하는 것 모두가 바로 이 CLR이 맡고 있는 역할이다.

CLR은 MFC나 ATL같은 개발을 돕는 라이브러리가 아니라, 프로그램이 실행되고 관리되는 환경이다. MFC나 ATL같은 개발을 돕는 라이브러리의 역할은 .NET 프레임워크 클래스 라이브러리와 Windows Forms 쪽에서 담당한다. 이 둘을 혼돈하는 것은 곤란하다. CLR은 어플리케이션의 개발을 쉽게 하고 튼튼하고 안전한 실행

환경을 제공한다. 또한, 다중 언어들을 지원하며 어플리케이션의 배포와 관리를 쉽게 해준다는 장점을 가지고 있다.

[그림 3-2] CLR(Common Language Runtime)의 구성



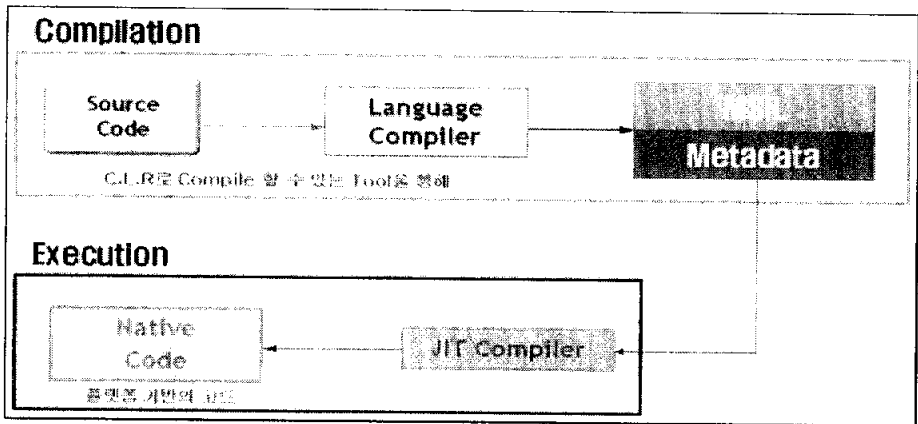
CLR은 [그림 3-2]와 같이 제일 상단에 .NET 기본 클래스 라이브러리를 지원할 수 있도록 하는 Base Class Library Support를 제공하고 있어서, 모든 .NET 언어가 .NET 기본 클래스 라이브러리를 사용할 수 있도록 해주고 있다.

2.3 .NET 실행(Execution) Model

[그림 3-3]은 코드 실행의 구조를 나타내고 있다. 모든 .NET 언어로 작성된 소스 코드는 언어에 따르는 컴파일러를 통해서 MSIL

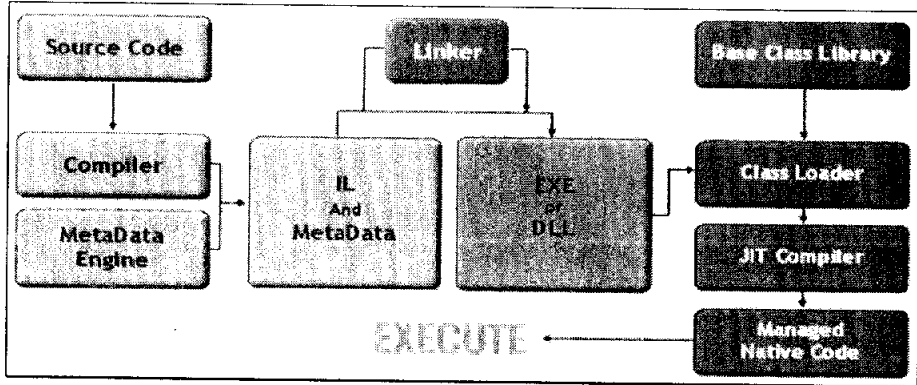
이라는 중간언어로 만들어진다. 여기까지를 Compilation 단계라고 부른다.

[그림 3-3] .NET 코드의 실행구조



이 단계의 결과물로는 중간 언어인 MSIL과 그 중간 언어에 대한 여러 가지 정보를 가지고 있는 MetaData라는 것이 생성된다. 이 때 생성된 파일의 확장자는 DLL이나 EXE이다. 이러한 중간적인 상태로 존재하던 파일들이 이제 실행시점에 들어가게 되면 CLR내의 JIT(Jitter라고도 호칭함)라는 컴파일러에 의해 다시 컴파일되고, 그 결과 실제로 실행 가능한 플랫폼 기반의 Native Code가 생성되게 된다. 여기까지의 단계를 Execution 단계라고 부른다. 이런 Execution 단계를 관할하는 것이 바로 CLR이다.

[그림3-4] 구체적 실행 모델

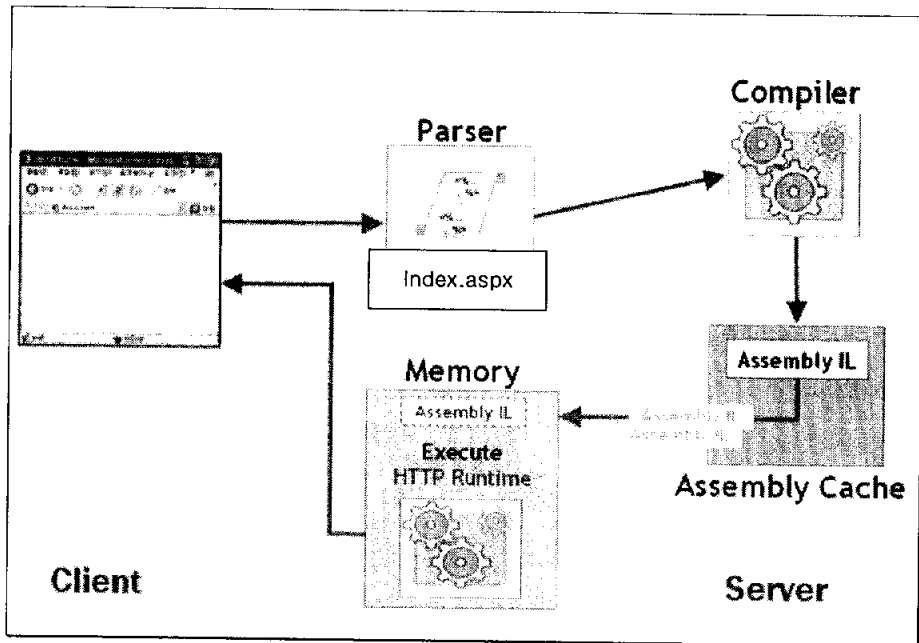


[그림3-4]는 구체적인 실행 모델이다. 먼저 소스코드가 컴파일러와 메타데이터 엔진에 의한 중간언어인 IL과 그 IL을 설명하는 메타데이터로 구성된다. 그렇게 구성된 MSIL은 그 파일과 연결된 다른 어셈블리들이 있을 경우, Linker라는 것을 통해서 하나의 EXE나 DLL로 만들어지게 된다.

그리고, 하나의 EXE나 DLL로 존재하고 있던 관리 코드들은 실행시가 되면 CLR에 의한 실행 파일이 되기 위해 다시 컴파일 과정을 거치도록 준비를 하는데, 그 첫번째가 바로 CLR의 Class Loader에 의한 클래스의 레이아웃이 잡히고, 로드되는 단계이다. 이 때, Base Class Library로부터 여러 가지 기본적인 클래스들도 같이 로드된다. 이렇게 클래스가 로드되면 그것을 Jitter로 컴파일 하게 되고 그로 인해 원시코드가 탄생하게 되는 것이다. 그리고 즉시 실행을 시작한다.

2.4 ASP.NET 프레임워크

[그림 3-5] ASP.NET의 실행구조

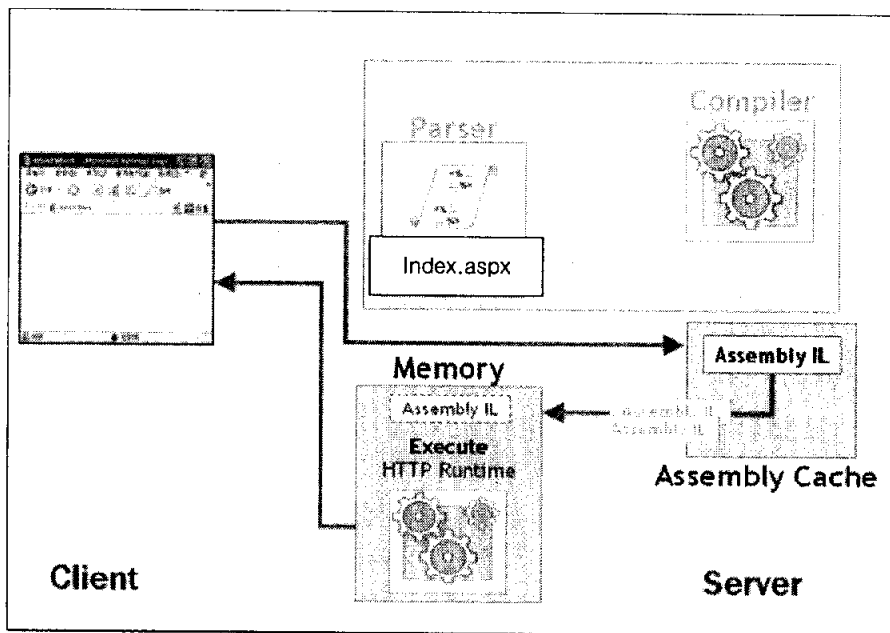


클라이언트는 서버로 특정 aspx(ASP.NET 페이지)를 호출한다. Index.aspx라는 페이지를 달라는 클라이언트의 호출을 받은 웹 서버는, 이 파일을 찾고 해당 파일의 확장자를 확인한다. 그러면 서버는 이 확장자와 연결되어 있는 DLL(ASP.NET)에게 이 파일의 처리를 의뢰한다. ASP.NET은 이 파일을 받아서 일단 파싱하는 작업을 한다. 라인 단위의 파싱 작업을 끝낸 다음에 이 파일은 서버에 의해서 일단 컴파일 된다. 컴파일이 되면 MSIL과 MetaData로 생성되고 이는 관리 코드인 Assembly IL이라는 것으로 전환된다. 여기서 확장자는 DLL이나 EXE가 된다.

관리 코드로 만들어진 것들은 어셈블리 캐쉬라는 공간에 쌓여지게 된다. 이 부분은 기존의 .NET 실행 모델과는 조금 다른데, 이것

은 ASP.NET이라는 웹 프로그래밍 특성으로 인해 제공되는 것이다. 즉, 컴파일된 aspx 페이지들은 그 중간 완성품을 캐쉬에 저장해 놓고, 이후 여러 사용자들이 같은 aspx 페이지를 요청할 경우에 새롭게 컴파일 해서 IL을 만들어 내는 것이 아니라 이 캐쉬에 이미 존재하는 IL을 재사용하게 되는 것이다. 일단, 어셈블리 캐쉬에 올라와 있는 IL들은 요청에 대한 응답을 기다리고 있는 사용자들에게 결과물을 건네주어야 하기 때문에 실행을 실시한다. 캐쉬에 있던 IL은 메모리로 올라가게 되고, 이것은 CLR의 Jitter에 의해 다시 한번 컴파일 되며, 그 결과 만들어지는 결과물들(대부분의 경우 동적으로 생성된 HTML)을 클라이언트의 브라우저에게 넘겨주게 되는 것이다.

[그림 3-6] Cache에 저장된 aspx페이지 실행구조



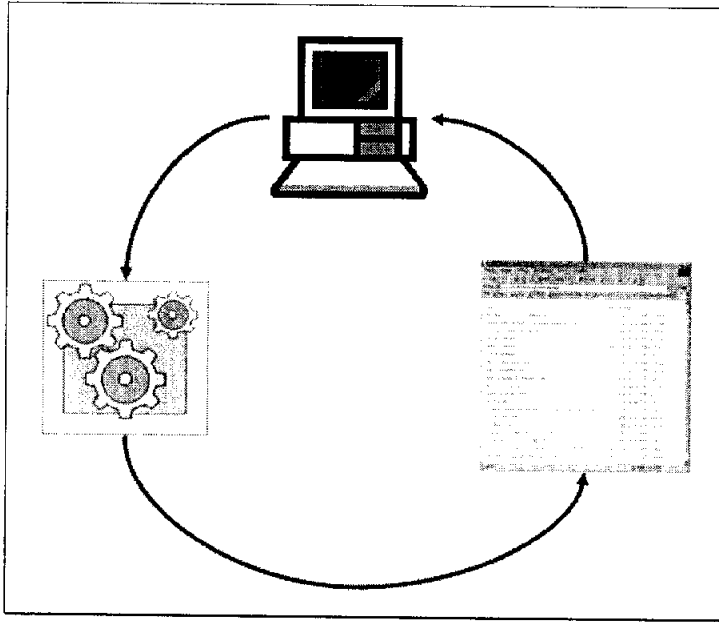
즉, 서버의 모든 aspx 페이지는 누군가가 처음 요청할 경우만 코

드 파싱, 컴파일을 수행하고, 그 후 어셈블리 캐쉬에 존재하도록 처음 페이지를 요청한 사용자는 조금 느리게 결과물을 받을 수 있겠지만, 그 이후의 사용자들은 대단히 빠르게 어셈블리 캐쉬에 이미 존재하는 관리코드를 통해서 결과를 확인할 수 있다. [그림 3-6]에 나타난 바와 같이, 일단 첫 요청에 의해 어셈블리 캐쉬에 관리 코드가 존재하게 되면, 두 번째 요청부터는 캐쉬로부터 빠르게 응답을 해 줄 수 있게 된다.

2.5 ASP.NET 개발 모델

ASP.NET의 개발 흐름을 보기 전에 먼저 이전의 ASP 개발 흐름을 보면, ASP의 경우는 컴파일 언어가 아닌 인터프리트 언어를 사용했었기에 컴파일이라는 단계가 필요하지 않았다. 단지 개발자의 몫은 단지 ASP 소스 코드를 수정하고 그냥 놔두기만 하면 되었다. 모든 ASP 페이지는 사용자의 요청시 서버에서 매번 인터프리트 되었고, 그 결과를 클라이언트에게 건네주었다. 그로 인해, 매 페이지는 매번 파싱, 인터프리트 단계를 거치는 불합리함이 있었다. 하지만, 개발자의 입장에서만은 단지 해당 소스만을 수정하면 그것으로 전부였기에 개발하기는 상당히 편리하였다. ASP.NET으로 들어서면 이제는 2번에 걸친 컴파일 단계가 요구된다. 코드 컴파일 단계와 런타임 컴파일 단계를 거치고, 코드 컴파일이 된 관리 코드는 어셈블리 캐쉬에 올라가게 되어 이후 더욱 빠르게 서비스를 제공할 수 있게 된다. 그래서 ASP.NET에서의 개발 흐름은 다음과 같은 모습을 띄게 된다.

[그림3-7] ASP.NET의 개발흐름도



마지막으로 ASP.NET에 대하여 정리해 보면, ASP.NET은 강력한 웹 어플리케이션을 구축하기 위해 서버에서 사용하는 Common Language Runtime 기반 프로그래밍 프레임워크라고 할 수 있다. ASP.NET 웹 폼(Web Forms)은 동적인 웹 유저 인터페이스(UI)를 구축하는 쉽고 강력한 방법을 제공한다. ASP.NET 웹 서비스(Web Services)는 웹 기반 분산 어플리케이션을 구축하기 위한 Building Block들을 제공한다. 웹 서비스는 HTTP와 XML같은 개방형 인터넷 표준을 기반으로 한다.

제 3 절 .NET의 특성

제 1 절과 제 2 절에서 언급한 것처럼 .NET 은 안정성과 빠른 성능을 가지고 있다. 다음의 [표 3-1]과 [표 3-2]는 Microsoft .NET 과 Sun Microsystems J2EE 를 비교한 성능비교 데이터이다.¹ Sun 의 대표적인 J2EE 애플리케이션인 Java Pet Store 의 성능과 C#으로 구현된 .NET Pet Shop 의 성능을 두 가지 관점에서 비교하고 있다. 첫째는 응답시간의 비교이다.

[표 3-1] J2EE와 .NET의 사용자 응답 시간

사용자 부하	J2EE 응답 시간 (초)	.NET 응답시간
100	0.231	0.025
200	0.308	0.028
250	0.348	0.031
300	0.361	0.036
350	0.451	0.032
400	0.538	0.034
450	0.952	0.034

[표 3-1]을 참조해 보면 J2EE로 만들어진 Pet Shop은 사용자가 450명 접속하여 사용할 때에 0.952초의 응답시간을 보였지

¹ Microsoft MSDN,” 벤치마크 결과 : Microsoft .NET vs. Sun Microsystem J2EE”,

<http://www.microsoft.com/korea/msdn/net/compare/default.asp>

만 .NET으로 만들었을 때는 0.034초에 불과하다는 것을 알 수 있다.

[표 3-2] J2EE와 .NET의 CPU사용량 비교

사용자 부하	J2EE CPU 사용 %	.NET CPU 사용 %
100	15%	5%
200	31%	8%
250	비공개	9%
300	49%	12%
350	61%	14%
400	70%	18%
450	83%	23%

두 번째는 CPU 사용량의 비교이다. [표 3-2]에서 보면 J2EE로 만든 Pet Shop에서는 450명 동시 접속시 83%의 CPU사용량을 나타내고 있지만, .NET으로 만든 것에서는 23%의 CPU사용량을 나타내고 있다. CPU사용량이 적다는 것은 그만큼 시스템의 부하가 적으므로 안정성과 응답률이 높다는 것을 의미한다.

제 4 절 SQL 서버

4.1 SQL 서버의 특징

SQL 서버는 마이크로소프트에서 인터넷 시대를 대비해서 개발한 RDBMS이다. 오라클보다 상대적으로 저렴한 가격과 간편한 조작방

법으로 웹 어플리케이션의 데이터베이스 시스템으로서 각광 받고 있다.

SQL서버의 특징을 살펴보면 다음과 같다.

첫째, 다른 DBMS와의 통합성이 뛰어나다는 것이다. SQL 서버는 오라클과 데이터조인 및 조작이 가능하고 Access나 Excel의 데이터를 SQL 서버의 데이터로 실시간으로 간편하게 변환하거나 그 반대의 경우로 변환하는 것이 가능하다.

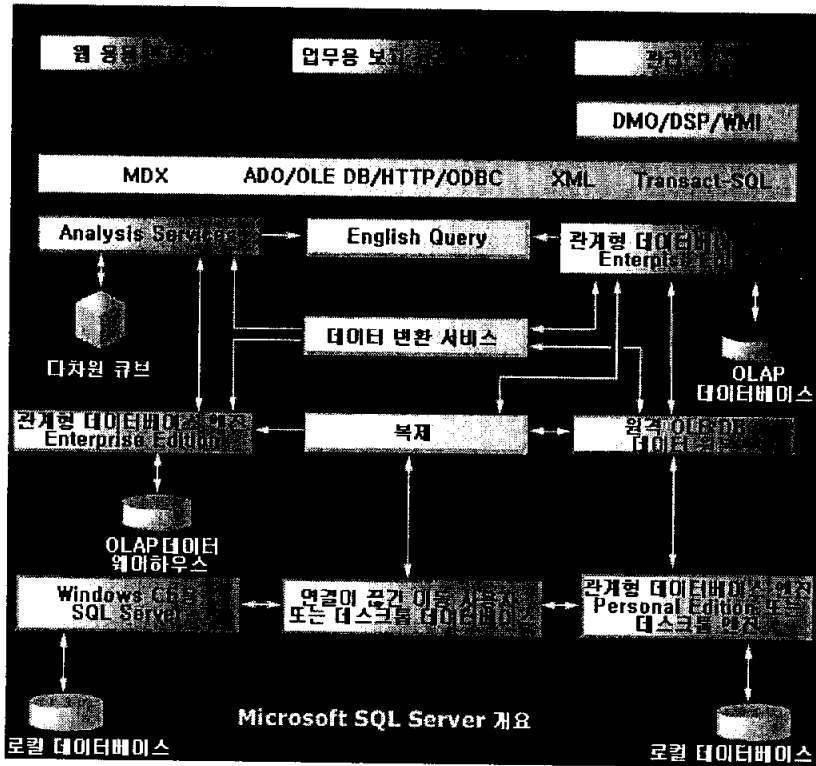
둘째, 누구나 사용하기 간편한 인터페이스를 가지고 있다. DBMS를 잘 모르더라도 윈도우즈의 사용도와 비슷하게 그래픽모드에서 데이터베이스나 테이블의 생성이 가능하고 백업이나 복구작업이 마법사 기능을 통해서 쉽게 이루어질 수 있다. 이러한 점은 학습평가 시스템이 교육현장에서 널리 사용되려면 전문인력이 아니더라도 간단한 교육만으로 관리하게 할 수 있는 것을 가능하게 한다.

셋째, 완벽하게 한글화되어 있다는 점이다. 데이터 입출력의 한글 지원은 물론이고, 각종 메뉴와 설명서를 대신하는 도움말까지 모두 한글화가 완료되어 쉽게 도움말을 참조할 수 있다.

4.2 SQL 서버의 동작원리

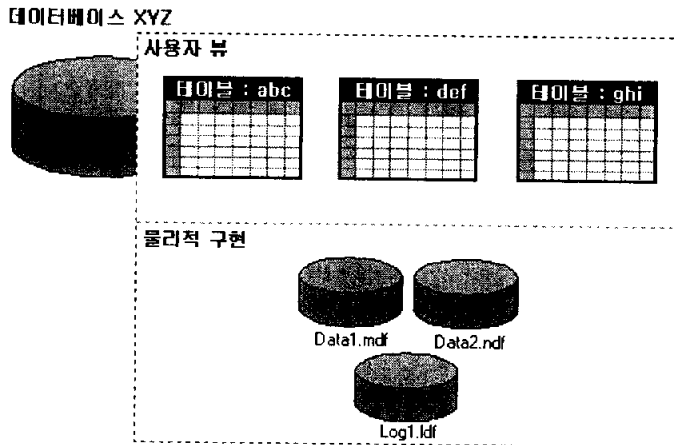
SQL 서버는 크게 나누어 관계형 데이터베이스 엔진과 저장소 엔진, 데이터베이스 파일로 나눌 수 있다. 우선 SQL 문의 실행을 담당하는 관계형 엔진의 구조는 [그림 3-8]과 같다.

[그림 3-8] SQL 서버 관계형 엔진 개요



관계형 엔진이 하는 역할은 테이블에 데이터를 저장하고 자체적으로 성능을 튜닝하며 동적으로 시스템의 리소스를 확보하고 해제하며 사용자 인증 등의 작업을 하는데 있다. 그리고 저장소 엔진은 데이터를 디스크의 파일에 저장하는 역할을 하는데 그림과 같은 구조를 가지고 있다.

[그림 3-9] 데이터베이스 저장소 엔진 개요



데이터베이스의 데이터는 사용자가 볼 수 있는 논리적 구성 요소로 구성되며 또한 데이터베이스는 물리적으로 디스크 상에서 두 개 이상의 파일로 구현된다. 그리고 데이터베이스를 사용할 때에는 주로 테이블, 뷰, 프로시저 및 사용자와 같은 논리적 구성 요소로 작업하게 된다.

제 4 장 .NET을 이용한 학습평가 시스템의 설계

제 1 절 시스템 개발 환경

본 연구에서는 시스템 및 소프트웨어 환경으로 서버는 Pentium III 급 이상의 PC 서버 환경으로 운용하여야 안정적이며 각종 문제 및 결과, 학생 데이터 등을 저장하기 위하여 10GB 이상의 데이터 저장장치를 필요로 한다. 또한 Windows 2000 Server 운용 체제와 SQL Server 2000을 설치하여 사용하기 위하여 최소한 256 MB 이상의 메모리를 필요로 한다. 초기 버전에서는 소수의 사용자를 대상으로 해서 높은 시스템 사양을 필요로 하지 않지만, 실제 개발되어 사용되는 Product 버전에서는 메모리 512 MB 이상이며 백업 디바이스 장치가 되어 있는 PC 서버급 장치를 필요로 한다.

응용 프로그램 환경으로는 웹 서비스를 위한 Internet Information Server 와 .NET 개발을 위한 Visual Studio .NET 환경이 있어야 한다. 이를 이용하는 클라이언트는 Pentium II 이상 64MB의 이상의 램을 갖추고 운영 체제로는 Windows 98 이상의 시스템과 웹 브라우저로는 Internet Explorer 5.5 이상의 버전을 필요로 한다. Netscape로도 구동할 수 있겠으나 Microsoft사의 .NET 플랫폼으로 구축하는 관계로 일부 기능에서는 오작동 할 수도 있다.

제 2 절 .NET을 이용한 학습평가 시스템의 구성 및 기능

.NET을 이용한 학습평가 시스템은 온라인 상에서 문제를 출제하고 시험에 응시할 수 있는 시험 테스트 시스템을 말한다. 이 시스템은 우선 사용자는 관리자, 학사 관리자, 문제 출제자, 학생으로 분류된다. 좀 세부적으로 보면 관리자는 온라인 학습평가 시스템의 전체를 관리하는 관리자를 말한다. 따라서 관리자는 학사 관리자, 문제 출제자, 학생 관리, 시험 일정 관리, 시험지 관리 등 전반적인 모든 사항을 관리한다. 학사 관리자는 학사를 관리하는 관리 요원을 말한다. 각 시험에 대한 성적을 여러 형식으로 볼 수 있으며, 개인에 대한 성적도 볼 수 있다. 또한 평가 과목에 대한 추가, 삭제 등의 업무를 할 수 있다. 문제 출제자는 교수나 강사를 말한다. 이들은 자신이 강의하는 과목에 대하여 문제를 출제할 수 있다. 또한 출제된 문제에 대하여 각종 정보를 볼 수 있다. 그리고 자신이 강의하는 과목에 대하여 학생들의 성적을 볼 수 있으며 출제한 과목의 시험에 대하여 관리할 수 있다. 학생은 수업을 수강하는 일반 학생을 말한다. 이들은 자신이 수강하고 있는 과목에 대하여 출제된 문제에 응시할 수 있다. 또한 자신이 응시한 문제에 대하여 각종 정보를 볼 수 있으며 자신의 성적을 모두 열람할 수 있다.

기능은 각 사용자를 기준으로 분류한다. 우선 관리자 모드는 관리자 아이디 및 비밀번호 관리, 출제자 관리, 수업 과목 추가/삭제/수정 및 과목의 단원 추가/삭제/수정(담당 강사의 역할 고려), 학사 관리자 관리, 학생 관리, 학과 관리, 로그 파일에 대한 내용 열람, 데이터 관리(백업, 복구), 시험 과목 등 시험 등록에 필요한 정보 관리

(추가, 수정, 삭제), 시험지 관리(추가, 수정, 삭제) 등을 가지고 있다. 그리고 학생 모드는 수강 신청 관리(신청, 수정), 자신이 수강하는 수업에 대한 목록 열람, 자신이 수강하는 수업 중 현재 진행 중인 시험 목록 열람 및 응시, 각 과목별 응시한 시험에 대한 결과 열람, 자신의 정보 관리 등의 기능을 가지고 있다. 우선 온라인 평가 시스템에서 데이터라 함은 시스템에서 사용되는 사용자 정보, 시험 정보, 문제 정보 등 모든 정보들을 말한다. 따라서 효율적인 관리와 백업을 위하여 데이터는 기본적으로 DBMS를 사용하고, 가공된 데이터는 최종적으로 DBMS에 저장됨을 원칙으로 한다.

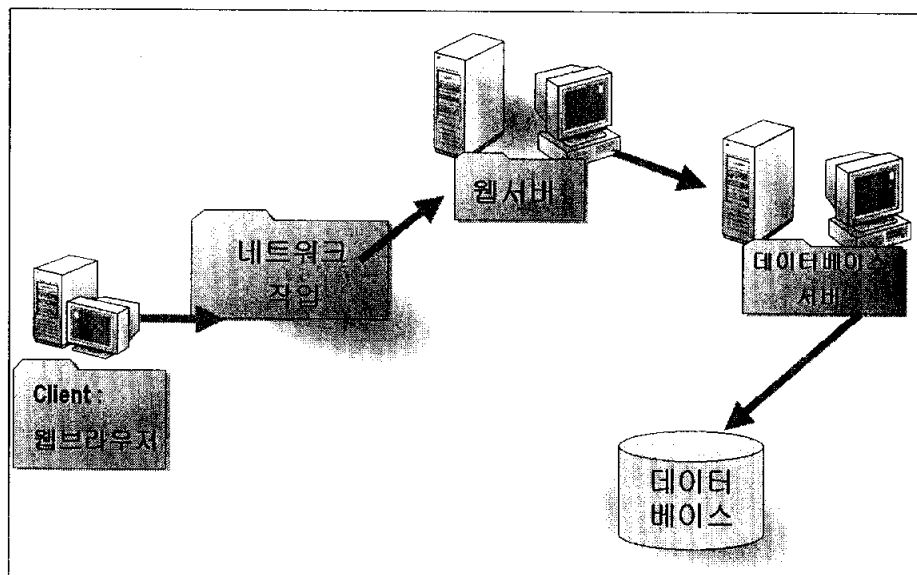
그리고 평가(시험)라는 특성상 학습평가 시스템을 구축할 때 문제 유형에 대한 정의가 중요하다. 문제 유형에 따라 평가 방법과 채점 방식, 채점 기간 등이 확연히 틀려지기 때문이다. 우선 출제 문제는 문제 은행식과 논술식으로 구분된다. 문제 은행식은 다시 무작위 출제식과 임의 지정식으로 구분된다. 무작위 출제식은 저장된 문제들을 각 단위별로 지정한 문제 수에 한해서 무작위로 추출하여 문제지를 생성하는 것을 말하는데 이 경우는 응시자에 따라 문제가 달라진다. 임의 지정식은 저장된 문제들 중에서 출제자가 문제를 선정하여 출제하는 것을 말한다. 이 경우에는 무작위 출제식과 달리 모든 응시자의 문제가 동일하다. 논술식은 저장된 논술 문제 중에서 문제를 선택하여 출제하며, 응시자가 응시한 후 공시 기간에 출제자가 문제를 채점하여 직접 결과를 입력하는 것을 말한다. 이런 논술식의 경우에는 출제자가 결과를 작성하기 전에는 응시자는 결과를 확인할 수 없다. 문제 채점에 대한 부분을 보면 문제 채점에는 크게 문제 은행식과 논술식으로 구분되는데 우선, 문제 은행식은 응시자가 응시 후 시스템이 바로 채점을 하며, 논술식은 공시기간 동안 출

제자가 직접 채점을 하게 된다.

2.1 .NET을 이용한 학습평가 시스템의 하드웨어적 구성

학습평가 시스템의 하드웨어 구성을 보면 다음과 같다.

[그림 4-1] 학습평가 시스템의 하드웨어 구성

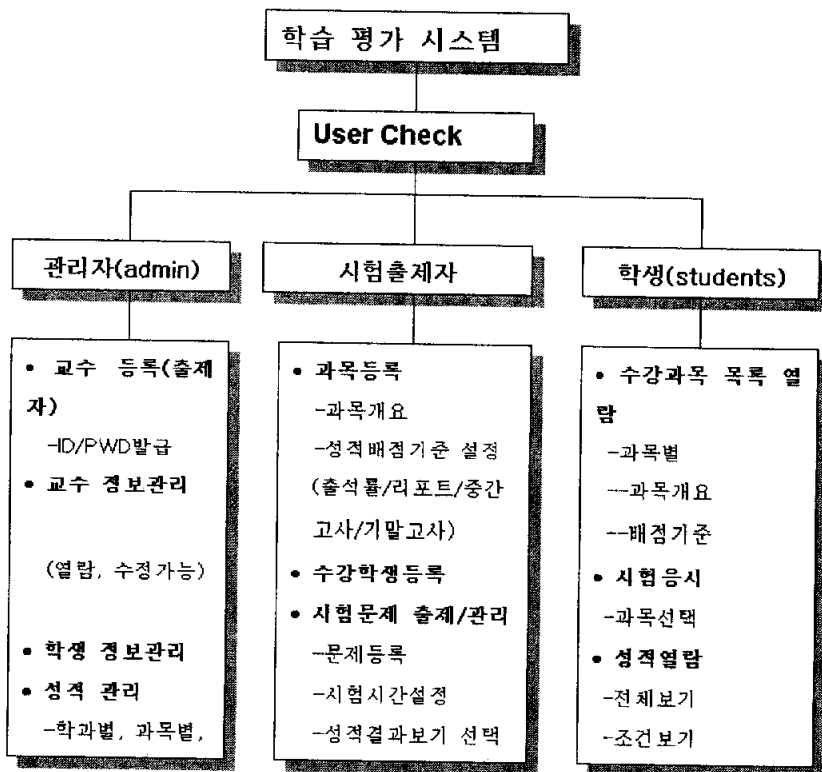


우선 클라이언트는 Windows 95 이상의 OS에서 인터넷 익스플로러를 가지고 학습평가 시스템에 접속한다. 그리고 학습평가 시스템을 운용하는 서버는 웹 서버와 데이터베이스 서버로 나뉘어진다. 이렇게 서버를 분할한 이유는 하나의 서버에 전용 웹 서비스와 데이터베이스 서비스 기능을 부여해 시스템의 부하를 줄이고 안정성을 높이기 위해서이다.

2.2 .NET을 이용한 학습 평가 시스템의 논리적 구성

.NET을 이용한 학습 평가 시스템은 web기반 시스템으로 온라인 상에서 문제를 출제하고 시험에 응시할 수 있고 성적조희가 가능한 시험 시스템을 말하며, 관리자, 시험출제자, 학생으로 구성된다. 관리자는 시험 출제자와 출제자가 신청한 강의 과목을 승인, 인증하며 학생들의 성적을 열람하는 등 온라인 시험센터의 전반적인 부분을 관리한다.

[그림 4-2] 학습평가 시스템의 구조



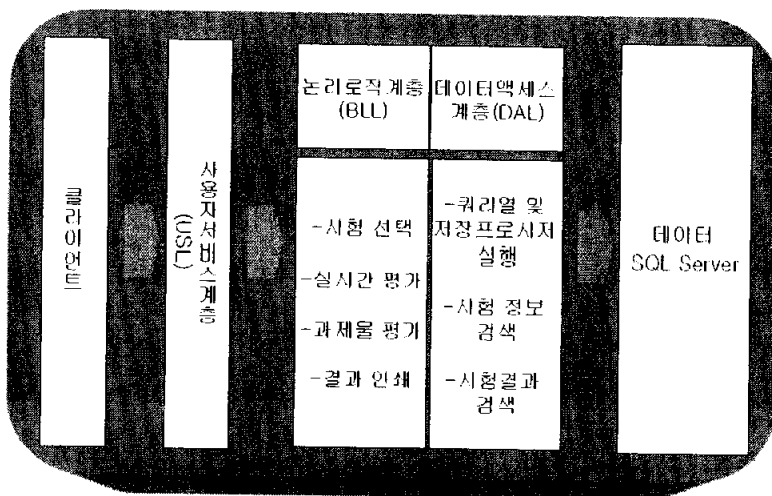
[그림 4-2]처럼 시험 출제자는 관리자가 승인한 계정으로 온라인

시험센터에 Login 할 수 있다. 그리고 담당과목을 수강 신청한 학생들의 정보를 입력하고 강의 과목의 개요와 시험 배점 등을 설정하며, 시험 문제를 출제하고 학생들의 성적을 입력, 수정, 삭제 할 수 있다. 학생은 로그인 후 시험에 응시할 수 있으며 성적을 조회 할 수 있다.

2.3 .NET을 이용한 학습 평가 시스템의 계층적 구성

온라인 시스템은 실시간으로 일정한 수의 사용자가 액세스하고 이들이 응시하는 시험에 대한 정확한 채점이 중요시 된다. 이러한 내용들의 DB 액세스를 위하여 Windows 2000 서버 상에서 상당히 빠른 요구 응답 시간을 나타내고 있는 Microsoft 사의 SQL Server 2000 과 ADO.NET 컴포넌트를 이용하였다.

[그림 4-3] .NET을 이용한 평가 시스템 계층 구조



[그림 4-3]처럼 각각의 기능별로 논리적 계층으로 나누어서 이후의 재활용성과 유지보수성, 코딩효율성을 높일 수 있는 컴포넌트형 구조를 갖추었다. 각각의 모듈을 살펴 보면, 우선 사용자 서비스 계층은 웹 계층으로서 웹 브라우저 클라이언트에 응용 프로그램에 대한 액세스를 제공한다. 일종의 웹 프로젝트로 구현되어서 웹 계층은 ASP.NET Web Forms과 코드 숨김 파일로 구성된다. 코드 숨김 파일에서는 다양한 컨트롤에 대한 이벤트 처리를 구현하는 반면, Web Forms에서는 HTML에서의 코드만을 가지고 있다. 비즈니스 논리 계층은 시험선택, 실시간 평가, 과제물 평가, 결과 인쇄 등의 서버상에서의 논리적 처리를 구현하는 계층이다. 또한 데이터엑세스 계층은 DBMS에 액세스 하는 SQL 쿼리열을 포함하는 계층으로서, 시험결과 검색, 과목검색, 학생, 출제 데이터를 검색하게 하는 모듈이다. 이 계층은 Microsoft사의 ADO.NET으로서 구현한다. 가장 후위에 위치하고 있는 데이터 계층은 SQL Server 2000으로서 구현하는 계층으로 가장 중요한 시험데이터와 결과데이터 사용자 정보 데이터 등을 가지고 있는 계층이다.

제 3 절 .NET을 이용한 학습 평가 시스템의 모듈별 구성

3.1 관리자 모듈 구성

[표 4-1] 관리자 모듈 구성표

구 분	기 능
교수등록 (출제자)	ID 와 Password를 발급한다.
교수 정보 관리	등록된 교수의 정보를 관리하는 기능으로 교수 아이디, 비밀번호, 성명, 주민번호 등을 조회와 수정을 할 수 있다. 교수가 담당과목을 신청하면 담당과목을 승인한다.
학생 정보 관리	학생정보를 관리하는 기능으로 학생의 아이디, 비밀번호, 성명 등을 조회할 수 있다.
성적 관리	과목별, 학과별, 학번으로 전체 시험 성적조회 및 열람이 가능하다.

3.2 학생 모듈 구성

[표 4-2] 학생 모듈 구성표

구 분	기 능
과목 목록 열람	과목별 리스트와 과목별 개요 및 배점기준 등을 열람할 수 있다.
시험 응시	과목을 선택하면, 승인 받은 교수(출제자)가 출제한 시험에 응시할 수 있다. 정해진 시험 시간 이외의 시간에는 시험응시와 열람이 불가능하다.
성적 열람	응시한 과목의 성적을 과목별로 볼 수 있으며, 출제자의 성적고시 시간에 또는 기간에 볼 수 있다.

3.3 교수(출제자) 모듈 구성

[표 4-3] 출제자 모듈 구성표

구 분	기 능
과목 등록	과목개요 신청한 담당 과목의 교육 안을 작성 할 수 있다. 교육안의 내용은 단원소개, 교육 목적, 사용교제 등에 대한 내용을 입력 할 수 있다.
	성적 배점 기준 성적 배점 기준을 설정 할 수 있다. - 출석률 % - 리포트 % - 중간 고사 % - 기말 고사 %
수강 학생 등록	수강 과목 수강 학생에 대한 정보를 입력, 수정,삭제 할 수 있다.(ID, PWD 기타)
시험 출제 관리	문제 등록 승인 받은 출제자(교수)가 해당 과목 문제를 출제하고 등록 한다. 등록 시 -시험시간 결정(시작/종료시간), 성적 결과 보기 시 -바로보기 -일정시간 경과 후 보기 -교수가 지정한 시간에 보기등 선택가능

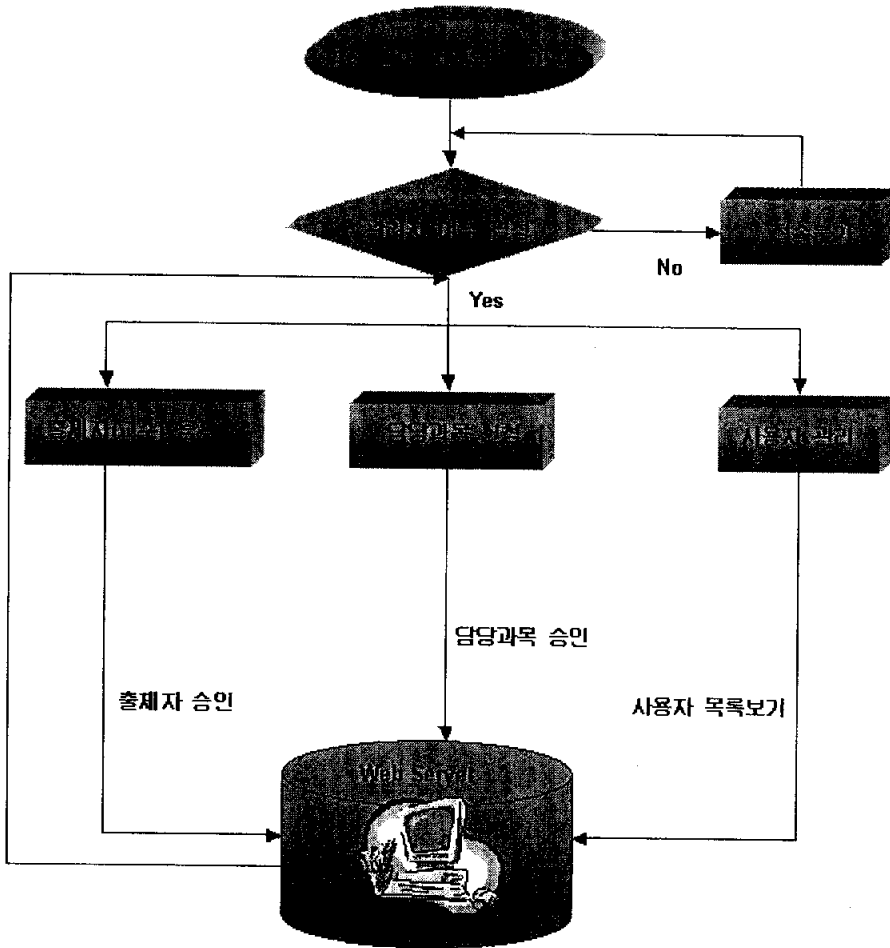
	문제 관리	승인 받은 출제자(교수)가 등록된 문제를 열람 할 수 있고, 수정, 삭제가 가능하다.
시험 응시자 관리	성적 확인	학과별, 과목별, 학번/이름으로 성적 조회 및 열람이 가능하다.
	성적 관리	시험응시 학생들의 성적 입력 및 수정, 삭제가 가능하다.

제 4 절 .NET을 이용한 학습 평가 시스템의 흐름도

4.1 관리자 모듈

온라인 시험 센터(On-Line Exam. Center)를 관리하는 관리자 모듈로서, 온라인 시험 센터에 접속해서 과목을 등록하고 시험을 출제할 수 있는 교수(출제자)를 등록 및 승인하며 학생의 성적을 열람할 수 있다. [그림 4-4]에서 보면 관리자가 자신의 ID로 로그인 하면 학습평가시스템은 데이터베이스의 admin테이블을 조회하여 관리자의 ID인지 그리고 암호가 맞는지 여부를 검사하여 인증을 하게 된다.

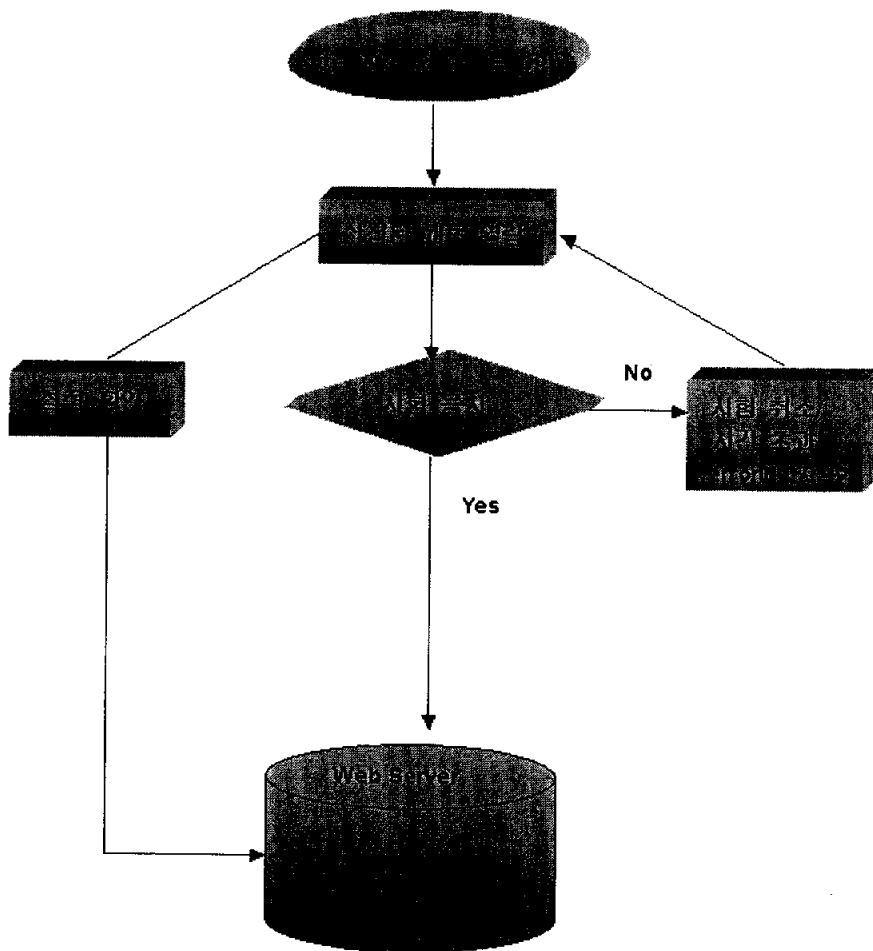
[그림 4-4] 관리자 모듈 흐름도



4.2 학생 모듈

승인 받은 출제자가 출제한 문제를 정해진 시간에 온라인 시험 센터에 접속해서 시험을 치고 성적을 확인할 수 있는 모듈로 시험 센터의 핵심 기능중의 하나라고 할 수 있다. 학생 모듈에서는 시험 응시와 성적 열람이 가능하다.

[그림 4-5] 학생모듈 흐름도

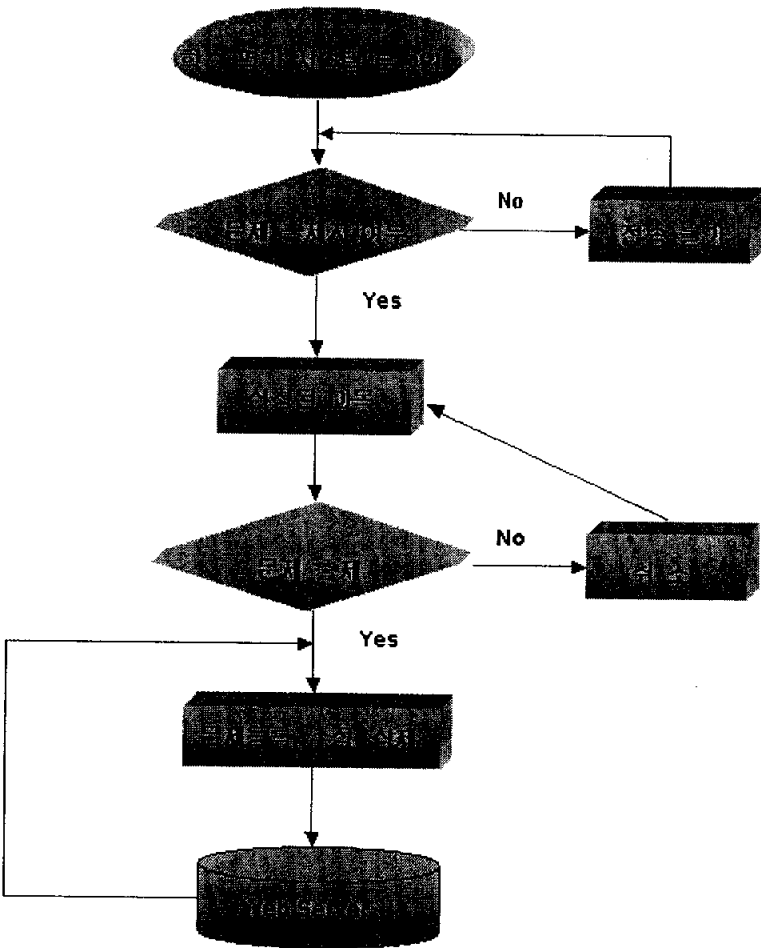


4.3 교수(출제자) 모듈

문제 출제자 모듈은 문제 출제자가 온라인 시험 센터에 접속해서 관리자기 승인하고 발급해준 ID와 Password로 로그인 후 담당과목에 따른 개요설명, 학습계획서, 성적 배점등을 설정할 수 있다. 그리

고 수강 등록한 학생의 개인 정보를 입력, 수정, 삭제가 가능하고
담당 과목에 대한 문제를 출제해서 학생들이 온라인 상에서 시험을
치고 학생의 성적을 입력, 수정, 삭제할 수 있다.

[그림 4-6] 출제자 모듈 흐름도



제 5 절 .NET을 이용한 학습평가 시스템의 데이터베이스 설계

5.1 DB의 Table Directory

[표 4-4] 학습평가 시스템의 테이블목록

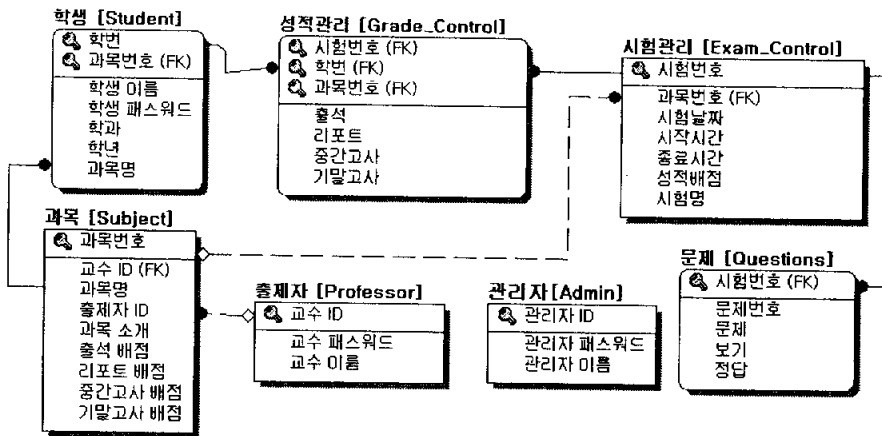
테이블 이름	설 명
Admin(관리자)	온라인 시험 센터의 관리자의 정보를 담고 있는 테이블
Student(학생)	시험에 응시하는 학생의 정보를 담고 있는 테이블
Professor(교수-출제자)	교수(문제출제자)의 정보를 담고 있는 테이블
Subject(과목)	교수(문제출제자)가 강의 하고 있는 과목의 정보를 담고 있는 테이블
Exam_Control(시험관리)	시험에 관한 모든 정보를 담고 있는 테이블
Grade_Control(성적관리)	시험응시한 학생들의 성적에 관한 정보를 담고 있는 테이블
Questions(문제)	승인 받은 교수(출제자)가 등록한 문제의 정보를 담고 있는 테이블

관리자 테이블은 .NET을 이용한 학습평가 시스템의 전반적인 부분을 관리하는 관리자의 ID, Password 등의 정보를 저장한다. 교수 테이블은 학생의 정보를 입력, 수정, 삭제하고 문제를 출제, 입력, 수정, 삭제하는 출제자의 ID, Password, 이름등의 정보를 저장한다. 학생 테이블은 시험 응시를 한 학생의 학번, 학과, 학년, 이름, 주민

번호, 수강 과목 등의 정보를 저장한다. 과목 테이블은 과목명과 과목의 강의자(출제자)의 ID(번호)와 과목소개, 출석 배점, 리포트 배점, 중간고사, 기말고사 등의 정보를 저장한다. 시험관리 테이블은 시험번호, 시험명, 시험날짜, 과목번호, 시작시간, 종료시간, 성적 확인방법 등의 정보를 저장한다. 성적관리 테이블은 학번, 시험번호, 출석, 리포트, 중간고사, 기말고사 성적이 저장되는 테이블이다. 문제테이블은 문제번호, 시험번호, 문제, 객관식의 보기, 정답 등 문제 출제가 등록한 문제의 정보를 저장한다.

5.2 Entity Relationship Diagram

[그림 4-7] .NET을 이용한 학습평가시스템의 ERD



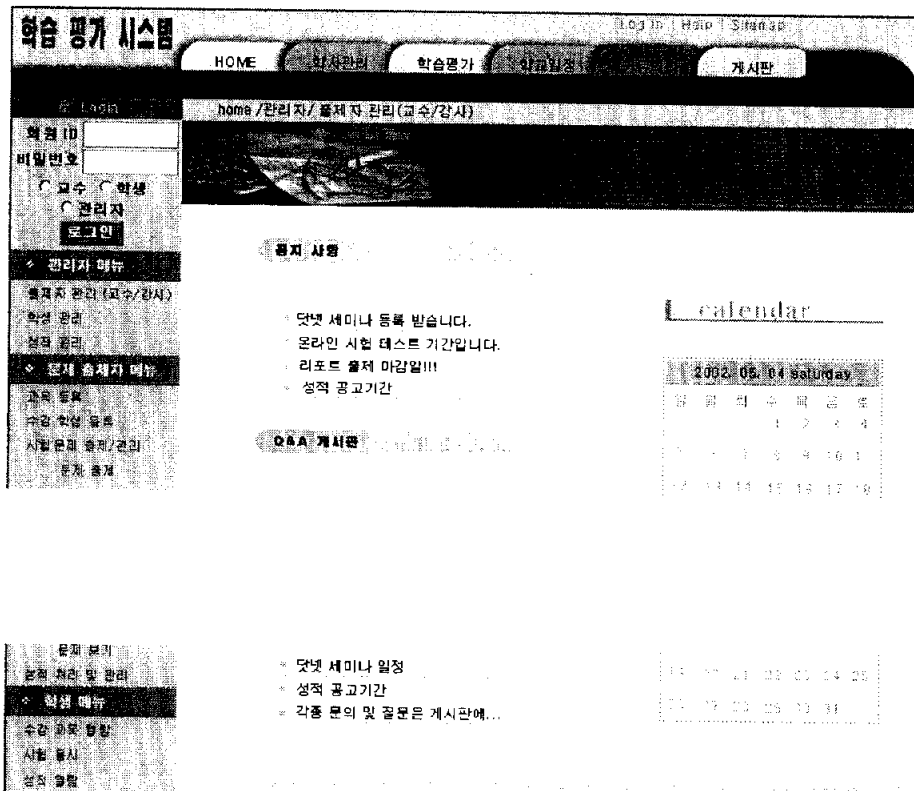
[그림 4-7]에서 보면 학생테이블은 학번과 과목번호를 주키로

가지고 있다. 그리고 성적관리테이블은 시험번호, 학번, 과목번호를 키로서 가지고 있다. 이렇게 여러 개의 복합키를 가지고 있는 것은 학생에 따라 같은 과목일지라도 중간고사, 기말고사의 형식으로 시험이 여러 개가 있을 수 있기 때문이다. 과목테이블은 출제자가 각각의 배점을 지정할 수 있게 배점비율 항목을 따로 컬럼으로 분리해 놓았다. 그리고 시험관리 테이블은 과목번호와 시험의 날짜, 시작 및 종료시간 배점, 시험명 등을 지정했다.

제 5 장 .NET을 이용한 학습평가 시스템의 구현

학습평가 시스템의 처음 화면은 회원가입, 학사관리(성적관리), 학습평가(온라인 시험), 학교일정, 커뮤니티, 게시판 등의 메뉴로 구성되어 있다. 로그인에 대상에 따라 사용자별로 관리자 메뉴(출제자 관리(교수/강사), 학생등록, 성적관리), 문제출제자(교수/강사) 메뉴(과목등록, 수강학생 등록, 시험문제 출제/관리 - 문제출제, 문제보기, 성적 처리 및 관리), 학생 메뉴(수강과목 열람, 시험응시, 성적 열람)로 구성된다.

[그림 5-1] 학습평가 시스템의 메인 화면



제 1 절 관리자 메뉴

학습평가 시스템에서 관리자는 문제 출제자의 ID와 Password를 발급 및 승인해 주고 등록된 문제 출제자의 정보를 수정 및 삭제할 수 있으며 학생정보를 관리해 ID, Password, 성명 등을 조회할 수 있다. 또한 과목별, 학과별, 학번으로 전체 시험 성적조회 및 열람이 가능하도록 메뉴가 구성되어 있다.

1.1 출제자 관리(교수/강사)

관리자로 접속을 하면 관리자 메뉴(출제자 관리(교수/강사), 학생 관리, 성적관리)가 뜬다. 여기서 '출제자 관리'라는 메뉴를 누르게 되면 관리자가 승인해준 출제자의 목록이 나타나고, 여기서 출제자의 정보를 수정 및 삭제할 수 있다.

[그림 5-2] 출제자 관리 화면

출제자 관리(교수/강사)

아이디	패스워드	이름	수정	삭제
angel	blue34	이원재	수정	삭제
jeonjo	joshua	전호인	수정	삭제
kimsh	sh2564	김선희	수정	삭제
professor	1234	최창욱	수정	삭제
rondo99	ron13	박현주	수정	삭제
sasimi	lee1564	이석용	수정	삭제
shonhs	hs9456	손훈상	수정	삭제
son2000	sn1956	손인순	수정	삭제

1.2 학생관리

여기서는 학생 정보를 열람 할 수 있는데 처음에 접속을 하게 되면 학번별로 정렬된 학생 목록을 볼 수 있으며 학과별, 과목별, 학번으로 조회가 가능하다. 학생의 정보는 학생 자신이 등록하는 것이 아니라 문제 출제자가 입력하기 때문에 관리자라고 해도 문제 출제자의 동의 없이는 학생 정보를 수정 또는 삭제할 수 없다.

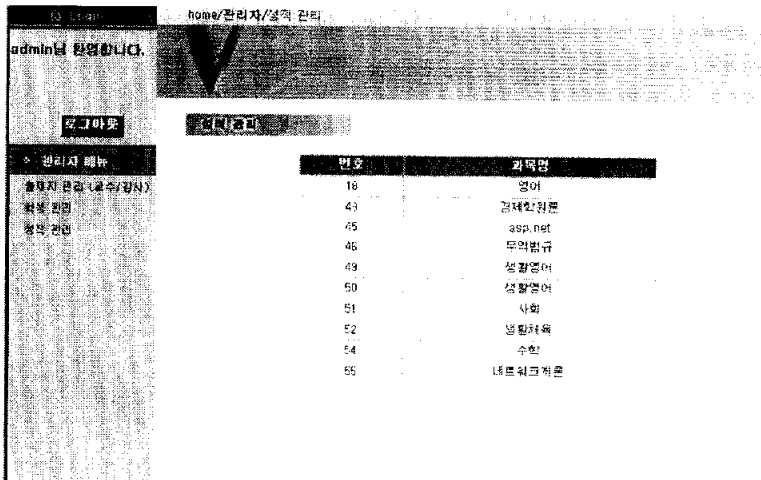
[그림 5-3] 학생 리스트 화면

학번	이름	내일번호	학과	학년	과목
13	Lee	13	computer	3	영어
20005475	김미정	2147821	컴퓨터공학	1	영어
20011245	이호리	2103336	경제	3	경제학원론
20014538	김민수	1105486	로봇공학	4	영어
20015639	이진	2104739	무역	4	경제학원론
2002345	김현수	261111	유아교육	1	영어
20025778	정두문	2102546	무역	4	무역법
214578	박세진	2148783	경영	4	무역법
998798	박현준	2105255	무역실무	2	영어
9912455	김선화	267896	유아교육	2	영어
991234	정태우	1234	건축학과	2	수학
991234	구진순	3456	유아교육	3	수학
9999	정유현	1111	경영학과	4	네트워크개론

1.3 성적관리

성적관리 메뉴를 선택하게 되면 과목별 학점이 나타난다. 여기서 과목을 선택하게 되면 그 과목에 대한 세부적인 성적을 조회할 수 있다. 성적 부분도 문제 출제자의 고유 권한이기 때문에 문제출제자의 동의 없이는 성적 정보를 수정 또는 삭제할 수 없다.

[그림 5-4] 성적관리 화면



[그림 5-5] 로그인 하는 C#, ASP.NET 코드

```
//Log_in하는 부분
private void Button1_Click(object sender, System.EventArgs e)
{
    sqlCommand1.Connection.Open();
    string check = null;
    //체크값에 따라 쿼리문 실행
    if(radAdmin.Checked)
    {
        sqlCommand1.CommandText = "select * from admin where
adminID='"+ txtID.Text+ "'";
        check = "admin";
    }
}
```

```

    }

    else if(radPro.Checked)
    {
        sqlCommand1.CommandText = "select * from professor where
pro_id='"+ txtID.Text+ "'";

        check = "professor";
    }

    else
    {
        sqlCommand1.CommandText = "select * from student where
st_id='"+ txtID.Text+ "'";

        check = "student";
    }

    reader = sqlCommand1.ExecuteReader();
    while(reader.Read())
    {
        //아이디가 존재하면

        if(txtPwd.Text.Trim().Equals(reader["pwd"].ToString().Trim()))
        {
            //Response.Write("true");

            if(Session["key"]==null)
            {
                Session["key"] = new System.Collections.ArrayList();
            }
        }
    }
}

```

```

Session["chk"] = new System.Collections.ArrayList();
Session["key"] = txtID.Text;
Session["chk"] = check;
//      Response.Redirect("main.aspx");
        }

```

그리고 관리자와 출제자, 학습자가 로그인하게 되면 각자의 상황에 맞는 메뉴가 나타나게 하여야 한다.

[그림 5-6] 로그인 후 다른 메뉴 생성을 위한 C#, ASP.NET 코드

```

Response.Write("<p><br><br><br><Br><Br><Br><Br><Br>");
Response.Write(Session["key"].ToString());
//Response.Write(Session["chk"].ToString());
Response.Write("님 환영합니다.");
Response.Write("<table>");
if(Session["chk"].ToString().Trim().Equals("admin"))
{Response.Write("<tr><td>width=180><img
src='images/menu_01.gif' width='150' height='20'><Br></td></tr>");
Response.Write("<tr>");
Response.Write("<td width=180>");
Response.Write("<ul><li><font face='돋움' size='2'><a
href='pro_control.aspx' target='iframe2'>출제자
관리(교수/강사)</a></font></li>");
Response.Write("<li><font face='돋움' size='2'><a

```

```

href='student_control.aspx'
target='iframe2'>학생 관리</a></font></li>");

        Response.Write("<li><font face='돋움' size='2'>성적
관리</font></li></ul>"); }

        else if(Session["chk"].ToString().Trim().Equals("professor"))
        {
            Response.Write("<tr><td><img src='images/menu_02.gif'
width='180' height='20'><br></td></tr>");

            Response.Write("<tr>");

            Response.Write("<td>");

            Response.Write("<ul><li><font face='돋움' size='2'>과목
등록</font></li>");

            Response.Write("<li><font face='돋움' size='2'>수강 학생
등록</font></li>");

            Response.Write("<li><font face='돋움' size='2'>시험문제 출제 /
관리</font></li>");

            Response.Write("<li><font face='돋움' size='2'>성적 처리 및
관리</font></li> </ul>"); }

        Else {

            Response.Write("<tr><td><img src='images/menu_03.gif'
width='143' height='20'><br></td></tr>");

            Response.Write("<tr>");

            Response.Write("<td>");

```

```

        Response.Write("<ul><li><font face='돋움' size='2'>수강 과목  
열람</font></li>");

        Response.Write("<li><font face='돋움' size='2'>시험  
응시</font></li>");

        Response.Write("<li><font face='돋움' size='2'>성적  
열람</font></li></ul>"); }

    Response.Write("</td>");

    Response.Write("</tr>");

    Response.Write("</table>");

```

제 2 절 문제 출제자(교수/강사) 메뉴

문제 출제자는 문제 출제자가 온라인 시험 센터에 접속해서 관리자가 승인하고 발급 해준 ID와 Password로 로그인한다. 로그인 후, 담당과목에 따른 개요설명, 학습계획서, 성적 배점등을 설정할 수 있으며 수강 등록한 학생의 개인 정보를 입력, 수정, 삭제가 가능하다. 그리고 담당 과목에 대한 문제를 출제해서 학생들이 온라인 상에서 시험을 치르고, 또한 학생의 성적을 입력, 수정, 삭제할 수 있다.

2.1 과목등록

담당과목의 개요 설명, 학습계획서, 성적 배점(중간고사, 기말고사, 출석률, 리포트 점수 등)을 설정하는 페이지로 문제출제자는 여기서 입력한 내용을 열람 또는 수정, 삭제할 수 있다.

[그림 5-7] 과목등록 화면

2.2 수강학생 등록

[그림 5-8] 학생등록 화면

과목번호	과목명	교수 ID	등록	관리
56	내트윅크계	leepro	학생등록(수강생)	등록학생(보기)
56	회계원리	leepro	학생등록(주강생)	등록학생(보기)

문제 출제자는 자기의 과목을 수강하는 학생을 등록하는데 이때 등록할 때 사용하는 ID와 Password는 학번과 주민등록번호 뒷자리 7자리로 한다. 물론 과목을 수강하는 학생의 정보를 수정 또는 삭제할 수 있는 관리기능도 가지고 있다.

2.3 시험문제 출제/관리

학습평가 시스템의 핵심이라고 할 수 있는 메뉴로 온라인 상에서 시험을 칠 수 있도록 시험문제를 입력하며 입력한 문제를 수정 및 삭제할 수 있다. 또한 출제한 문제를 저장해서 다음에 참고할 수 있도록 문제은행의 기능을 가지고 있는 메뉴이기도 하다.

[그림 5-9] 시험 출제시 설정사항 등록 화면

The screenshot shows a web application interface for setting exam questions. On the left is a sidebar menu with a '로그아웃' (Logout) button and a '문제 출제시 메뉴' (Menu for setting exam questions) section containing links for '출제 목록' (List of questions), '수강 학생 등록' (Register student), '시험 출제/삭제/관리' (Set/delete/manage exam), and '문제 처리 및 관리' (Process and manage questions). The main content area is titled 'home/학사 관리/문제 처리 및 관리' and contains the following fields and controls:

- 시험명** (Exam Name): A dropdown menu currently showing '중간고사' (Midterm Exam).
- 시험날짜** (Exam Date): A date picker set to 2002.06.17.
- 시각시간** (Start Time): A time picker set to 02:05.
- 종료시간** (End Time): A time picker set to 02:55.
- 선택형인** (Multiple Choice): Two radio buttons, '바로 보기' (View immediately) and '이정된 시간' (Set time), with the latter being selected.
- Buttons**: '저장' (Save) and '취소' (Cancel) buttons at the bottom.

또한 정해진 시간에 시험에 응시할 수 있고 시험 시작 후 일정 시간이 흐르면 시험에 응시 할 수 없으며 시험 결과를 언제 공고하겠

다는 것을 옵션으로 선택할 수 있다.

2.4 성적 처리 및 관리

시험에 응시하고 나면 시험결과는 채점이 되어서 자동으로 데이터 베이스에 먼저 저장이 된다. 시험 외에 성적에 관여하는 요소(출석률, Report 등)의 성적을 직접 입력할 수 있는 메뉴이다. 물론 시험 성적도 여기서 수정 및 삭제가 가능하게 된다. 이 메뉴에서는 과목 등록시 입력한 점수 배점에 따라 합산된 성적 총합계를 구할 수도 있다.

[그림 5- 10] 시험 출제를 위한 C#, ASP.NET 코드

```
//문제 입력하는 코드
private void Button1_Click(object sender, System.EventArgs e)
{
    exam_num = Int32.Parse(Request["exam_num"].ToString());
    sqlCommand1.Parameters["@exam_num"].Value = exam_num;
    sqlCommand1.Connection.Open();
    sqlCommand1.CommandText = "Insert into questions
(que_num, content, example_list1,example_list2, example_list3,
example_list4, answer, exam_num) values
('"+ TextBox1.Text+ "','"+ TextBox2.Text+ "','"+ TextBox3.Text+ "','"+
TextBox4.Text+ "','"+ TextBox5.Text+ "','"+ TextBox6.Text+ "','"+ Tex
tBox7.Text+ "','"+ exam_num+ ")";
    sqlCommand1.ExecuteNonQuery();
}
```

```

        sqlCommand1.Connection.Close();

        Response.Redirect("questions_list.aspx");
    }

    //취소를 클릭했을 때
    private void Button2_Click(object sender, System.EventArgs e)
    {
        TextBox1.Text = "";
        TextBox2.Text = "";
        TextBox3.Text = "";
        TextBox4.Text = "";
        TextBox5.Text = "";
        TextBox6.Text = "";
        TextBox7.Text = "";
    }

```

성적처리 - 과목을 입력할 때 성적 배점을 입력하면 시험(중간고사, 기말고사) 성적은 시험 종료와 함께 채점되어 저장되고, 출석점수와 리포트 점수는 문제 출제자가 입력을 하게 되면 처음에 입력된 성적 배점에 따라 100점 만점으로 처리되는 부분이다.

$$((\text{attendanc}/10)*\text{attendanc_point}) + ((\text{report}/10)*\text{report_point}) + ((\text{middle_ex}/10)*\text{middle_point}) + ((\text{end_ex}/10)*\text{end_point})$$

- attendanc : 출석점수
- attendanc_point : 출석점수 배점

- report : 리포트 점수
- report_point : 리포트 점수 배점
- middle_ex : 중간고사 점수
- middle_point : 중간고사 점수 배점
- end_ex : 기말고사 점수
- end_point : 기말고사 점수 배점
- 10점 만점으로 환산 해서 점수를 나누고 점수 배점을 곱해서 100점으로 환산

[그림 5-11] 성적처리를 위한 C#, ASP.NET 코드

```

<< 생 략 >>

//성적 총 합계를 구하는 부분
sqlCommand1.Parameters["@sub_num"].Value = sub_num;
sqlCommand1.Connection.Open();

SqlDataReader reader = sqlCommand1.ExecuteReader();
while(reader.Read())
{
    double attendanc = double.Parse(reader["attendanc"].ToString());
    double report = double.Parse(reader["report"].ToString());
    double middle_ex = double.Parse(reader["middle_ex"].ToString());
    double end_ex = double.Parse(reader["end_ex"].ToString());

    Response.Write("<table border = '1'>");

    Response.Write("<tr><td>학번</td><td>출석점수</td><td>리포트점
```

```

수</td><td>중간고사</td><td>기말고사</td><td>총점</td><td>&nbsp;</td></tr>");

    Response.Write("<tr><td>" + reader["st_id"] + "&nbsp;</td>");
    Response.Write("<td>" + attendanc + "&nbsp;</td>");
    Response.Write("<td>" + report + "&nbsp;</td>");
    Response.Write("<td>" + middle_ex + "&nbsp;</td>");
    Response.Write("<td>" + end_ex + "&nbsp;</td>");

    double dou =
        ((attendanc/10)*attendanc_point) + ((report/10)*report_point) + ((middle_ex/10)*middle_point) + ((end_ex/10)*end_point);
    Response.Write("<td>" + dou + "&nbsp;</td>");
    Response.Write("<td><a href =
        'editScore.aspx?st_id=" + reader["st_id"] + "'>수정</a></td></tr>"
    );}

    Response.Write("</table>");
    sqlCommand1.Connection.Close();

```

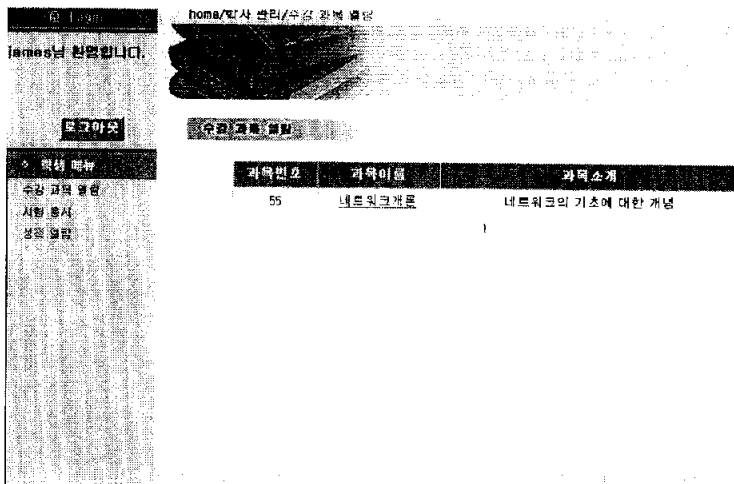
제 3 절 학생 메뉴

승인 받은 출제자가 출제한 문제를 정해진 시간에 학습평가 시스템에 접속해서 시험에 응시하고 성적을 확인할 수 있다.

3.1 수강과목 열람

문제 출제자가 등록한 학생으로 로그인 하고 수강과목 열람을 선택하게 되면 현재 등록된 수강과목 목록이 보인다. 여기서 수강과목 이름을 다시 선택하면 그 과목에 대한 개요, 학습 계획서, 성적 배점 등을 열람할 수 있다.

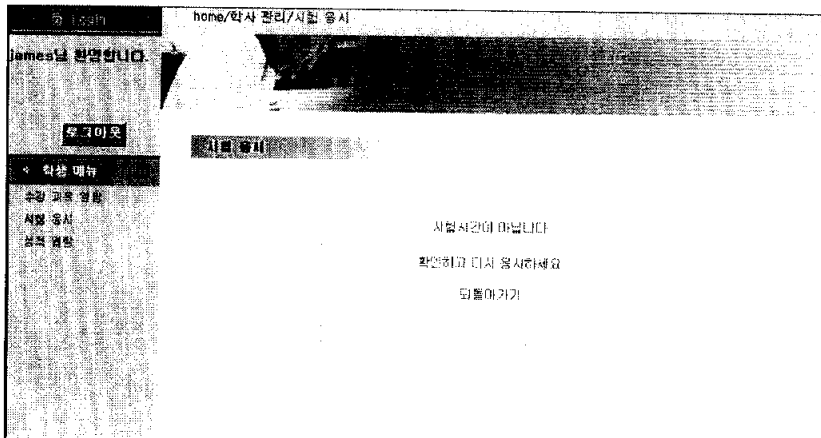
[그림 5-12] 수강과목 열람 화면



3.2 시험응시

문제 출제자가 공고한 시험 시간에 학습평가 시스템에 접속하면 시험을 칠 수 있는 목록이 나타난다. 거기서 내가 원하는 시험을 클릭하게 되면 시험이 시작된다. 끝내고 확인 버튼을 클릭하게 되면 자동적으로 채점이 되어서 데이터베이스에 저장되고 시험리스트 페이지로 돌아가게 된다. 문제 출제자가 시험을 공고하면서 정한 옵션에 따라 시험에 응시할 수 있다.

[그림 5-13] 시험응시 화면



또한 지정된 시험시간이 아니면 학생은 시험에 응시할 수 없게 된다. [그림 5-13]은 응시가자가 시험시간이 아닌 시간에 시험응시를 할려하면 나타나는 화면이다.

3.3 성적열람

성적 열람을 클릭하게 되면 내가 수강하고 있는 과목 목록에서 원하는 성적을 열람 할 수 있는 메뉴로써 시험 출제자가 시험 공고하면서 설정한 옵션(언제부터 성적 공고하겠다는)에 따라 성적을 열람할 수 있다.

[그림 5-14] 시험응시를 위한 C#, ASP.NET 코드

```

<< 생 략 >>

//응시자 답을 저장하는 부분
int summ;
```

```

int z;
int a;
ArrayList arl = new ArrayList();
ArrayList ar = new ArrayList();
string exam_name;
int sub_num;
private void Button1_Click(object sender, System.EventArgs e)
{
    for(int i = 0; i < DataList1.Items.Count; i++)
    {
        string b1 =
        ((RadioButton)DataList1.Items[i].FindControl("RadioButton1")).Checked
        .ToString();

        string b2 =
        ((RadioButton)DataList1.Items[i].FindControl("RadioButton2")).Checked
        .ToString();

        string b3 =
        ((RadioButton)DataList1.Items[i].FindControl("RadioButton3")).Checked
        .ToString();

        string b4 =
        ((RadioButton)DataList1.Items[i].FindControl("RadioButton4")).Checked
        .ToString();

        if(b1.Equals("True")) z = 1;
        else if(b2.Equals("True")) z = 2;
        else if(b3.Equals("True")) z = 3;
        else if(b4.Equals("True")) z = 4;
    }
}

```

```

        Else    z = 0;

        arl.Add(z); //arraylist에 답 담기
    }

```

시험을 응시하면 출제자의 설정에 따라 채점을 하게 된다. 채점을 하는 코드는 다음과 같다.

[그림 5-15] 채점을 위한 C#, ASP.NET코드

```

//정답을 가져오는 부분

        sqlCommand2.Connection.Open();

        sqlCommand2.Parameters["@exam_num"].Value =
Int32.Parse(Request["exam_num"].ToString());

        SqlDataReader reader = sqlCommand2.ExecuteReader();
        while(reader.Read())

        { a = Int32.Parse(reader["answer"].ToString());
          ar.Add(a); }

        sqlCommand2.Connection.Close();

//응시자 답과 정답을 비교하는 부분

        for(int g = 0; g < arl.Count; g++)

        { if(ar[g].Equals(arl[g]))

            { summ += 1;

            }

        }

        string st_id = Session["key"].ToString();

```

채점이 끝나면 이를 MS SQL 데이터베이스에 저장하여야 한다.
다음은 채점된 결과를 저장하는 코드이다.

[그림 5-16] 채점결과를 DB에 저장하기 위한 C#, ASP.NET 코드

```
//시험 결과를 데이터 베이스에 저장하는 부분

    int exam_num = Int32.Parse(Request["exam_num"].ToString());
    sqlCommand1.Parameters["@exam_num"].Value = exam_num;
    sqlCommand1.Connection.Open();
    SqlDataReader reader1 = sqlCommand1.ExecuteReader();
    while(reader1.Read())
    { exam_name = reader1["exam_name"].ToString();
      sub_num =
Int32.Parse(reader1["sub_num"].ToString()); }
      sqlCommand1.Connection.Close();
      if(exam_name.Trim().Equals("중간고사"))
      { sqlCommand3.CommandText = "Insert into
grade_control(st_id, middle_ex, sub_num)
Values('"+ st_id+ "',"+ summ+ ", "+ sub_num+ ")"; }
      else
      { sqlCommand3.CommandText = "Update grade_control set
end_ex = '"+ st_id+ "'"; }
      sqlCommand3.Connection.Open();
      sqlCommand3.ExecuteNonQuery();
```

```
sqlCommand3.Connection.Close();  
  
Response.Write("수고하셨습니다.");  
// 시험이 끝나고 다시 시험 리스트로 돌아가는 부분  
Response.Redirect("subList.aspx");}
```

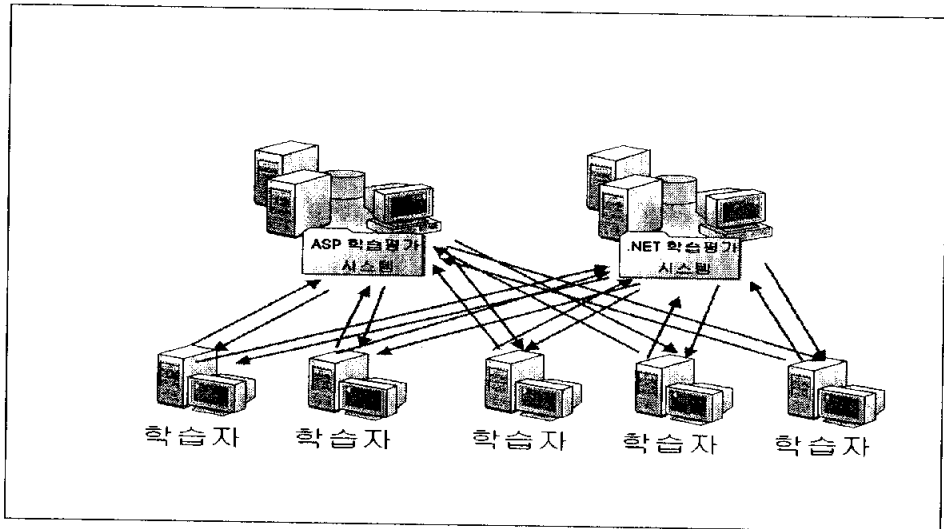
제 4 절 .NET과 ASP를 이용한 학습평가 시스템의 비교

4.1 시스템 환경

다른 언어를 이용하여 구축된 학습평가 시스템과 .NET을 이용한 학습평가 시스템과의 비교를 위해 기존의 연구들이 많이 채용하였던 ASP를 이용하여 시스템을 구축하였다. ASP를 이용한 학습평가 시스템은 서버 운용 체제로서 Windows 2000 에 IIS 5.0 웹 서버, 그리고 ASP와 SQL서버 2000을 사용하였다. 이것은 .NET을 이용한 학습평가 시스템과 구현 언어가 다르다는 점만 빼고는 동일한 환경이다.

시스템의 비교는 같은 강의실 환경에서 학습평가 시스템에 접속하는 경우를 가정하였다. 실험 참가 학습자는 30명이고 동일한 사양의 클라이언트로 서버에 접속하도록 하였다. 구성도는 [그림 5-17]과 같다.

[그림 5-17] .NET과 ASP 학습평가시스템 비교를 위한 테스트 환경 구성도



CPU의 속도는 Pentium III 1GHz 이고, 메모리는 512MB를 사용하였고, 클라이언트 브라우저는 마이크로소프트의 인터넷 익스플로러 6.0버전을 사용하였다. 시스템의 응답시간을 측정하는 도구는 Quest Software사의 Benchmark Factory라는 제품을 이용하였다. 그리고 서버의 부하를 측정하는 도구로는 Windows 2000 Server에 내장되어 있는 성능감시 모니터를 이용하였다. 실험시간은 응시자들이 학습평가 시스템에 접속하여 로그아웃할 때까지 실험시간으로 하였다. 공정한 테스트를 위하여 ASP를 이용한 학습평가 시스템과 .NET을 이용한 학습평가 시스템은 모두 하나의 스위치에 연결되었고 30명의 학습자들은 같은 강의실에서 실험토록 하였다.

4.2 실험결과

실험결과인 [표 5-1]을 보면 ASP를 이용한 학습평가 시스템에서

는 30명 접속시 약 10% 정도의 CPU 사용량을 나타냈다. 그러나 .NET을 이용한 학습평가 시스템에서는 5%의 낮은 CPU 사용량을 나타내어 ASP보다는 .NET을 이용한 학습평가 시스템이 더 안정적으로 시스템 자원을 활용하는 것을 알 수 있었다.

[표 5-1] ASP와 .NET의 CPU사용량 비교

사용자 부하	ASP CPU 사용 %	.NET CPU 사용 %
30	10%	5%
100	15%	7%

그리고 속도의 비교를 위하여 Benchmark Factory를 이용하여 100명의 가상 사용자를 로그인 시켜서 데이터베이스에서 30개의 행을 가져오는 페이지를 실행시켜 보았다. [표 5-1]을 보면 ASP를 이용한 학습평가 시스템은 평균 0.310초가 걸린 반면 .NET을 이용한 학습평가 시스템에서는 평균 0.260초가 걸려 보다 빠르게 반응함을 알 수 있었다.

[표 5-1] ASP와 .NET의 사용자 응답 시간

사용자 부하	ASP 응답 시간 (초)	.NET 응답시간
30	0.251	0.225
100	0.310	0.260

이번 실험결과는 .NET을 이용하여 구축한 학습평가 시스템이 ASP를 이용한 학습평가 시스템보다 CPU사용량이 적어 안정적이면서도 더 빠른 속도를 보장한다는 것을 보여주었다.

제 6 장 결 론

최근 많은 각광을 받고 있는 e-learning은 앞으로 기업체 재교육 시장, 사이버 교육의 확대, 일반인 교육 등으로 분야를 넓혀가며 더욱 확장될 것으로 보인다. 이러한 e-learning은 제 2장에서 기술한 바와 같이 시간적, 공간적, 비용적 측면뿐만 아니라 교육 콘텐츠의 신속한 업데이트, 학습자의 참여도 향상 등의 이점으로 계속적으로 발전해 나갈 것이다. 이와 동시에 학습의 내용이 올바르게 전달되고 학습자가 이 내용을 습득하기 위하여 얼마나 노력하였는가를 평가하는 학습평가 시스템의 중요성도 커질 것이다. 또한 많은 학습자를 동시에 시험에 응시하게 하고 이를 시스템 오류 없이 원활히 처리하기 위하여 신속한 처리속도와 대용량 처리능력을 갖춘 시스템이 중요한 역할을 하게 될 것이다.

본 연구에서는 이를 위하여 .NET 기술을 이용하였고 C# 언어와 ASP.NET 그리고 SQL Server 2000 데이터베이스를 사용하였다. 논리적으로는 시스템을 사용할 관리자, 시험출제자, 학생(시험응시자)의 세가지 모듈로 나누었다. 관리자모듈은 시험을 출제할 수 있는 교수(출제자)에게 ID와 패스워드를 발급하는 기능과 교수 정보관리, 학생정보관리, 성적관리를 할 수 있는 기능을 두었고 시험출제자 모듈은 과목을 등록하고 수강학생을 등록할 수 있게 하였다. 또한 시험문제를 출제하고 관리할 수 있는 기능을 두었다. 마지막으로 학생모듈은 자신이 수강하는 과목의 목록을 열람하고 시험에 응시하고 성적을 확인 할 수 있는 기능을 가지도록 하였다.

이 논문에서 연구하고자 했던 부분은 .NET을 이용하였을 때 다른 언어로 구현된 학습평가 시스템보다 더 빠르게 학습자가 사용할

수 있고 더욱 안정적인 시스템을 구현할 수 있는냐는 점이였다. 이를 위하여 ASP를 이용하여 동일한 학습평가 시스템을 구축하여 비교실험을 한 결과 다음과 같은 사항을 확인하였다.

첫째, .NET을 이용한 개발로 다수의 동시 사용자 요구를 처리할 수 있게 하여 시스템의 안정성을 가져 왔다. ASP로 만들어진 학습평가 시스템의 경우 100명 정도의 사용자가 동시에 접속하여 사용할 때 CPU 사용률이 15%를 넘어서는 현상을 보였지만 .NET을 이용한 학습평가 시스템은 7%의 낮은 CPU 사용량을 나타냈다.

둘째, .NET을 이용한 학습평가 시스템에서 데이터를 가져올 때는 ASP를 이용한 학습평가 시스템보다 속도가 빨랐다. 이는 다수의 사용자가 접속할수록 차이가 나는 것으로 나타났다.

이러한 비교 결과에 의하여 .NET을 이용한 학습평가 시스템은 다수의 동시접속 사용자에게 빠른 속도와 낮은 CPU 사용량으로 더 안정적인 서비스가 가능하는 것을 알수 있게 되었다.

마지막으로 향후 연구과제는 .NET을 이용한 학습평가 시스템의 특성상 가지게 되는 공간적 제약을 극복하는 것이 될 것이다. 이를 해결하기 위하여 앞으로는 모바일 기기 즉, PDA와 같은 기기를 이용하여 시험을 응시할 수 있는 시스템의 연구가 필요할 것이다.

참 고 문 헌

- [1] 김덕중, 김연주 “e-learning 기획실무 스타일가이드”, 2002.
- [2] 김병곤, 김종욱, “멀티미디어 정보시스템을 이용한 기업체 사원 교육의 효과에 영향을 미치는 요인에 관한 연구”, 경영학연구, pp. 400 ~ 540, 2000.
- [3] 나일주, “Web 기반 교육”, 교육과학사, 1999.5.
- [4] 남윤희, ”원격 교육을 위한 평가시스템 설계 및 구현”, 신라대학교 교육대학원 석사학위 논문, 2001.
- [5] 노일순, “Servlet과 JDBC를 사용한 가상교육 평가시스템 구축에 관한 연구”, 서강대학교 정보통신대학원 석사논문, 1999.
- [6] 박성순, 김성규, 김우분, ”멀티미디어 데이터베이스를 기반으로 한 가상대학의 구축,” 정보과학회지, 제 14권 12호, pp.5-15, 1996년 12월.
- [7] 배상현, “Web 기반 원격교육을 위한 실시간 평가시스템의 설계 및 구현”, 경상대학교 대학원 석사학위 논문, 1998.
- [8] 백영균, “Web 기반 학습의 설계”, 양서원, 2002.3.

- [9] 백영균, 강신천, “웹 기반 학습환경의 준비와 개발도구”, 원미사, 1999.11.
- [10] 신경아, “원격 평가 시스템의 설계 및 구현”, 한국교원대학교 교육대학원 석사학위 논문, 2001.
- [11] 양영선, 조은순(역), “원격교육의 이해와 적용”, 예지각, 1998.4.
- [12] 이석호, “인터넷 환경의 대화형 학습평가 시스템 설계 및 구현”, 부경대학교 교육대학원 석사학위 논문, 1998.
- [13] 이진경, “Web 기반 학습을 위한 평가시스템의 설계 및 구현”, 서울교육대학교 교육대학원 석사논문, 2000.
- [14] 이태욱, “컴퓨터 교육론”, 도서출판 좋은 소프트, 1999.3.
- [15] 임정훈, “효율적인 가상수업 구축을 위한 가상교육 플랫폼의 분석과 선정”, 한국컴퓨터교육학회 논문지 제2권 제4호, pp. 119-129, 1999.
- [16] 유인출, “성공적인 e-learning 비즈니스 전략”, 이비컴, 2001.4.
- [17] 정재삼, 임규연, “웹 기반 토론에서 학습자의 참여도, 성취도

- 및 만족도 관련 요인의 효과 분석, 교육공학연구, 제 16권 2호,
pp. 107-135, 2000.
- [18] 정인성, “원격교육의 이해”, 교육과학사, 1999.
- [19] 정원혁, “Microsoft 전문가로 가는 지름길① SQL Server
2000”, 대림출판사, 2001.
- [20] 하성광, “클릭하세요. 한글비주얼C#”, 대림출판사, 2001.10.
- [21] 하성광, “클릭하세요 ASP.NET”, 대림출판사, 2002.3.
- [22] 마이크로소프트 Developer Network, .NET Online 정보페이지,
“<http://www.microsoft.com/korea/msdn/net/>”
- [23] Microsoft Developer Network, “Microsoft .NET vs. Sun
Microsystem J2EE 벤치마크”,
[http://www.microsoft.com/korea/msdn/net/compare
/default.asp](http://www.microsoft.com/korea/msdn/net/compare/default.asp)
- [24] 태오의 ASP 홈페이지, 닷넷온라인강좌, “[http://www.taeyo.
pe.kr/lecture/list_AspNet.htm](http://www.taeyo.pe.kr/lecture/list_AspNet.htm)”
- [25] Andrew G. Duthie, “Microsoft ASP.NET Step By Step”,
Microsoft Press, 2001.12.

- [26] Badrul H., Khan, "Web-based Instruction", Educational Technology Publication, 1996.
- [27] Driscoll, Margaret M., "The application of adult education principles in the development of a manual for practitioners creating Web-based training programs.", Columbia University Teachers College.(0055), 1999.
- [28] Jung, I., "Building a theoretical framework of web-based instruction. British Journal of Educational Technology, 2001.
- [29] Karli Watson, "Beginning C#", Wrox Press, 2001.9.
- [30] Larry Alexander, Margaret Driscoll, "Web-Based Training", Jossey-Bass, 1998.
- [31] Ollie Cornes, Chris Goode, Rob Birdwell, "Beginning ASP.NET Using C#", Wrox Press, 2001.11.
- [32] Salmon, G. & Almog, T., "Educational psychology and technology: A matter of reciprocal relations." Teachers College Record, 100(2), 1