

서 동 수 교수지도
석사학위 청구논문

객체지향 테스트기법을 적용한
웹 교육시스템의 설계 및 구현

2006

성신여자대학교 교육대학원
교육학과 전자계산교육전공
이 혜 리

객체지향 테스트기법을 적용한
웹 교육시스템의 설계 및 구현

서 동 수 교수지도

이 논문을 석사학위 논문으로 제출함.

2005년 11월

성신여자대학교 교육대학원

교육학과 전자계산교육전공

이 혜 리

인 준 서

이혜리의 석사학위 논문을 인준함.

심사위원 (인)

심사위원 (인)

심사위원 (인)

성신여자대학교 교육대학원

논문 개요

정보통신기술의 발전과 웹의 급속한 성장으로 기존의 시스템들이 웹을 기반으로 통합되고 있다. 교육 분야 역시도 이러한 변화를 적극적으로 수용하여 웹을 통한 온라인/오프라인 교육에 많은 노력을 기울이고 있다.

웹 기반교육(Web based Instruction: WBI)은 단순히 기존 교육의 디지털화를 넘은 새로운 교육의 형태이며, 그에 맞게 교수-학습자간의 상호작용에 의한 효율적인 학습형태가 이루어져야 한다. 또한 이러한 요구사항은 시스템에 정확히 반영되어야하고, 시스템의 효율적인 개발과 유지보수가 뒷받침되어야 한다. 이러한 방향에서 웹기반 시스템에 대한 효율적인 개발과 유지보수 및 신뢰할 수 있는 시스템을 구축하기 위해서는 웹기반의 테스트가 요구된다.

본 논문에서는 전통적인 클라이언트-서버 시스템에 비해 더 복잡해지고 다양해진 사용자의 요구가 반영된 웹기반 교육시스템 개발을 위해서 적합한 테스트 설계 방법을 제안한다. 이를 위해 구현된 C언어 기초교육을 위한 시스템에 대해 유스케이스 다이어그램과 시퀀스 다이어그램을 바탕으로 실제 테스트 케이스 설계 및 테스트를 적용한다. 본 논문에서 추출한 정보들은 향후 유사한 형태의 웹기반 교육 시스템의 테스트를 위한 기초정보로 활용될 수 있다.

목 차

논문개요

제 1 장 서론	1
제 2 장 관련 연구	3
2.1 웹 기반교육(WBI)의 배경	3
2.1.1 웹 기반교육의 개념	3
2.1.2 웹 기술과 WBI의 관계	4
2.2 웹 기반 테스트	6
2.2.1 테스트의 필요성	6
2.2.2 웹 테스트의 특성	8
2.2.3 웹 테스트를 위한 지원 방법 및 도구	9
제 3 장 웹 교육시스템 설계	11
3.1 시스템의 개요 및 구성	11
3.2 시스템 설계	13
3.2.1 모듈설계	13
3.3 데이터베이스 설계	16
3.3.1 강의관련 테이블	17
3.3.2 회원 테이블	18
3.3.3 시험평가 관련 테이블	18

제 4 장 테스트 설계	21
4.1 테스트 방법 및 전략	21
4.2 테스트 케이스의 설계	24
4.2.1 유스케이스를 이용한 기능테스트	24
4.2.2 순차도를 이용한 화이트박스 테스트	27
4.2.3 폼 입력을 통한 블랙박스 테스트	31
제 5 장 시스템 구현 및 평가	34
5.1 시스템 구현	34
5.1.1 시스템 구성	34
5.1.2 개발환경	34
5.1.3 시스템 구현	35
5.2 테스트 케이스에 기반한 시스템 테스트	41
5.2.1 유스케이스를 이용한 기능테스트	41
5.2.2 순차도를 이용한 화이트박스 테스트	45
5.2.3 폼 입력을 통한 블랙박스 테스트	47
5.2.4 테스트케이스 실행 에러처리	48
5.3 시스템의 평가 및 활용방안	48
제 6 장 결론	50

참고문헌

ABSTRACT

표 목 차

(표 2-1) 웹 테스트 도구	10
(표 3-1) 강좌 Master Table	17
(표 3-2) 강좌 내용 Table	17
(표 3-3) 사용자 강의 기록 Table	18
(표 3-4) 회원테이블	18
(표 3-5) 강좌 테스트 문제 테이블	19
(표 3-6) 사용자가 풀어본 테스트 문제 저장 테이블	19
(표 3-7) 사용자의 테스트 로그 저장 테이블	20
(표 4-1) 강의참여 유스케이스	25
(표 4-2) 강의참여 유스케이스의 테스트 시나리오	26
(표 4-3) 강의참여 유스케이스의 테스트 케이스	27
(표 4-4) 등록 유스케이스 기술서	32
(표 4-5) 등록 유스케이스의 테스트 케이스	33
(표 5-1) 하드웨어 환경	35

(표 5-2) 소프트웨어 환경	35
(표 5-3) 강의참여 유스케이스의 테스트 케이스	42
(표 5-4) Test Case/Log	45
(표 5-5) 등록 유스케이스의 테스트 케이스의 실행 결과표	47
(표 5-6) 테스트 오류처리 결과 보고서	48

그림 목차

(그림 2-1) 3-티어 클라이언트-서버 아키텍처	6
(그림 2-2) 네 개의 주요 소프트웨어 테스트 생명주기	7
(그림 3-1) 학습지원 시스템의 프로세스	12
(그림 3-2) 모듈 구조도	13
(그림 3-3) 회원인증 모듈 구조	14
(그림 3-4) 강의참여 모듈 구조	15
(그림 3-5) 학습자의 강의참여 흐름도	15
(그림 3-6) 시험평가 모듈 구조	16
(그림 4-1) 블랙박스 테스트와 화이트박스 테스트	22
(그림 4-2) 동적 테스트 프로세스	23
(그림 4-3) 강의참여 유스케이스의 이벤트 흐름	26
(그림 4-4) 강의참여에 대한 시퀀스 다이어그램	29
(그림 4-5) 학습자 수강 진행상태 시퀀스 다이어그램	29
(그림 4-6) 학습자 수강 마지막 페이지에 대한 시퀀스 다이어그램	30
(그림 4-7) 학습자 테스트 시퀀스 다이어그램	30
(그림 4-8) 보충학습 시퀀스 다이어그램	31
(그림 5-1) 시스템 구성	34
(그림 5-2) 초기화면	36

(그림 5-3) 강의리스트 출력화면	36
(그림 5-4) 강의참여 화면	37
(그림 5-5) 시험평가확인 리스트 화면	37
(그림 5-6) 시험평가 상세정보 화면	38
(그림 5-7) 강의관리 화면	39
(그림 5-8) 강의등록 화면	39
(그림 5-9) 시험문제 등록 화면	40
(그림 5-10) 시험문제 상세보기 화면	40
(그림 5-11) 학습자 관리 화면	41
(그림 5-12) 학습자 정보 수정 화면	41
(그림 5-13) 수강완료 테스트 화면	43
(그림 5-14) 테스트 페이지 화면	43
(그림 5-15) 정답확인 화면	44
(그림 5-16) 보충학습 메시지 화면	44
(그림 5-17) 보충학습 화면	45
(그림 5-18) log.txt 파일에 기록된 사용자 정보	46
(그림 5-19) 로그인 화면	47

제 1 장 서론

웹의 급속한 성장과 함께 비즈니스, 지식, 경영, 마케팅, 전자상거래 등 기존의 다양한 IT 시스템들이 웹을 기반으로 변화되어 가고 있다. 교육도 웹으로 확장되고 있는 추세이며, 웹기반의 교수학습 방법인 WBI(Web Based Instruction)형태로 발전하면서 원거리의 학습자를 교육시키는 유용한 방법으로 알려지고 있다. 웹 기반 교육은 전통적인 교실 수업에 비해 수요자 중심의 학습 방법으로 교육 현장에서 인정되고 있지만, 교육과정 면에서 다양한 사용자의 요구를 수렴하기 어렵고, 학습자의 모든 피드백(feed-back) 상황을 만들어 주지 못하고 있다. 또한 기존의 단순한 HTML 위주의 교육 시스템으로는 학습자의 요구를 충분히 반영하지 못하므로, 교수-학습자간의 상호 작용에 의한 효율적인 학습형태가 이루어지기 위해서는 요구사항을 시스템에 정확히 반영되어야 한다.

시스템의 효율적인 개발과 유지보수는 테스트를 통해 가능하다. 학습자의 요구에 부합된, 그리고 효율적인 웹 교육시스템의 구성을 위해서는 구현 시스템에 대한 테스트가 이루어져야 한다.

웹을 기반으로 한 통합 개발과정에서 필연적으로 파생되어지는 복잡함은 개발 및 유지보수를 더욱 어렵게 한다. 일반적인 어플리케이션과는 달리 웹 어플리케이션들은 다양한 기술의 접목으로 개발되어지고 있으며, 이로 인해 복잡하고 방대해진 시스템의 개발 및 관리에 있어 어려움은 더욱 증대되고 있다. 또한 웹 기반 시스템에 대한 테스트 뿐만 아니라 개발과정 자체도 일반적인 소프트웨어의 개발 과정과 많은 차이를 보인다. 따라서 기존의 구조적 코

딩/개발방법이 아닌 객체지향 기술이 사용된 설계코딩 패러다임의 변화에 대한 테스트 방법 및 기준 또한 필요하다.

본 논문에서는 더 복잡해지고 다양해지는 웹을 기반으로 한 교육 시스템의 효율성, 신뢰성, 정확성, 기능성 등을 테스트를 통해서 검증해 보고 그 중요함을 보이하고자 하며, 이러한 웹 기반 시스템에 대한 효율적인 개발과 유지보수 및 확장에서 신뢰성을 유지하기 위한 시스템 테스트 방법을 제시하고자 한다.

본 논문의 구성은 총 6장으로 구성되며 2장에서는 관련연구로 웹 기반교육의 정의 및 특징에 대한 요약, 기존 테스트 방법과 테스트의 필요성, 특히 웹 테스트에 관하여 살펴본다. 3장에서는 웹 교육 시스템을 설계하고, 4장에서 구체적인 테스트 전략과 함께 이를 바탕으로 한 테스트 케이스를 설계한다. 5장에서는 시스템 구현에 대한 설명과 4장에서 설계한 테스트 케이스를 바탕으로 시스템을 평가한다. 마지막으로 6장은 결론과 향후 연구방향을 제시한다.

제 2 장 관련 연구

2.1 웹 기반교육(WBI)의 배경

2.1.1 웹 기반교육의 개념

정보통신 기술의 발전과 급격한 인터넷 보급률 증가는 시간과 공간의 제약을 완화시키고 있으며 사회전반의 시스템에 영향을 미치고 있다. 교육에 있어서도 교육방식, 교육체계, 교수의 역할, 수업진행 및 평가방법 체계 등에 많은 변화를 가져오기에 이르렀다[1]. 이러한 변화를 적극적으로 수용하기 위해 최근 컴퓨터 및 인터넷을 활용한 다양한 형태의 교육방식들이 제안되고 있으며, 그중 대표적인 것이 WBI(Web Based Instruction)이다.

웹 기반 교육은 웹의 속성과 자원을 이용함으로써 편리한 학습 환경을 제공하는 하이퍼미디어 기반의 교육 프로그램이다[2]. WBI는 인터넷의 발달과 더불어 여러 가지 교육방식에 적용되고 있으며 최근에는 교육평가 시스템에 WBI를 적용하고자 하는 시도들이 많이 이루어지고 있다[3].

웹을 활용하는 교육은 전통적인 교실수업을 통해서 획득할 수 있었던 학습 성과는 물론, 학습자에게 시간과 장소에 구애받지 않는 학습기회를 부여하고, 동시에 교수자와 학습자, 동료학습자간에 다양한 상호작용의 기회를 제공해주므로, 의사소통기술 뿐만 아니라 문제해결 능력과 학습 능력을 높이는데 중요한 역할을 한다. 웹기반 교육의 교육적 유용성 중에서 많은 사람들이 가장 핵심적인 특징으로 지적하는 것은 다양한 형태의 상호작용 기회를 제공한다는 점이다.

기존의 학습과는 달리 웹 기반 교육은 다음과 같은 독특한 교육적 특징을 갖고 있다[4].

첫째, 웹을 활용한 교육은 어떤 통신 수단보다도 많은 양의 최신 정보를 빠른 시간 내에 교류할 수 있도록 함으로써 학교체제와 같이 외부 정보의 습득에 뒤지기 쉬운 사회에 효과적인 정보 교류의 수단을 제공한다.

둘째, 웹을 활용한 교육은 기존의 단방향 매체들과는 달리 고도의 상호작용적 의사소통을 가능하게 해준다. 학습자들은 컴퓨터 통신을 이용하여 다른 학습자나 교수자, 혹은 다른 전문가들과 정보나 의견을 교환하고, 온라인 토론 등을 통해 창의적이면서도 활발한 상호작용 활동을 수행할 수 있다.

셋째, 웹을 활용한 교육은 기존의 전통적인 교실수업 체제나 면대면 수업, 혹은 전화 통화에서와 같은 동시적 상호작용뿐만 아니라 시간과 공간을 초월한 비동시적 상호작용을 가능하게 해준다.

넷째, 웹을 활용한 교육은 고도의 동시적/비동시적 상호작용을 통해 협력학습 체제를 가능하게 해준다.

다섯째, 웹을 활용한 교육은 독특한 사회심리적 커뮤니케이션 구조를 제공하여 줌으로써 면대면 수업에서 어려운 긍정적 학습 효과를 가져 올 수 있다.

마지막으로, 웹을 활용한 교육은 다른 매체들의 활용보다 교육의 비용효과 면에서 보다 경제적이라는 특징을 갖고 있다.

2.1.2 웹 기술과 WBI의 관계

웹 기반 교육은 수요자 중심의 학습 방법으로 교육현장에서 인정되고 있으며, 열린교육, 평생교육으로 확대할 수 있다는 점에서 웹의 교육적 활용에 대한 관심이 높아지고 있다. 반면에 교육과정 면에서 다양한 수요자의 요구를

수요하기 어렵고, 학습자의 모든 피드백 상황을 만들어 주지 못하고 있으며, 웹 기술에 기초한 가상교육이 교육학상 기존의 교실학습에 비해 과연 효과적인가라는 근본적인 질문이 제기되고 있다[4][5].

웹 기반 교육의 특징이 효과적으로 이루어지기 위해서는 우수한 웹 기반 교수-학습체제 개발과 사용자의 요구사항이 정확히 반영된 교육 시스템의 구축이 무엇보다도 우선이라고 할 수 있다. 이러한 활동들이 이루어지기 위해서는 단순한 HTML 파일이 아니라 웹기반 응용 프로그램과 데이터베이스가 복잡하게 연결되어 있는 설계원리와 엔지니어링 기술이 분명 필요하다. 이러한 웹 기술이 뒷받침 되었을 때 웹기반 교육은 교육 환경을 보다 역동적이고 학습자 중심적이 환경으로 변모시킬 것이다.

웹 어플리케이션은 단순한 응용 프로그램보다 다양한 콘텐츠가 존재한다는 특징이 있으며, 자원의 공유로 시작한 웹이 최근 E-비즈니스로 발전하면서 구성요소도 다양하게 되었다. 이러한 웹 기반 어플리케이션들은 웹 브라우저, 웹 서버 그리고 어플리케이션 서버와 같은 웹 기술들을 이용하여 개발된다. 웹 시스템은 간단한 텍스트로부터 그래픽, 오디오, 비디오, 콘텐츠 등을 공유하기 위한 자원뿐만 아니라 분산 시스템을 이루는 다양한 소프트웨어 컴포넌트들이 존재한다[7].

일반적으로 웹 어플리케이션들은 시장의 요구에 따라 매우 짧은 개발 기간을 통해 상품화되는 경우가 많으며, 빈번한 개발자의 교체로 인해 소프트웨어 유지보수에 필요한 모듈화, 구조화가 되어있지 않다[8]. 또한 웹과 관련된 기술들이 빠르게 발전한다는 점에서 기존에 만들어진 웹 어플리케이션들의 생명주기가 매우 짧다는 특징을 가지고 있고, 웹을 통해 접근되는 어플리케이션으로 임의의 사용자가 어디에서나 어플리케이션의 제공 능력을 활용할 수 있다는 특징을 가지고 있다. 웹 기반 클라이언트-서버 시스템에

서 어플리케이션은 일반적으로 (그림 2-1)과 같이 세 개의 논리적 세그먼트, 즉 티어들로 분할된다[9].



(그림 2-1) 3-티어 클라이언트-서버 아키텍처

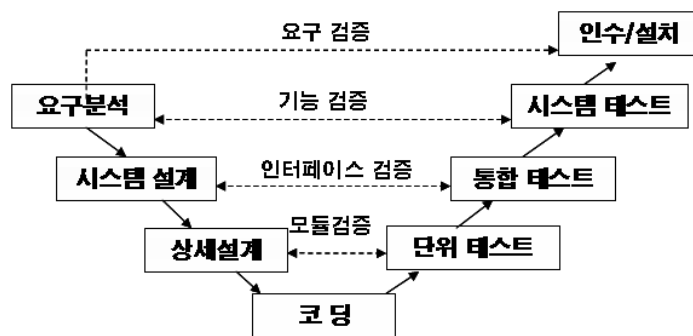
2.2 웹 기반 테스트

2.2.1 테스트의 필요성

웹의 급속한 성장 및 웹 관련 기술의 발전과 다양한 비즈니스 요구사항은 많은 전통적인 어플리케이션이 웹 기반 어플리케이션으로 이행하게 했다[10]. 특히 웹 어플리케이션의 통합 형태인 웹사이트는 파일 중복, 웹문서 양식의 부조화, 휘발성 링크와 같은 근본적인 문제점을 안고 있다.

웹사이트는 소프트웨어 공학에 대한 일반적인 지식이 전혀 없는 사람들조차도 매우 짧은 시간 안에 만들 수 있다. 특히 작은 규모의 개인 웹사이트는 문제점을 가지고 있던지 아니면 업데이트를 언제 하든지 등의 문제점들이 문제가 되지 않는다. 하지만 상업적인 대규모의 사이트는 웹사이트의 유지보수가 사업 성공의 핵심이 된다.

특히 웹을 기반으로 한 교육사이트는 제작된 콘텐츠가 기획 및 설계 단계에서 설정되었던 학습목표 및 코스의 개설 목적 또는 웹사이트 서비스의 목적에 부응하여 제작되었는지, 설계단계에서 제시되었던 상호작용 및 교수학습 전략이 제대로 구현되었는지에 대한 평가가 매우 중요하며, 각 단계별로 이에 대한 평가 준거에 의해 검토되어 지는 것이 시간 및 비용 등의 제한된 자원을 효율적으로 운용할 수 있고, 시스템의 완성도 및 품질을 제고하는 것이 될 수 있다.



(그림2-2) 네 개의 주요 소프트웨어 테스트 생명주기

(그림 2-2)에서 보여준 것처럼 시스템 개발 생명주기 동안에 대규모의 클라이언트-서버나 웹 어플리케이션의 개발에서 테스트 프로세스는 개발 노력의 후반부, 즉 프로그램들 모두가 프로그래머들에 의하여 코딩된 후 발생하는 사후(after-the-fact)이벤트가 아니라는 점이다. 반대로 테스트 프로세스는 어플리케이션 분석의 초기단계에서부터 시스템 구현과 유지 보수까지 연계된 시스템 개발 생명주기 프로세스에 걸쳐있다[11].

몇몇 연구에서 클라이언트-서버 환경에서 어플리케이션을 테스트 하는 것은 전통적인 메인프레임 문자-기반 개발 환경에서보다 아주 어려운 일이라고 분

명하게 지적하고 있다[11]. 클라이언트-서버 어플리케이션을 테스트 하는 동안 개발자들이 직면하는 여러 과제들 때문에 효과적인 클라이언트-서버 테스트 프로세스를 구축할 필요성이 생겼다.

2.2.2. 웹 테스트의 특성

웹 개발의 특성상 웹 어플리케이션은 데이터베이스, GCI, C++, JAVA, ASP, JSP, DCOM, EJB 등 단일 개발에 대해 여러 가지 언어와 기술이 통합적으로 사용된다는 점에서 웹에 대한 테스트는 일반적인 시스템들과는 다른 방법으로 다루어진다. 또한 웹 기반 시스템에 대한 테스트 뿐만 아니라 개발과정 자체도 일반적인 소프트웨어 개발 과정과 많은 차이를 보인다. 예를 들면 UML과 같은 설계 표현 방법이 웹 기반 시스템에 도입되기 위하여 여러 가지 확장이 필요하다[12]. 테스트 방법도 일반 소프트웨어와 구별되어야 하는 부분이 있는데, 다음과 같은 특성을 가지고 있다[13].

- 구성요소에 맞는 다양한 단위 테스트 방법을 동원하여야 함.
- 단위 테스트에서 통합 테스트로의 전환이 쉽지 않음
(즉 인터페이스 시험이 어려움)
- 기본적으로 이벤트 중심 응용인 웹 소프트웨어는 이벤트의 조합이 다양하므로 이를 테스트하기 위한 노력이 많이 든다.

특히 유지보수 측면에서도 파일중복, 웹 문서 양식의 부조화, 휘발성 링크와 같은 근본적인 문제점을 안고 있다.

2.2.3 웹 테스트를 위한 지원 방법 및 도구

웹의 발전과 함께 웹 기반 시스템들을 평가하기 위한 방법들이 최근 많이 연구되고 있다. 전체 웹사이트를 테스트 가능한 컴포넌트(또는 객체)로 구분하고 이를 반복적으로 테스트함으로써 전체 웹사이트를 테스트하는 방법[14], 접근로그(access log)를 이용하여 통계학적 테스트를 위한 모델을 만들고 에러로그(error log)를 분석하여 신뢰성을 검증하는 연구[15], 웹 어플리케이션을 객체, 행위, 구조적 관점에서 분석하는 방법[16], 웹 페이지를 구성하는 응용 컴포넌트들의 구조를 파악하고 각 페이지를 구동시키는 액션들을 찾아내어 상태 기반의 테스트 데이터를 찾아내는 방법[17][18] 등이 있다.

웹을 기반으로 한 시스템의 평가에 있어서 자동 웹 테스트 툴들은 주요 기능 특성들의 수동 테스트를 보완시키기 위해 포괄적인 방법으로 사용될 수 있다. 특히 웹 어플리케이션 성능 능력을 산정하는데 도움을 준다. 웹 테스트 도구의 종류는 시험 방법에 따라 다음과 같이 나눌 수 있다[19].

- 정적 분석기 - 네비게이션 링크, 철자법, HTML 호환 오류 등을 찾아내기 위한 도구
- 성능 테스트 도구 - 웹 부하 분석 및 시뮬레이션 도구
- GUI 캡처 - 웹 사용자 인터랙션(interaction) 과정을 캡처하고 재생하는 도구
- 런타임 오류 도구
- 웹 보안 시험 도구
- 자바 응용 및 애플릿 시험 도구

최근에는 자동화된 웹 테스트 도구들이 많이 나와 있는데, 실제로 유용한 특정영역은 스트레스 테스트와 부하 테스트, 회귀 테스트, 형상 테스트이다.

다음 (표 2-1)은 웹 테스트 도구 사례이다[19].

(표 2-1) 웹 테스트 도구

도구	기능	플랫폼	제조회사
Astra Site Test (LoadRunner, SiteManager)	스트레스 테스트 웹 통합테스트	Windows	Mercury Unteractive 사
SilkTest	기능 및 리그레이션 테스트	Windows, Unix	Seque
SiteRuler	링크 및 HTML 호환성 체크	Windows, Linux, Solaris	ParaSoft
W3C Validator	온라인 HTML 호환성 및 링크 테스트	-	W3C
VisualTest	GUI 성능테스트	Windows, Unix	Rational
NMAP	네트워크 보안(TCP,패킷, 포트)	Windows, Unix	insecure

제 3 장 웹 교육시스템 설계

3.1 시스템의 개요 및 구성

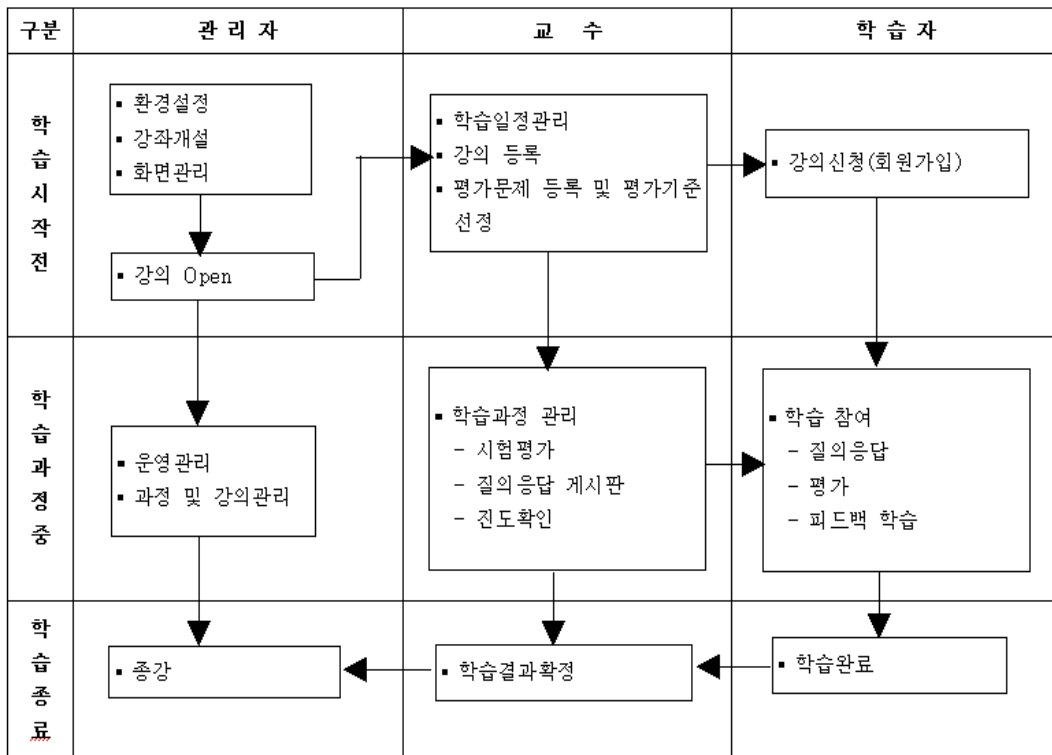
본 장에서는 테스트 기법이 적용될 대상이 되는 소규모의 웹 교육시스템을 설계하고 구현한다. 그 대상으로는 웹을 매체로 한 'C언어 교육'을 지원하는 시스템을 선정하였다. 대상 시스템은 기존의 단순한 HTML 위주의 교육에서 벗어나 학습자의 요구를 충분히 만족시켜주는 교수-학습자간의 상호작용에 의한 효율적인 학습형태가 이루어 질 수 있도록 객체지향적 방법론을 적용하여 시스템을 구성한다. 시스템의 설계 목표는 다음과 같다.

- 학습자는 각 장(chapter) 별로 강의를 수강하며, 해당 연습문제를 통해서 일정 점수를 획득해야만 다음 장을 수강할 수 있도록 하고, 각 페이지를 모두 수강해야만 문제풀이 단계로 넘어갈 수 있도록 한다.
- 각 장별 문제풀이를 통해서 학습자는 자신의 학습상태를 확인할 수 있다.
- 강의 리스트에 제시된 강의순서에 따라서 수업에 참여할 수 있고, 이전 강의를 수강하지 않았을 때에는 다음 강의를 수강할 수 없도록 학습절차에 제어를 둔다.
- 연습문제는 객관식(4지선다형)과 주관식(단답형) 문제의 혼합으로 제시되고, 학습자는 자신이 선택한 답과 정답을 비교해 볼 수 있도록 한다.
- 학습에 대한 평가는 서버에 의한 자동채점으로 이루어지며, 일정 점수를 획득하지 못했을 시에는 보충학습을 통해서 재 테스트를 실시할 수 있도록

록 설계한다.

- 교수자는 해당 강의를 웹을 통해서 등록할 수 있으며, 시험문제를 출제하고 학습자에게 피드백을 할 수 있도록 한다.

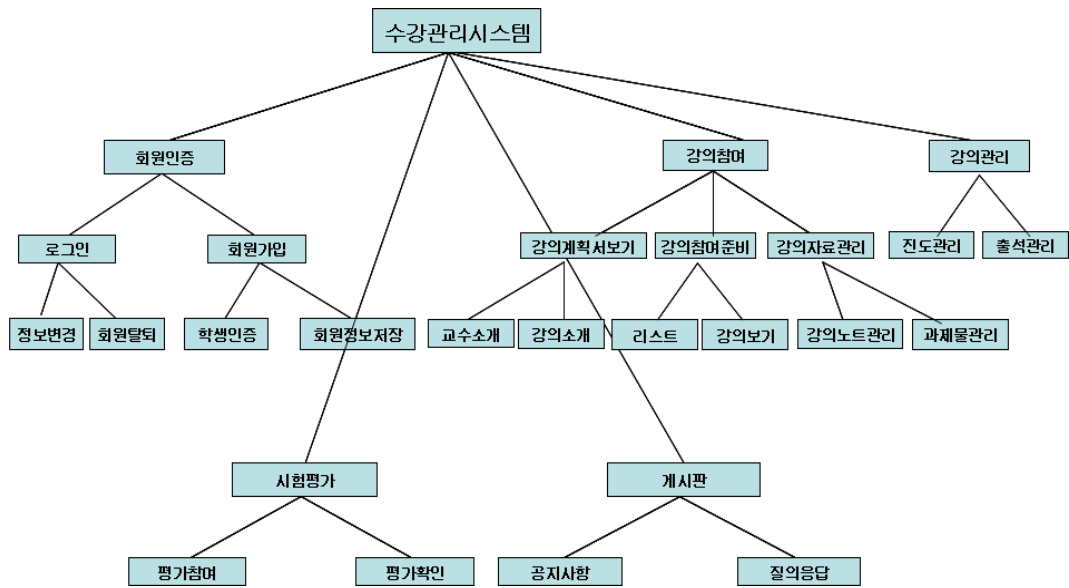
본 논문의 C 언어 교육의 학습지원을 위한 시스템의 프로세스 구조는 (그림 3-1)와 같다.



(그림 3-1) 학습지원 시스템의 프로세스

3.2 시스템 설계

본 시스템의 구조는 (그림 3-2)와 같이 크게 학습자 회원인증 모듈, 강의 참여 모듈, 강의관리 모듈, 시험평가 모듈, 게시판 모듈로 구성된다. 이 모듈 중에서 회원인증과 강의참여, 시험평가를 중점적으로 모듈별 기능에 대하여 소개한다.



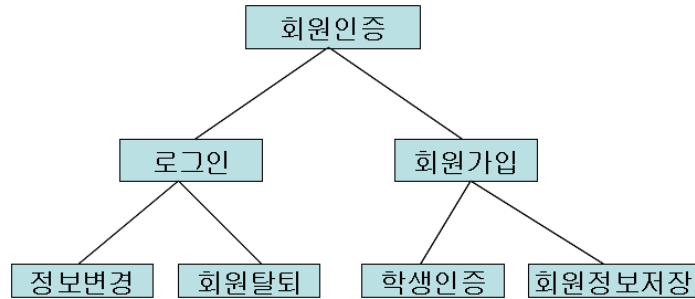
(그림3-2) 모듈 구조도

3.2.1 모듈설계

1) 회원인증 모듈

회원 인증은 크게 학습자, 교수자, 관리자로 구분되며, 회원인증 모듈은 다시 로그인처리에 관한 모듈과 회원가입처리와 가입된 회원의 정보를 관리하는 모듈로 구성된다. 이 모듈은 로그인에 대해 총괄적으로 관리하며, 회원가

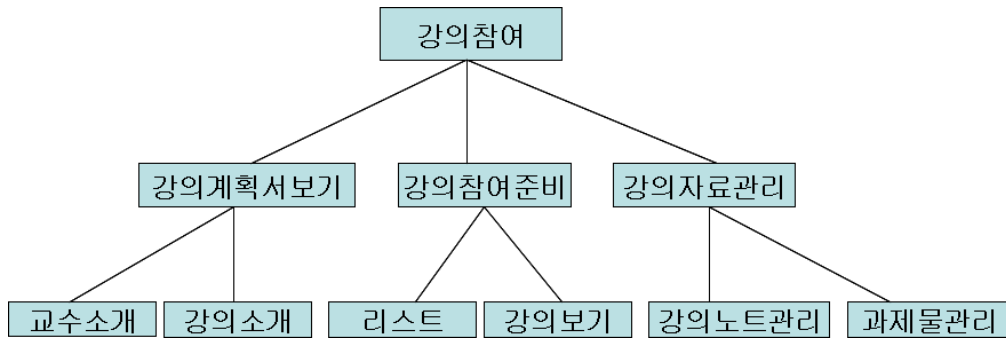
입에 있어서 크게 학습자와 교수자 구분하기 위한 속성을 가진다.



(그림 3-3) 회원인증 모듈 구조

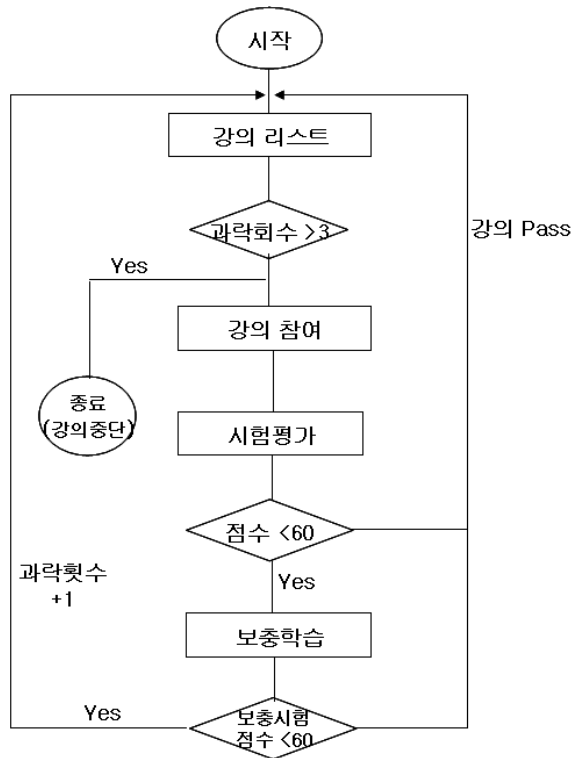
2) 강의참여 모듈

강의 수강에 관해 총괄적으로 관리하는 모듈로써, 특히 하위 모듈로 강의 참여준비 모듈과 강의관리 모듈이 핵심기능을 담당한다. 교수자는 로그인을 하면 강의관리와 회원관리, 게시판 관리를 할 수 있으며, 강의관리 메뉴를 통해서 해당 강의를 등록/수정/삭제 할 수 있다. 또한 시험문제를(일반강의 시험문제와 보충강의 시험문제) 등록하며, 해당 문제에 대한 정답입력과 주관식/객관식을 혼용하여 문제를 등록할 수 있다. 학습자는 본 시스템에 등록된 강의를 강의참여 모듈을 통해서 수강할 수 있으며, 강의진행에 있어서 제어흐름을 따라 단계별로 수강할 수 있다. 또한 강의관리 모듈을 통해서 수업진행 상태를 확인할 수 있다.



(그림 3-4) 강의참여 모듈 구조

(그림 3-5)은 학습자의 강의참여에 대한 흐름을 나타내고 있다.

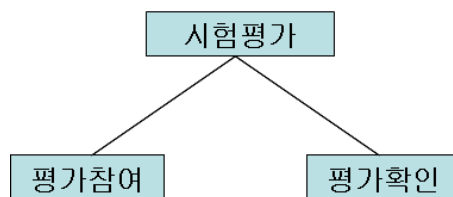


(그림 3-5) 학습자의 강의참여 흐름도

학습자는 사용자 인증을 통해서 해당 강의를 수강할 수 있으며, 로그인 후 등록된 강의리스트 중에서 해당 강의를 선택, 강의를 수강하기에 앞서 과락의 조건을 체크한 후 강의를 수강할 수 있다. 해당 강의를 수강한 학습자는 반드시 시험을 보아야하며, 시험점수가 60점 미만일 경우 보충수업을 통해 강의를 복습하고, 재시험을 보아야한다. 재시험결과가 60점 미만일 경우 과락의 횟수가 하나씩 증가하며, 총 과락의 수가 3번을 넘으면 본 시스템의 강의를 더 이상 수강할 수 없도록 한다.

3) 시험평가 모듈

학습자는 해당 강의를 수강 후 시험문제를 접하게 되며, 일정점수를 받아야만 다음 강의를 수강할 수 있는 자격이 주어진다. 등록된 문제 채점은 서버에 의해서 자동으로 채점된다. 학습자는 자신의 시험점수를 확인할 수 있고, 자신의 작성한 답과 정답을 비교, 응시한 시험에 대한 정보를 볼 수 있다.



(그림 3-6) 시험평가 모듈 구조

3.3 데이터베이스 설계

본 시스템에서 사용된 데이터베이스는 크게 강의관련(강좌 Master, 강좌 내

용, 사용자 강의기록)테이블과 회원테이블, 시험평가에 관한(강좌 테스트문제, 사용자가 풀어본 테스트문제저장, 사용자의 테스트 로그저장) 테이블로 구성된다.

3.3.1 강의관련 테이블

강의 관련 테이블은 강좌마스터, 강좌내용, 사용자 강의기록 테이블로 구성된다. 학습절차의 제어를 위해서 각 강의 완료시점, 수강중인 페이지번호 등 학습자의 강의참여에 대한 상세 강의기록과 교수이름, 강의ID와 강의명, 강좌시간, 강의시작일자와 종료일자에 대한 정보를 저장한다.

(표 3-1) 강좌 Master Table

열 이름	데이터 형식	길이	내용	Key
lectureID	Integer	10	강의ID	PK
lecturename	vchar	255	강의명	
teachername	vchar	255	교수님 이름	
lecturestart	vchar	8	시작일자	
lectureend	vchar	8	종료일자	
lecturetime	Integer	10	강좌시간	

(표 3-2) 강좌 내용 Table

열 이름	데이터 형식	길이	내용	Key
lectureID	Integer	10	강의ID	FK
lecturepage	Integer	10	강의Page	PK
lectureweek	vchar	20	주차	
lecturestart	vchar	8	시작일자	
lectureend	vchar	8	종료일자	
lecturecontent	text	-	내용	

(표 3-3) 사용자 강의 기록 Table

열 이름	데이터 형식	길이	내용	Key
memberID	varchar	30	사용자ID	PK
lectureID	Integer	10	강의ID	PK
ordertime	Integer	10	기준시간	
ordercount	Integer	10	기준횟수	
lecturepage	Integer	10	강의Page	
isEnd	char	1	완료체크	

3.3.2 회원 테이블

회원테이블은 사용자ID와 비밀번호를 포함 회원의 기본적인 정보와 사용자 등급에 관한 정보를 저장하기 위해 (표 3-4) 와 같이 생성한다.

(표 3-4) 회원테이블

열 이름	데이터 형식	길이	내용	Key
seq	Integer	10	시퀀스번호	
memberid	Varchar	30	사용자ID	PK
passwd	Varchar	8	비밀번호	
membername	Varchar	16	사용자이름	
phone	Varchar	16	전화번호	
email	Varchar	255	이메일	
memberlev	Int	1	사용자등급	
timestamp	datetime	-	저장시간	

3.3.3 시험평가 관련 테이블

시험평가에 관련 테이블은 강좌 테스트문제, 사용자가 풀어본 테스트문제 저장, 사용자의 테스트 로그저장 테이블로 구성된다. 학습강좌 테스트문제 테

이블은 학습자에게 시험문제를 제공하며, 교수자가 시험문제를 객관식/주관식 혼합으로 출제할 수 있도록 하고, 문제형식에서 일반강의문제는 'A'타입으로 지정하고, 보충수업문제는 'B'타입으로 지정한다. 사용자가 풀어본 테스트와 로그저장 테이블은 사용자ID와 강의ID, 정답여부, 획득점수, 시험통과, 보충여부 등 시험에 관련된 정보를 저장하고 최종 생성한다.

(표 3-5) 강좌 테스트 문제 테이블

열 이름	데이터 형식	길이	내용	Key
qnum	Integer	10	문제번호	PK
lectureID	Integer	10	강의ID	
questionID	char	1	문제형식	
question	text	-	문제	
testweek	char	2	문제주차	
answer1	varchar	255	보기1	
answer2	varchar	255	보기2	
answer3	varchar	255	보기3	
answer4	varchar	255	보기4	
rightanswer	varchar	255	정답	

(표 3-6) 사용자가 풀어본 테스트 문제 저장 테이블

열 이름	데이터 형식	길이	내용	Key
seq	Integer	10	저장번호	PK
memberID	varchar	30	사용자ID	
qnum	varchar	10	문제번호	
useranswer	varchar	255	사용자입력	
correct	char	1	정답여부	
isRe	char	1	보충여부	
lectureID	Integer	10	강의ID	
timestamp	datetime	-	저장시간	

(표 3-7) 사용자의 테스트 로그 저장 테이블

열이름	데이터형식	길이	내용	Key
lectureID	Integer	10	강의ID	PK
memberID	varchar	30	사용자ID	PK
lecturescore	Integer	10	획득점수	
timestamp	datetime	14	기록시간	
isPass	char	1	통과여부	

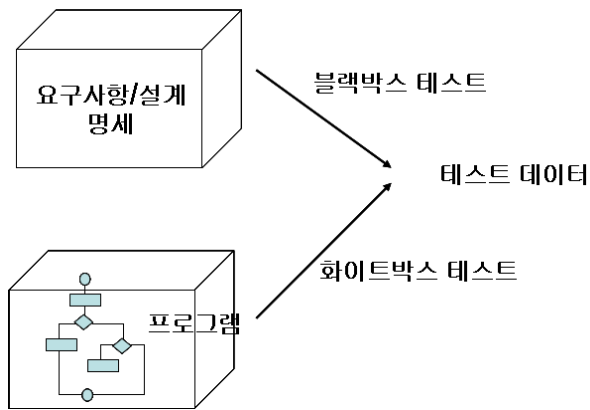
제 4 장 테스트 설계

4.1 테스트 방법 및 전략

소프트웨어 테스트 방법은 다양한 관점에서 분류할 수 있다. 우선 테스트 정보를 얻는 문서(분석 및 설계 명세, 프로그램 코드 등)의 종류에 따라 명세에 기반을 둔 테스트 방법과 구현 기반 테스트 방법으로 분류할 수 있다[20].

명세에 기반을 둔 테스트 방법은 프로그램 구현 내역을 참조하지 않고 명세에 나타난 정보만을 이용하여 테스트를 하는 방법이기 때문에 블랙박스(black-box) 테스트 방법이라고도 한다. 이러한 형태의 테스트 방법은 오류 수정과 같은 작업활동으로 프로그램의 구현 내역이 바뀌었다 할지라도 명세에 나타난 내용이 변하지 않는 한 이미 개발한 테스트 정보(예를 들면, 테스트 데이터를 만들 수 있는 정보)를 수정 없이 이용할 수 있다는 이점을 지니고 있다.

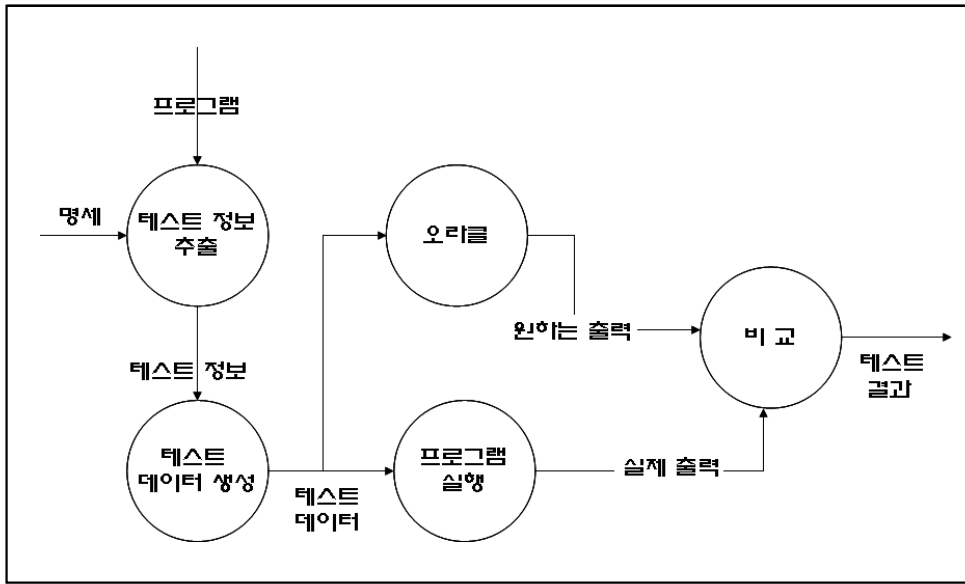
이에 반해 구현에 기반을 둔 테스트 방법은 프로그램의 논리 구조 및 구현 내역으로부터 테스트 정보를 추출하는 방법이다. 즉 프로그램 코드에서 제어 흐름 및 자료 흐름 정보를 분석하여 코드의 여러 특성을 검사할 수 있는 테스트 데이터를 생성한다. 이러한 방법은 프로그램의 구현 내역에 대한 지식이 필요하기 때문에 화이트박스(white-box) 테스트 방법이라고 한다.



(그림 4-1) 블랙박스 테스트와 화이트박스 테스트

또한, 프로그램의 오류를 찾아내기 위한 목적으로 프로그램 실행을 요구하는지에 따라 동적 테스트(dynamic test) 방법과 정적 테스트(static test) 방법으로 분류할 수 있다. 동적 테스트 방법은 프로그램의 실행을 요구하며, 정적 테스트 방법은 프로그램의 실행을 요구하지 않는다.

앞서 언급한 블랙박스 테스트 방법과 화이트박스 테스트 방법은 동적 테스트 방법으로 분류할 수 있다. 동적 테스트 방법이란 실제 프로그램을 테스트 데이터에 의해 실행하는 방식이다. (그림 4-2) 은 동적 테스트 과정을 보인 그림으로 동적테스트 방법은 프로그램 명세나 구현 논리 구조를 바탕으로 테스트 데이터를 선정 및 생성하여 프로그램의 입력으로 실행한다. 프로그램의 실제 실행 결과가 올바른 결과인지는 오라클을 사용하여 판단한 후에 테스트 결과를 생성한다[20]. 여기서 말하는 오라클이란 테스트 오라클(test oracle)을 말하며, 프로그램의 실제 실행결과가 올바른 결과인지를 판단하는 메커니즘이다. 프로그램을 테스트할 때 테스트 오라클 역할은 보통 사람이 담당한다.



(그림 4-2) 동적 테스트 프로세스

따라서, 웹 어플리케이션을 테스트하는 방법은 웹페이지가 정적이냐 동적이냐에 따라서 달라질 수 있다. 정적인 경우는 구문 테스트와 기능 혹은 서비스 테스트 방법으로 가능하지만, 동적으로 서버측에서 생성할 경우에는 모듈테스트, 통합 테스트, 시스템 테스트가 필요하다[18].

객체지향 시스템의 특성은 추상화, 상속성, 다형성과 서로 다른 클래스 혹은 같은 클래스내의 멤버 함수들간의 다양한 형태의 상호작용에 의해 기능이 수행된다는 점이다[22]. 이러한 특성들이 테스트에서도 의미를 갖는다.

본 논문에서는 크게 블랙박스 테스트와 화이트 박스 테스트 방법을 사용하며, 시스템 요구사항의 명세에 맞는지를 파악하는 시험으로 시스템의 내부 구조를 블랙박스로 보고 입력과 출력을 확인함으로써 시스템의 오류를 발견하는 방식의 블랙박스 테스트 방법을 사용하여 웹 서비스를 테스트 한다. 그리고

순차도를 기반으로 한 화이트박스 테스트를 실시한다.

4.2 테스트 케이스의 설계

4.2.1 유스케이스를 이용한 기능테스트

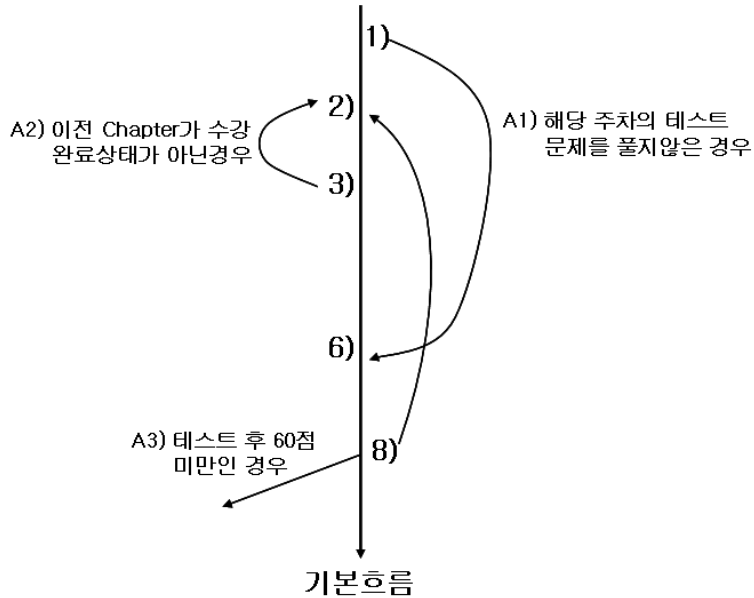
시스템 테스트는 시스템에 대한 내부적인 설계 정보를 이용하지 않고 시스템에 대한 입력/출력 정보에 바탕을 두고 수행한다. 즉, 시스템에 대한 요구 명세서를 바탕으로 테스트 케이스를 선정하는데, 기존의 소프트웨어의 시스템 테스트가 요구사항 명세서를 바탕으로 수행되는 것처럼 객체지향 소프트웨어에서 요구사항 명세서 역할을 하는 유스케이스 명세서를 바탕으로 시스템을 테스트한다. 즉, 각 유스케이스의 기능에 대한 구체적인 정의를 담은 유스케이스 명세서에 기술된 이벤트 흐름 등의 정보를 이용하여 테스트 케이스를 설계한다. 객체지향 방법론의 유스케이스 명세서는 시스템이 제공하는 단위 기능 즉 유스케이스에 대한 액터와 시스템 간의 상호 작용을 정의하고 있기 때문에 시스템을 테스트할 때 시스템에 주어질 입력 값과 출력 값을 결정하는 기준으로 사용될 수 있는 것이다.

(표 4-1)은 강의참여에 대한 유스케이스 기술서이며, 여기서 기술된 이벤트 흐름 정보를 바탕으로 시나리오별로 테스트 케이스를 설계한다. 강의참여 유스케이스 ID는 UC_LECTURE 이며, 크게 강의참여와 보충학습의 두 가지 기능으로 시나리오를 작성하였으나, 여기서는 강의참여에 대하여 시나리오를 기술하고 테스트케이스 설계를 보이고자 한다.

(표 4-1) 강의참여 유스케이스

유스케이스 ID	UC_LECTURE	유스케이스 명	강의참여
기능명	강의참여		
설명	회원으로서 강의를 들을 수 있다.		
관련액터	학습자		
사전조건	1. 로그인이 되어있어야 한다. 2. 이전 강의가 완료되었는지를 체크한다. 3. 강의를 수강할 수 없는 과락의 횟수가 3회 이상인 경우 본 강의를 수강할 수 없다. 과락의 조건 -> 각 chapter 수강 후 시험결과가 60점 미만으로 보충수업을 수강한 후 재시험에 응시했으나 또다시 60점 미만을 받은 경우		
사후조건	1. 강의를 듣고 나면 강의진도상태가 갱신되어야 한다. 2. 특정 Chapter에 포함된 모든 페이지를 수강해야한다. 3. 최종 페이지에서 Chapter를 마치고 나면, 쪽지시험(웹상)을 통해서 일정 점수(60점이상)를 받아야만 해당 Chapter를 수강 완료한 것으로 한다. 만약 특정 Chapter의 모든 페이지를 Open 하지 않은 상태에서 로그아웃하면 강의진도는 완료상태가 되어서는 안된다.		
기본흐름	1. 액터가 강의참여 메뉴를 클릭한다. (A-1) 2. 시스템은 강의목차를 표시한다. 3. 액터가 강의목차(강의리스트)중 수강할 강의를 선택한다. (A-2) 4. 시스템은 액터가 선택한 강의를 화면에 표시한다. 5. 액터가 모든 페이지를 수강 후 마지막 페이지에서 “다음” 버튼을 클릭한다. 6. 시스템은 해당 Chapter의 수강여부를 체크한 후 테스트 페이지를 표시한다. 7. 액터는 시험이 완료되면 테스트 확인 버튼을 클릭한다. 8. 시스템은 자동으로 시험지를 채점하고 60점 이상이면, 최종 수강이 완료임으로, 강의목차를 화면에 표시한다. (A-3)		
대안흐름	A-1. 그 주차의 수업을 성공적으로 다 듣고 시험을 안보고 그냥 빠져나왔을 시에는 강의참여를 선택, 강의 목차를 선택하면 바로 테스트 페이지를 보여준다. A-2. 만약 이전 Chapter가 완료되지 않았을 시(해당 Chapter의 모든 수강을 완료하지 못했을 때) -> “이전 강의를 완료해야만 강의를 보실 수 있습니다.” 라는 메시지를 표시하고, 확인버튼을 클릭하면 강의목차를 화면에 보여준다. A-3. 자동채점 후 60점 미만인 경우는 별도의 보충학습(그 주차의 학습내용 복습)을 통해서 다시 테스트 후 통과하면 해당 Chapter가 최종 완료된 것으로 함.		

강의참여 유스케이스에서 학습자가 정상적으로 해당 Chapter의 강의 수강을 완료하는 경우(기본흐름)와 그렇지 못한 경우(대안흐름)가 존재한다. (그림 4-3)은 강의참여 유스케이스의 각 이벤트 흐름을 도식화한 것을 보여준다.



(그림 4-3) 강의참여 유스케이스의 이벤트 흐름

(표 4-2)는 강의참여 유스케이스에서 선정한 테스트 시나리오와 해당 이벤트 흐름과의 관계를 보여준다.

(표 4-2) 강의참여 유스케이스의 테스트 시나리오

시나리오	수행경로	
S1 - 정상적인 수강완료	기본흐름	
S2 - 해당 주차의 수업을 성공적으로 듣고, 테스트 문제를 풀지 않았을 경우	기본흐름	A1) 대안흐름
S3 - 이전 Chapter가 수강완료 상태가 아닌 경우	기본흐름	A2) 대안흐름
S4 - 테스트 후 60점 미만인 경우	기본흐름	A3) 대안흐름

테스트 시나리오를 결정한 후 각 테스트 시나리오 별로 프로그램이 해당 경로를 수행할 수 있도록 입력 값들을 결정한다. (표 4-3)은 강의참여 유스케이스의 각 테스트 시나리오 별로 주어져야 하는 입력과 예상결과를 기술한 것이다.

(표 4-3) 강의참여 유스케이스의 테스트 케이스

테스트 케이스 ID	시나리오	입력				예상결과	실행결과	특수조건
		“강의참여” 메뉴	강의목차 (강의리스트)	강의 마지막 페이지의 “다음” 버튼	test 완료후 “확인”버튼			
TC 1	S1	클릭	클릭	클릭	클릭	정상적인 수강완료이며, 강의 목차를 화면에 표시		테스트 점수가 60점 이상
TC 2	S2	클릭	클릭	클릭	없음	해당 테스트 페이지 표시		
TC 3	S3	클릭	클릭	없음	없음	수강완료가 되지 않았다는 메시지 출력, 강의목차 표시		
TC 4	S4	클릭	클릭	클릭	클릭	보충학습을 실시		테스트 점수가 60점 미만

4.2.2 순차도를 이용한 화이트박스 테스트

화이트박스 테스트 방법은 프로그램을 분석하여 테스트에 필요한 정보를 추출하여 테스트 데이터를 생성한다. 화이트박스 테스트 방법을 이용하여 프로그램을 테스트하는 절차는 보통 다음과 같은 절차들로 구성된다[20].

① 테스트 데이터 적합성 기분(test data adequacy criterion) 선정

현실적으로 프로그램의 모든 입력 데이터를 사용하여 프로그램을 테스트할 수 없기 때문에 테스트 데이터를 선정 또는 생성하는 기준을 정하여 이 기준을 만족하는 입력 데이터만을 테스트 데이터로 사용한다.

② 데이터를 생성(test data generation)

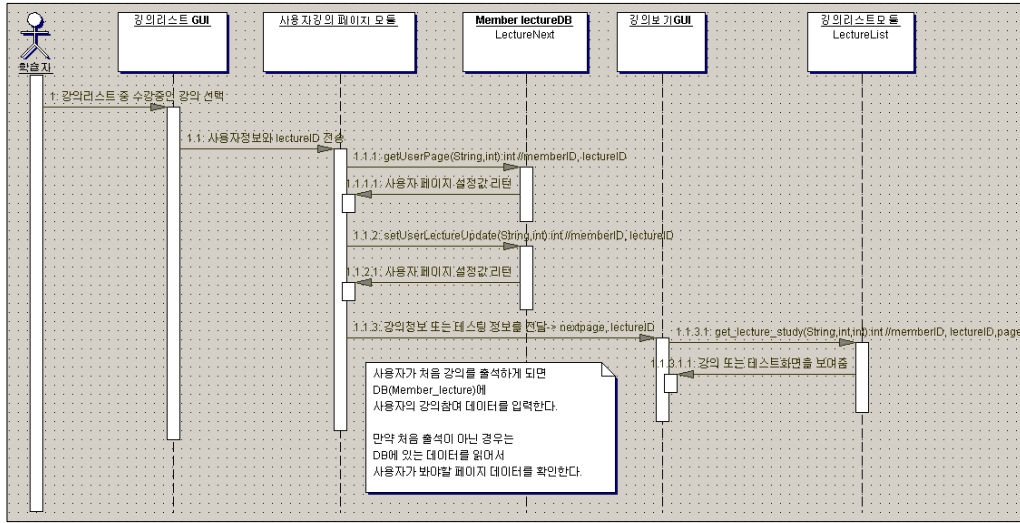
데이터를 생성하는 단계가 프로그램 테스트 단계 중에서 가장 노력이 많이 들고 자동화하기 어렵다. 선정된 기준을 만족하는 입력데이터를 테스트 데이터로 생성한다. 수작업 또는 자동화를 통하여 적합성 기준을 만족하는 테스트 데이터를 생성할 수 있으나 현실적으로 자동으로 테스트 데이터를 생성하는 작업은 매우 어렵기 때문에 보통 수작업을 통하여 테스트 데이터를 선정한다. 테스트 데이터를 생성하기 위해 명세나 프로그램 코드를 분석하여 생성하는 방법들이 있다.

③ 테스트 데이터 실행(test data execution)

생성된 테스트 데이터를 프로그램의 입력으로 실행하여 실행 결과를 분석한다. 만약 오류가 발견되었다면 오류의 원인을 찾아 제거한다.

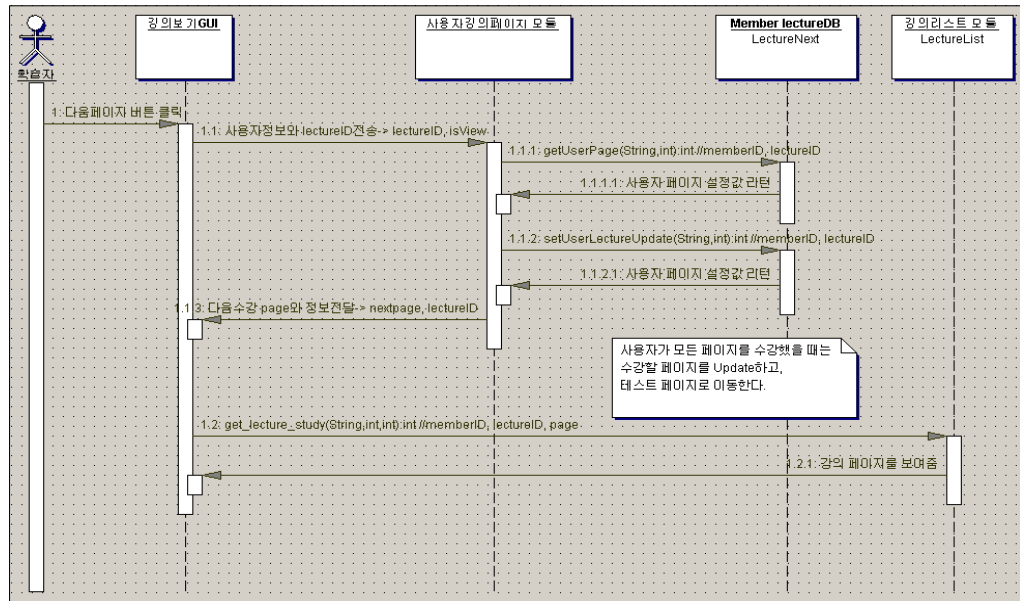
화이트박스 테스트 기법은 실제적인 프로그램 제어의 흐름에 착안하고 있고, 모듈 내부의 모든 구성요소를 집중적으로 테스트 할 수 있으며, 어떤 문장, 어떤 분기조건, 어떤 경로가 테스트 되었는지 확인하고, 테스트 결과를 보고 프로그램을 수정하는데 용이하다[20]. 따라서 본 논문에서는 화이트박스 테스트를 위해 테스트 케이스를 설계하고, 테스트 데이터를 생성하기 위해 유스 케이스의 기능명세를 바탕으로 프로그램 코드를 분석, 이를 바탕으로 순차도(Sequence Diagram)를 그린다. UML 순차도는 객체들끼리 주고받는 메시지의 순서를 시간의 흐름에 따라 보여주므로, 시간에 따른 제어 흐름이 가시적으로 명확히 눈에 들어온다.

테스트 케이스 ID : TC 5



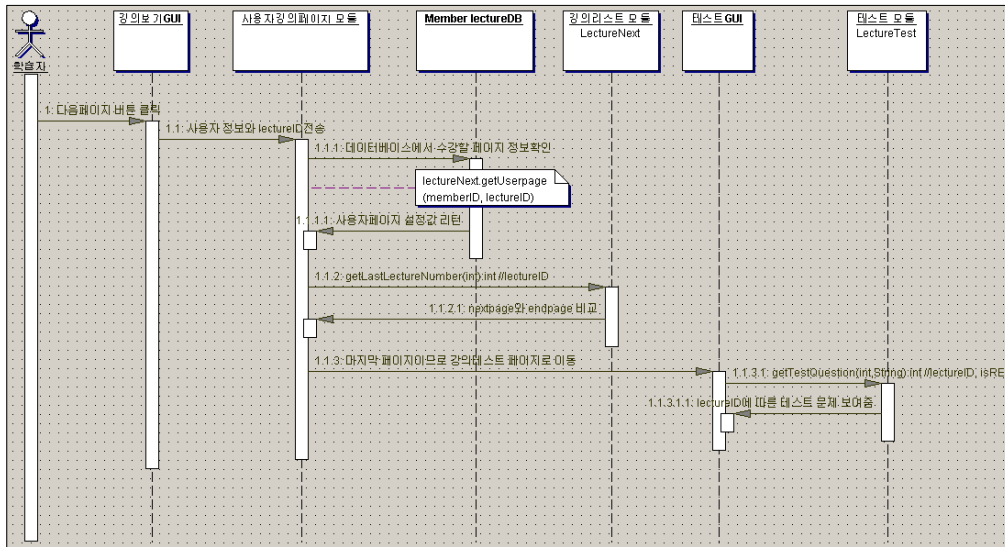
(그림 4-4) 강의참여에 대한 시퀀스 다이어그램

테스트 케이스 ID : TC 6



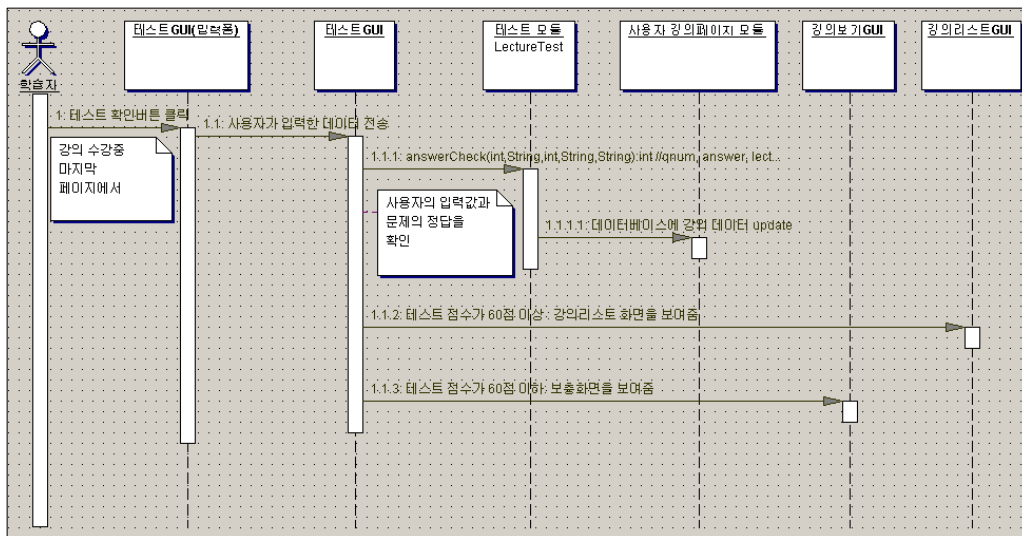
(그림 4-5) 학습자 수강 진행상태 시퀀스 다이어그램

테스트 케이스 ID : TC 7



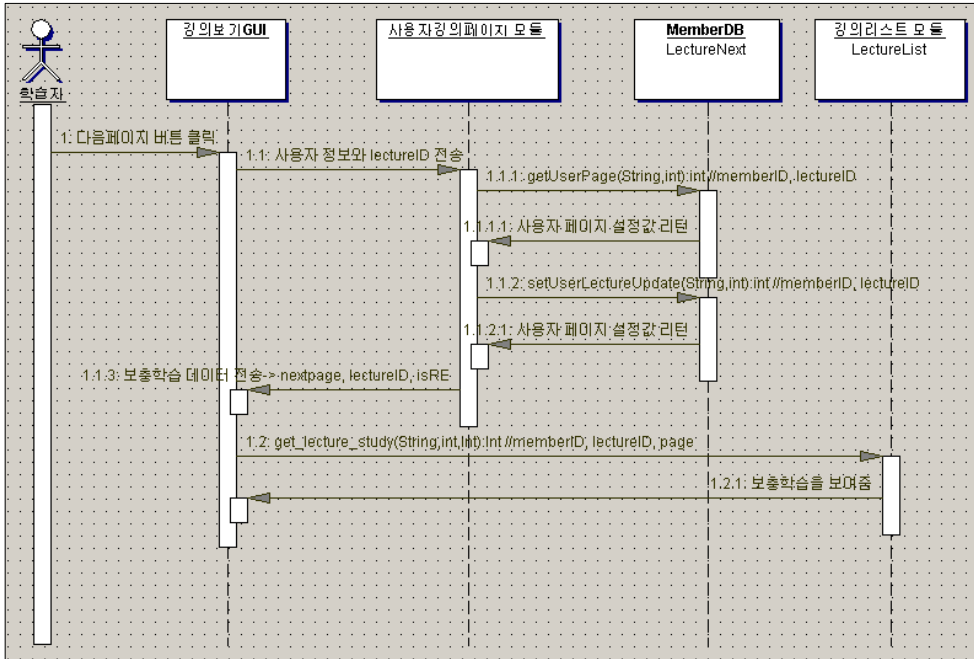
(그림 4-6) 학습자 수강 마지막 페이지에 대한 시퀀스 다이어그램

테스트 케이스 ID : TC 8



(그림 4-7) 학습자 테스트 시퀀스 다이어그램

테스트 케이스 ID : TC 9



(그림 4-8) 보충학습 시퀀스 다이어그램

각 시퀀스 다이어그램에서 볼 수 있듯이 교류에 참가하는 객체들은 다이어그램의 위쪽 부분에 가로축에 배치하고, 객체들이 주고받는 메시지들은 세로축에 따라 배치시켜서 위에서 아래로 시간흐름을 맞춘다. 위에서 작성한 시퀀스 다이어그램을 바탕으로 한 화이트박스 테스트는 웹서버에 로그파일을 두어서 사용자에게 대한 이벤트 정보를 확인할 수 있게 한다. 이 로그파일을 분석함으로써 테스트를 검증할 수 있다.

4.2.3 폼 입력을 통한 블랙박스 테스트

블랙박스 테스트는 프로그램의 내부 구조는 고려하지 않고 프로그램에 입력

을 하고 그에 대한 출력으로 얻어지는 결과를 검사하는 테스트 방법이다[21]. 즉, 테스트 대상 프로그램의 논리를 생각하지 않고, 기능이나 인터페이스에 관한 명세서를 기반으로 테스트 데이터를 선정해서 테스트를 실시한다. 본 논문에서는 기능적인 요구 사항에 집중하는 테스트 방법으로 화이트박스 테스트를 보완하는 테스트로서 블랙박스 테스트를 실시한다. 따라서 사용자의 입력을 통해서 실제적인 결과를 얻을 수 있는 로그인처리 기능에 대하여 테스트 설계를 보이겠다. (표 4-4)는 등록 유스케이스로 회원으로 학습자와 교수자로 나뉘며, 정상적인 아이디와 암호로 로그인해야만 시스템을 사용할 수 있다.

(표 4-4) 등록 유스케이스 기술서

유스케이스 ID	UC_LOGIN	유스케이스 명	로그인처리
기능명	등록		
설명	회원 ID와 암호를 사용하여 사용자를 인증한다.		
관련액터	회원(학습자, 교수자)		
사전조건	없음		
사후조건	로그인 성공 시 액터는 시스템사용 권한을 갖는다.		
기본흐름	1. 액터는 해당 ID 및 암호를 입력하고 로그인 버튼을 클릭한다. 2. 시스템은 ID와 암호를 조회 후, 일치하면 사용자를 인증하고, 해당 페이지를 표시한다.(A-1)		
대안흐름	A-1. 시스템은 ID 및 암호가 일치하지 않으면, 실패 메시지와 로그인 페이지를 표시한다.		

(표 4-5) 등록 유스케이스의 테스트 케이스

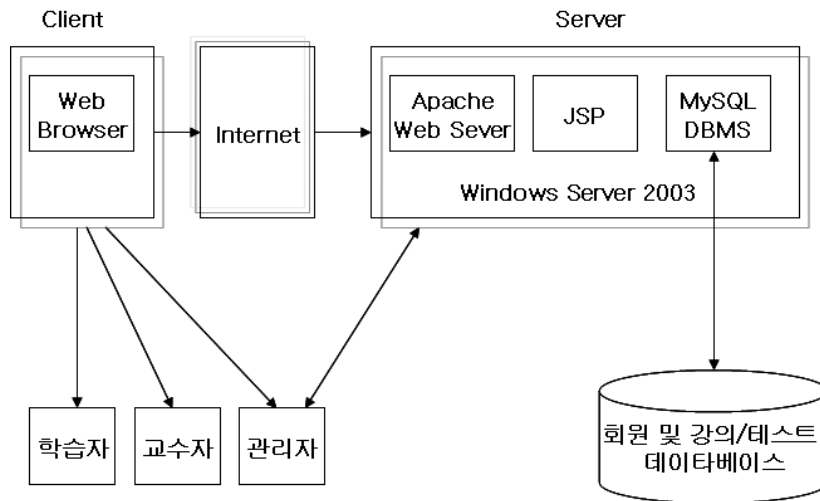
유스케이스 ID : UC_LOGIN				
목적 : 시스템의 화면 테스트				
특수 요구 : 사용자 ID와 패스워드의 검증				
단계	입력	예상출력	실행결과	특수조건
시스템 메인화면에서 로그인				
1. 정상적인 사용자 ID와 패스워드로 로그인을 시도한다.				
1	사용자 ID필드 클릭	사용자 ID 필드 반전		
2	정상 사용자 ID 입력	사용자 ID 화면에 보임		
3	패스워드 필드 클릭	패스워드 필드 반전		
4	정상 패스워드 입력	화면에 보이지 않음		
5	“로그인” 버튼을 클릭	로그인된 화면		
2. 사용자 ID와 패스워드 없이 로그인을 시도한다.				
1	“로그인” 버튼을 클릭	로그인 실패창이 뜬		실패 메시지의 “확인”버튼을 클릭하면, 정상적인 입력 데이터 형식을 보임
2	“확인” 버튼 클릭	로그인 화면으로 리턴		
3. 정상 사용자 ID와 패스워드가 없이 로그인을 시도한다.				
1	사용자 ID필드 클릭	사용자 ID 필드 반전		
2	정상 사용자 ID 입력	사용자 ID 화면에 보임		
3	“로그인” 버튼을 클릭	로그인 실패창이 뜬		실패 메시지의 “확인”버튼을 클릭하면, 정상적인 입력 데이터 형식을 보임
4	“확인” 버튼 클릭	로그인 화면으로 리턴		
4. 정상 사용자 ID와 틀린 패스워드로 로그인을 시도한다.				
1	사용자 ID필드 클릭	사용자 ID 필드 반전		
2	정상 사용자 ID 입력	사용자 ID 화면에 보임		
3	패스워드 필드 클릭	패스워드 필드 반전		
4	틀린 패스워드 입력	화면에 보이지 않음		
5	“로그인” 버튼을 클릭	로그인 실패창이 뜬		실패 메시지의 “확인”버튼을 클릭하면, 정상적인 입력 데이터 형식을 보임
4	“확인” 버튼 클릭	로그인 화면으로 리턴		
5. 사용자 ID의 입력없이 정상적인 패스워드로만 로그인 시도한다.				
1	패스워드 필드 클릭	패스워드 필드 반전		
2	정상 패스워드 입력	화면에 보이지 않음		
3	“로그인” 버튼을 클릭	로그인 실패창이 뜬		실패 메시지의 “확인”버튼을 클릭하면, 정상적인 입력 데이터 형식을 보임
4	“확인” 버튼 클릭	로그인 화면으로 리턴		

제 5 장 시스템 구현 및 평가

5.1 시스템 구현

5.1.1 시스템 구성

본 시스템의 구조는 (그림 5-1)과 같이 웹 기반의 클라이언트-서버 시스템으로 구성된다. Windows Server 2003 운영체제를 기반으로 JSP와 MySQL 데이터베이스시스템, 아파치 웹 엔진으로 구성된다.



(그림 5-1) 시스템 구성

5.1.2 개발환경

본 연구는 다음과 같은 하드웨어와 소프트웨어의 환경에서 개발되었다.

(표 5-1) 하드웨어 환경

구 분	사 양
CPU	Pentium IV 1.8GHz
RAM	512MB
HDD	80G

(표 5-2) 소프트웨어 환경

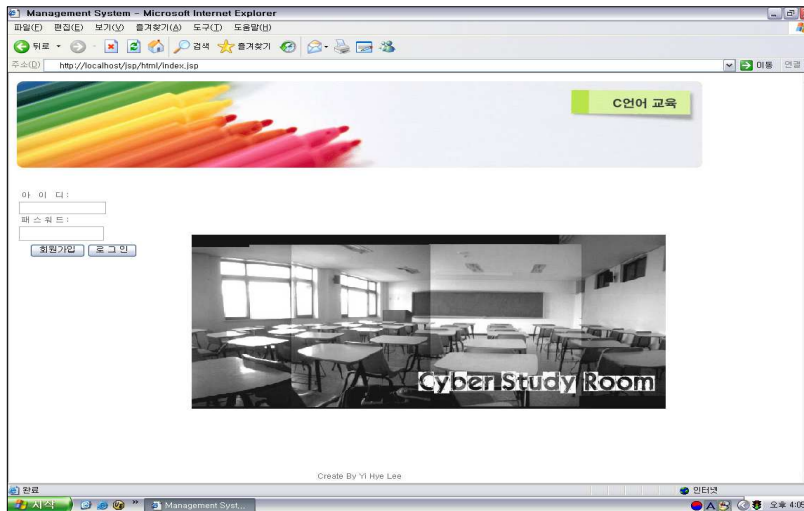
구 분	사 양
OS	Windows Server 2003
웹서버	Apache Web Server
DBMS	MySQL
개발언어	HTML, JSP, JavaScript
사용된 툴	Adobe Photoshop8.0 Together 6.0
웹브라우저	Internet Explorer 4.0 이상

5.1.3 시스템 구현

본 시스템의 구현에서는 JSP page 문서와 Module Package, 로그 기록, 기타 구현 관련 문제에 대해 살펴본다. PC에서의 모듈별 내용들은 검색 내용에 따라 JSP와 Module Package를 사용하여 데이터베이스에서 검색되며 검색된 데이터는 JSP로 원하는 결과의 출력을 클라이언트의 웹브라우저에 자동으로 제공된다.

아래의 (그림 5-2)는 시스템의 초기 화면으로 회원가입은 필수이며, 로그인

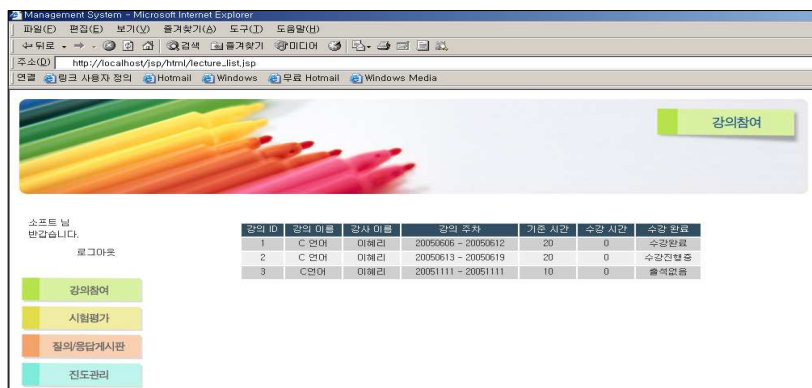
을 하지 않은 상태에서는 본 강의를 수강할 수 없도록 하였다.



(그림 5-2) 초기화면

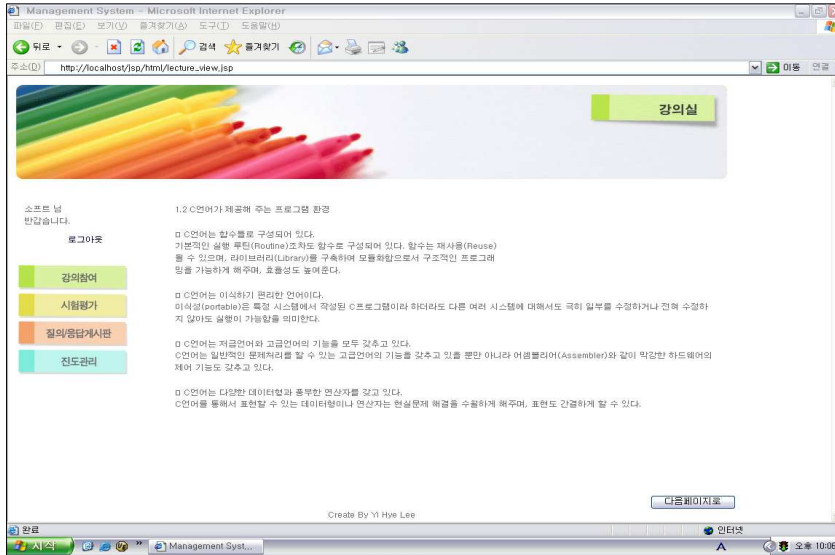
(1) 학습자의 주요화면

(그림 5-3)은 로그인한 사용자의 강좌 리스트를 데이터베이스에서 검색하여 해당 강의리스트의 정보를 학습자에게 보여주는 화면이다. 학습자는 강의리스트를 클릭하여 강의에 참여할 수 있다.



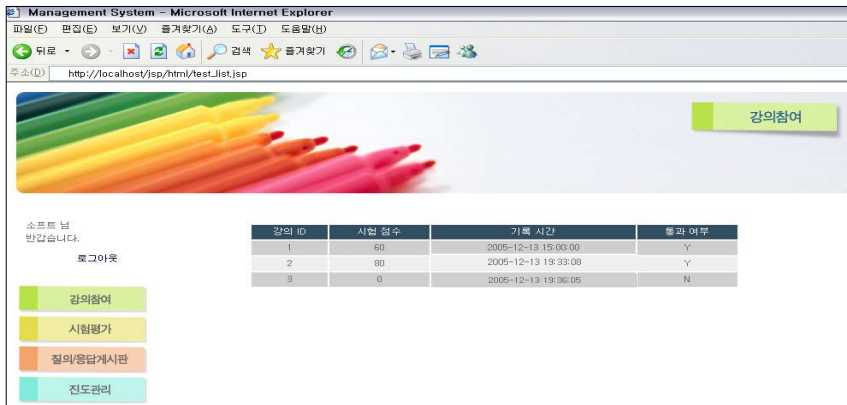
(그림 5-3) 강의리스트 출력화면

특히 강의를 참여하기전에 학습자의 현재 강의상태, 즉 일반강의인지 보충 강의인지를 확인하고, 사용자가 강의를 건너뛰었는지를 반드시 체크하게 된다. (그림 5-4)는 해당 학습자의 데이터와 일치하는 강의를 보여주는 화면이다.



(그림 5-4) 강의참여 화면

(그림 5-5)는 시험평가 메뉴를 클릭했을 때 학습자가 응시한 시험정보를 나타내고 있다.



(그림 5-5) 시험평가확인 리스트 화면



(그림 5-6) 시험평가 상세정보 화면

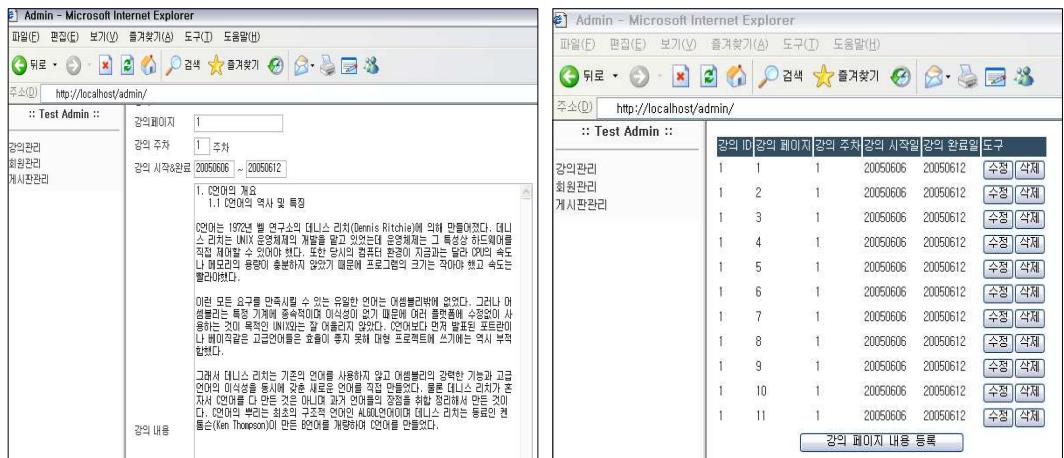
시험평가확인 리스트 중에서 확인하고자하는 시험리스트를 클릭하면 시험에 응시한 주차정보와 시험문제번호 및 시험내용, 정답여부를 확인할 수 있다.

(2) 교수자의 주요화면

(그림 5-7)과 같이 로그인 후 교수자는 강의관리/회원관리/게시판관리 메뉴를 통해서 강의를 등록, 회원정보 수정 및 게시판 등록, 삭제를 할 수 있다.



(그림 5-7) 강의관리 화면

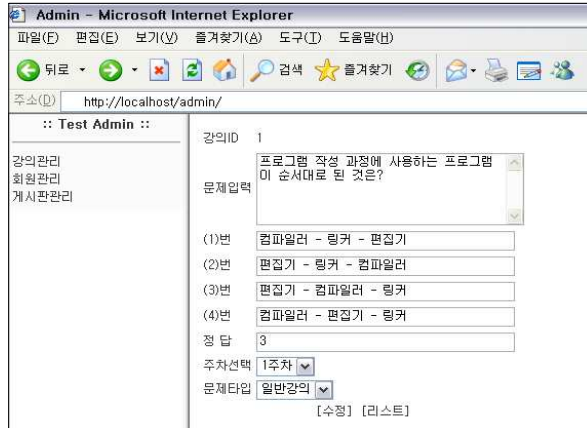


(그림 5-8) 강의등록 화면

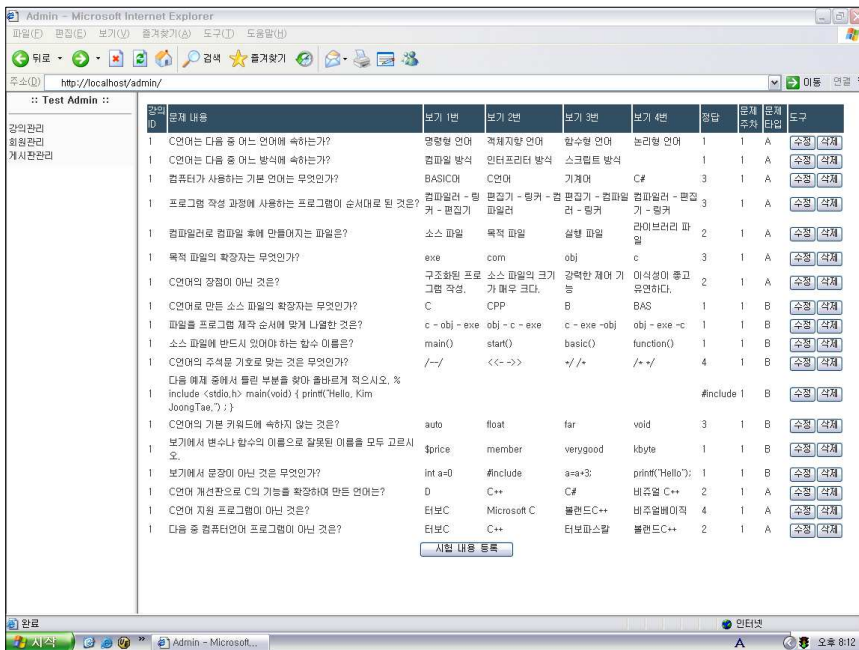
교수자는 강의관리메뉴를 통해서 새로운 강의를 등록할 수 있고, 등록된 강의를 수정 및 삭제할 수 있다. 또한 페이지별로 상세 강의내용을 수정하거나 삭제할 수도 있다. (그림 5-8)은 이러한 내용을 나타내고 있는 화면이다.

(그림 5-9)은 시험문제를 등록하는 화면이다. 각 주차를 선택할 수 있고, 문제타입별로 일반강의 또는 보충강의를 선택하고, 객관식 문제일 경우에는 보

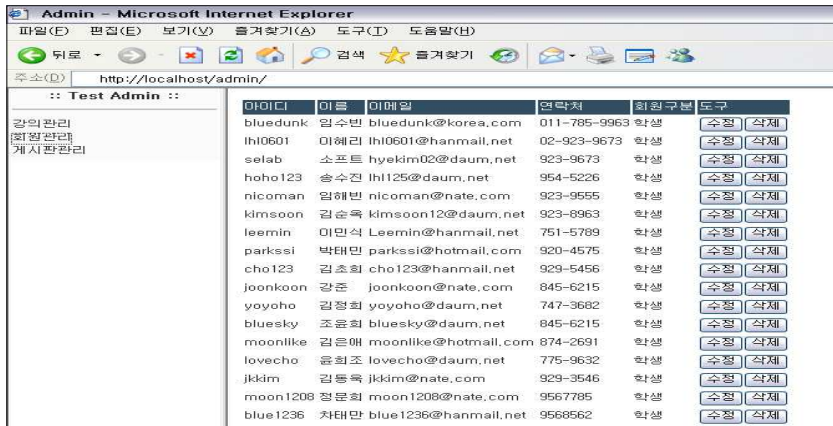
기를 선택하여 작성하면 되고, 주관식일 경우에는 단답형의 답안을 작성하면 된다.



(그림 5-9) 시험문제 등록 화면

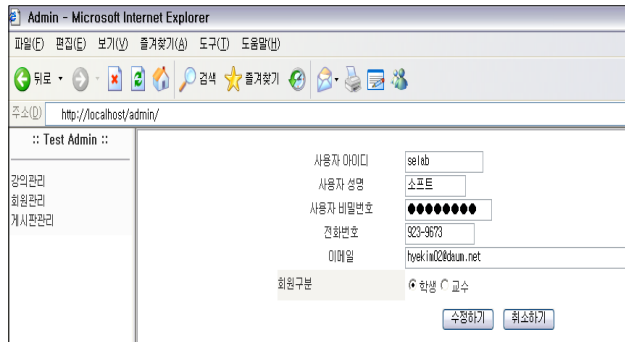


(그림 5-10) 시험문제 상세보기 화면



(그림 5-11) 학습자 관리 화면

(그림 5-12)는 등록된 전체 학습자에 대한 상세정보를 나타내고 있는 화면이다. 교수자는 각 학습자의 정보를 수정하거나 삭제할 수 있다.



(그림 5-12) 학습자 정보 수정 화면

5.2 테스트 케이스에 기반한 시스템 테스트

5.2.1 유스케이스를 이용한 기능테스트

본 시스템을 평가하기 위해서 4장에서 테스트 시나리오별 설계한 테스트

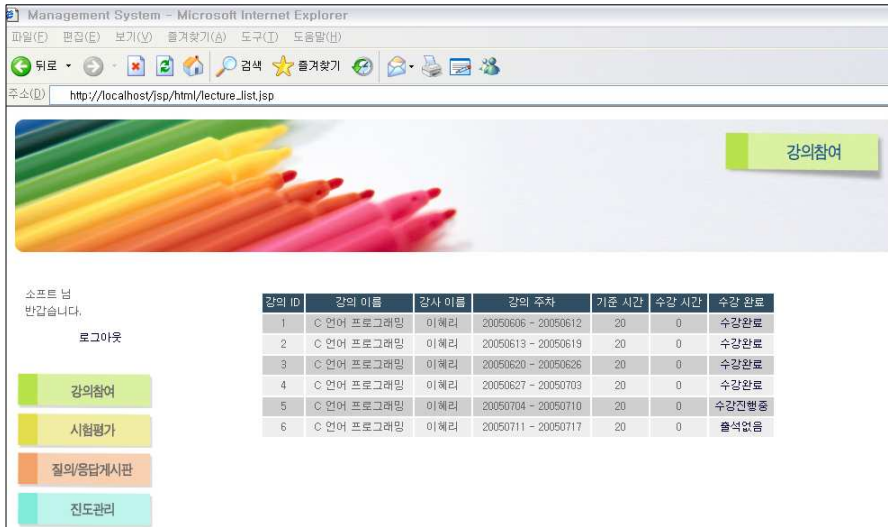
케이스를 사용하여 시스템을 테스트 한다. (표 5-3)은 강의참여 유스케이스의 각 시나리오 별로 주어져야 하는 입력과 예상결과, 그리고 테스트를 통한 실행결과를 보여주고 있다.

(표 5-3) 강의참여 유스케이스의 테스트 케이스

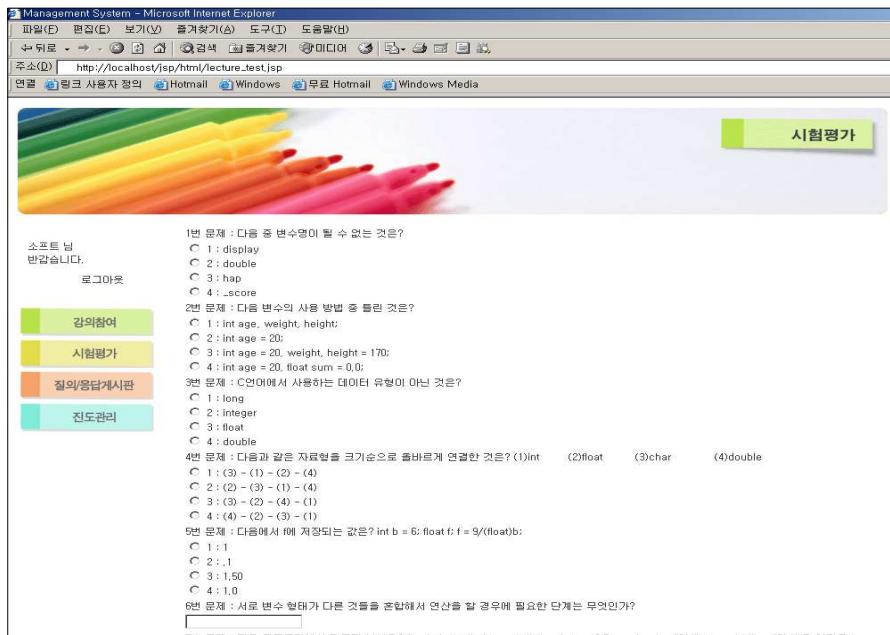
테스트 케이스 ID	시나리오	입력				예상결과	실행결과	특수조건
		"강의참여" 메뉴	강의목차 (강의리스트)	강의 마지막 페이지의 "다음" 버튼	test 완료후 "확인"버튼			
TC 1	S1	클릭	클릭	클릭	클릭	정상적인 수강완료이며, 강의 목차를 화면에 표시	합격	테스트 점수가 60점 이상
TC 2	S2	클릭	클릭	클릭	없음	해당 테스트 페이지 표시	합격	
TC 3	S3	클릭	클릭	없음	없음	수강완료가 되지 않았다는 메시지 출력, 강의목차 표시	합격	
TC 4	S4	클릭	클릭	클릭	클릭	보충학습을 실시	합격	테스트 점수가 60점 미만

다음 테스트 실행결과의 화면들을 보이고 있다.

(그림 5-13)은 정상적으로 일반강의를 참여한 후 시험평가를 거쳐서 다음 강의를 수강할 수 있도록 강의 리스트를 보여주는 화면이다.



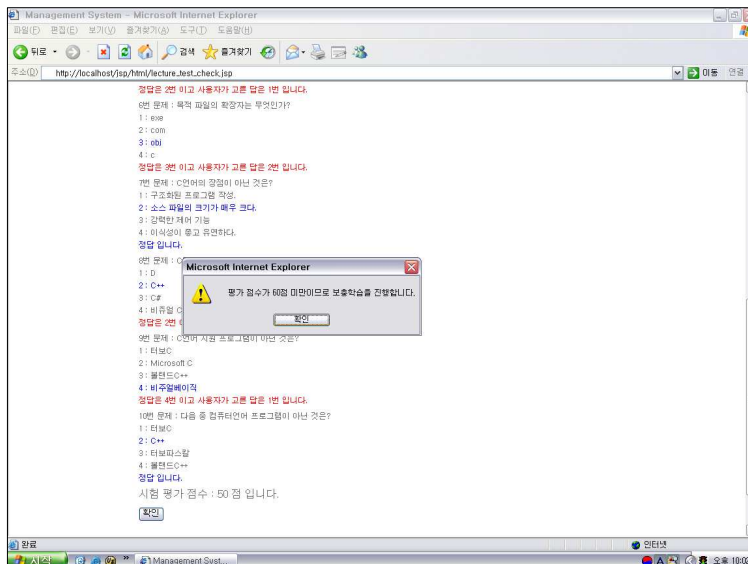
(그림 5-13) 수강완료 테스트 화면



(그림 5-14) 테스트 페이지 화면



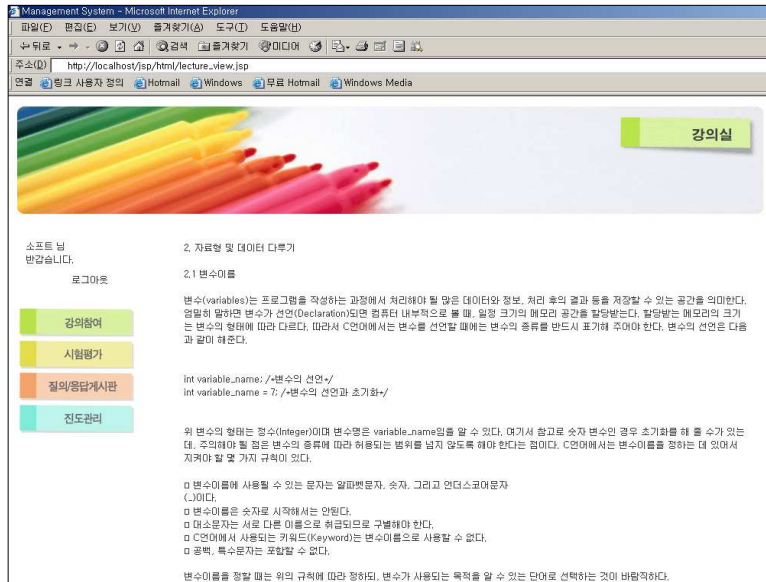
(그림 5-15) 정답확인 화면



(그림 5-16) 보충학습 메시지 화면

(그림 5-14)의 테스트 페이지 화면에서 학습자는 시험을 보고나서 확인버튼을 선택하면 자신이 작성한 답과 정답의 확인화면이 출력된다. 정답일 경

우 파랑색으로 색인, 틀린 경우는 해당 문제의 아래쪽에 빨강색으로 정답이 프린트된다. 정답 확인 후 60점 미만의 점수를 받은 학습자는 (그림 5-16)과 같이 보충학습을 해야 한다는 메시지와 함께 보충학습단계로 전환된다. 학습자의 보충학습 화면은 (그림 5-17)과 같다.



(그림 5-17) 보충학습 화면

5.2.2 순차도를 이용한 화이트박스 테스트

(표 5-4) Test Case/Log

테스트 케이스ID	테스트 케이스	예상결과	수행결과	PASS/FAIL	비고
TC 5	학습자의 정상적인 학습참여	강의 또는 테스트화면 출력	정상적인 강의화면/테스트화면출력	PASS	
TC 6	강의 진행 상태 중 다음페이지로 전환	다음페이지 출력	다음페이지 출력	PASS	
TC 7	수강 마지막페이지에 대한 테스트	테스트 페이지로 출력	해당 학습자에 따른 테스트 문제를 보여줌	PASS	
TC 8	테스트 문제풀이	강의리스트 또는 보충화면을 보여줌	강의 리스트 또는 보충화면 출력	PASS	
TC 9	보충학습	보충학습 화면출력	보충학습 화면출력	PASS	

(표 5-4)는 4장에서 작성한 시퀀스 다이어그램을 사용하여 화이트박스 테스트 기법을 적용한 모듈 테스트이다. 웹서버에 로그파일을 두어서 각 객체들 사이의 주고받는 메시지에 대한 이벤트 정보를 확인할 수 있게 하였다. 이 로그 파일을 분석함으로써 테스트를 검증할 수 있다.

```

1 Sun Nov 27 18:39:00 KST 2005 >>> LectureNext.getLastLectureNumber 함수 호출.
2 쿼리문 : select isPass from lecture_test_log where lectureID=1 and memberID='selab'
3 Sun Nov 27 18:39:00 KST 2005 >>> LectureNext.getUserLecturePage 함수 호출.
4 쿼리문 : select lecturepage from member_lecture where memberID='selab' and lectureID=1
5 Sun Nov 27 18:39:00 KST 2005 >>> LectureNext.getUserLecturePage 함수 호출. 사용자 데이터 없음. 초기 데이터 입력.
6 쿼리문 : insert into member_lecture (memberID,lectureID,ordertime,ordercount,lecturepage) values('selab',1,0,1,1)
7 Sun Nov 27 18:39:00 KST 2005 >>> LectureNext.getUserLecturePage 함수 호출. 초기데이터 입력후 nextPage값 : 1
8 Sun Nov 27 18:39:00 KST 2005 >>> LectureNext.getLastLectureNumber 함수 호출.
9 쿼리문 : select max(lecturepage) as lecturepage from lecture_study where lectureID=1
10 Sun Nov 27 18:39:00 KST 2005 >>> LectureNext.getUserLecturePage 함수 호출. 사용자 데이터 있음
11 Sun Nov 27 18:39:00 KST 2005 >>> LectureNext.getUserLecturePage 함수 완료후 lastpage 값 : 11
12 Sun Nov 27 18:39:03 KST 2005 >>> LectureNext.getLastLectureNumber 함수 호출.
13 쿼리문 : select isPass from lecture_test_log where lectureID=1 and memberID='selab'
14 Sun Nov 27 18:39:03 KST 2005 >>> LectureNext.getUserLecturePage 함수 호출.
15 쿼리문 : select lecturepage from member_lecture where memberID='selab' and lectureID=1
16 Sun Nov 27 18:39:03 KST 2005 >>> LectureNext.getUserLecturePage 함수 호출. 사용자 데이터 있음
17 Sun Nov 27 18:39:03 KST 2005 >>> LectureNext.setUserLectureUpdate 함수 호출. 사용자 데이터 업데이트
18 쿼리문 : update member_lecture set lecturepage=1 where memberID='selab' and lectureID=1
19 Sun Nov 27 18:39:03 KST 2005 >>> LectureNext.getUserLecturePage 함수 완료후 nextPage값 : 1
20 Sun Nov 27 18:39:03 KST 2005 >>> LectureNext.getLastLectureNumber 함수 호출.
21 쿼리문 : select max(lecturepage) as lecturepage from lecture_study where lectureID=1
22 Sun Nov 27 18:39:03 KST 2005 >>> LectureNext.getUserLecturePage 함수 호출. 사용자 데이터 있음
23 Sun Nov 27 18:39:03 KST 2005 >>> LectureNext.getUserLecturePage 함수 완료후 lastpage 값 : 11
24 Sun Nov 27 18:39:03 KST 2005 >>> LectureNext.setUserLectureUpdate 함수 호출. 사용자 데이터 업데이트
25 쿼리문 : update member_lecture set lecturepage=1 where memberID='selab' and lectureID=1
26 Sun Nov 27 18:39:13 KST 2005 >>> LectureNext.getLastLectureNumber 함수 호출.
27 쿼리문 : select isPass from lecture_test_log where lectureID=1 and memberID='selab'
28 Sun Nov 27 18:39:13 KST 2005 >>> LectureNext.getUserLecturePage 함수 호출.
29 쿼리문 : select lecturepage from member_lecture where memberID='selab' and lectureID=1
30 Sun Nov 27 18:39:13 KST 2005 >>> LectureNext.getUserLecturePage 함수 호출. 사용자 데이터 있음
31 Sun Nov 27 18:39:13 KST 2005 >>> LectureNext.setUserLectureUpdate 함수 호출. 사용자 데이터 업데이트
32 쿼리문 : update member_lecture set lecturepage=2 where memberID='selab' and lectureID=1
33 Sun Nov 27 18:39:13 KST 2005 >>> LectureNext.getUserLecturePage 함수 완료후 nextPage값 : 2
34 Sun Nov 27 18:39:13 KST 2005 >>> LectureNext.getLastLectureNumber 함수 호출.
35 쿼리문 : select max(lecturepage) as lecturepage from lecture_study where lectureID=1
36 Sun Nov 27 18:39:13 KST 2005 >>> LectureNext.getUserLecturePage 함수 호출. 사용자 데이터 있음
37 Sun Nov 27 18:39:13 KST 2005 >>> LectureNext.getUserLecturePage 함수 완료후 lastpage 값 : 11
38 Sun Nov 27 18:39:13 KST 2005 >>> LectureNext.setUserLectureUpdate 함수 호출. 사용자 데이터 업데이트
39 쿼리문 : update member_lecture set lecturepage=1 where memberID='selab' and lectureID=1
40 Sun Nov 27 18:39:13 KST 2005 >>> LectureNext.getLastLectureNumber 함수 호출.

174 Sun Nov 27 18:40:33 KST 2005 >>> LectureNext.getLastLectureNumber 함수 호출.
175 쿼리문 : select isPass from lecture_test_log where lectureID=2 and memberID='selab'
176 Sun Nov 27 18:40:33 KST 2005 >>> LectureNext.getUserLecturePage 함수 호출.
177 쿼리문 : select lecturepage from member_lecture where memberID='selab' and lectureID=2
178 Sun Nov 27 18:40:33 KST 2005 >>> LectureNext.getUserLecturePage 함수 호출. 사용자 데이터 없음. 초기 데이터 입력.
179 쿼리문 : insert into member_lecture (memberID,lectureID,ordertime,ordercount,lecturepage) values('selab',2,0,1,1)
180 Sun Nov 27 18:40:33 KST 2005 >>> LectureNext.getUserLecturePage 함수 호출. 초기데이터 입력후 nextPage값 : 1
181 Sun Nov 27 18:40:33 KST 2005 >>> LectureNext.getLastLectureNumber 함수 호출.
182 쿼리문 : select max(lecturepage) as lecturepage from lecture_study where lectureID=2
183 Sun Nov 27 18:40:33 KST 2005 >>> LectureNext.getUserLecturePage 함수 호출. 사용자 데이터 있음
184 Sun Nov 27 18:40:33 KST 2005 >>> LectureNext.getUserLecturePage 함수 완료후 lastpage 값 : 6

```

(그림 5-18) log.txt 파일에 기록된 사용자 정보

(그림 5-18)은 log.txt 파일에 기록된 사용자의 이벤트정보 흐름을 보여주고 있다. 학습자 아이디가 'selab' 인 사용자가 강의참여에 대한 로그파일 내용이 다. 1강 수강 후 2강의 수강 흐름의 이벤트들에 대한 함수호출 및 정보들을 보여주고 있다.

5.2.3 폼 입력을 통한 블랙박스 테스트

다음은 사용자의 폼 입력을 받아서 로그인 기능을 테스트하기 위한 시스템의 초기 화면이다.



(그림 5-19) 로그인 화면

(표 5-5) 등록 유스케이스의 테스트 케이스의 실행 결과표

유스케이스 ID : UC_LOGIN				
목적 : 시스템의 화면 테스트				
특수 요구 : 사용자 ID와 패스워드의 검증 (시스템 메인화면에서 로그인)				
단계	입력	예상출력	실행결과	특주조건
1. 정상적인 사용자 ID와 패스워드로 로그인을 시도한다.				
1	사용자 ID필드 클릭	사용자 ID 필드 반전	OK	
2	정상 사용자 ID 입력	사용자 ID 화면에 보임	OK	
3	패스워드 필드 클릭	패스워드 필드 반전	OK	
4	정상 패스워드 입력	화면에 보이지 않음	OK	
5	“로그인” 버튼을 클릭	로그인된 화면	OK	
2. 사용자 ID와 패스워드 없이 로그인을 시도한다.				
1	“로그인” 버튼을 클릭	로그인 실패창이 뜬	OK	실패 메시지의 “확인”버튼을 클릭하면, 정상적인 입력 데이터 형식을 보임
2	“확인” 버튼 클릭	로그인 화면으로 리턴	OK	
3. 정상 사용자 ID와 패스워드가 없이 로그인을 시도한다.				
1	사용자 ID필드 클릭	사용자 ID 필드 반전	OK	
2	정상 사용자 ID 입력	사용자 ID 화면에 보임	OK	
3	“로그인” 버튼을 클릭	로그인 실패창이 뜬	OK	실패 메시지의 “확인”버튼을 클릭하면, 정상적인 입력 데이터 형식을 보임
4	“확인” 버튼 클릭	로그인 화면으로 리턴	OK	
4. 정상 사용자 ID와 틀린 패스워드로 로그인을 시도한다.				
1	사용자 ID필드 클릭	사용자 ID 필드 반전	OK	
2	정상 사용자 ID 입력	사용자 ID 화면에 보임	OK	
3	패스워드 필드 클릭	패스워드 필드 반전	OK	
4	틀린 패스워드 입력	화면에 보이지 않음	OK	
5	“로그인” 버튼을 클릭	로그인 실패창이 뜬	OK	실패 메시지의 “확인”버튼을 클릭하면, 정상적인 입력 데이터 형식을 보임
4	“확인” 버튼 클릭	로그인 화면으로 리턴	OK	
5. 사용자 ID의 입력없이 정상적인 패스워드만으로 로그인 시도한다.				
1	패스워드 필드 클릭	패스워드 필드 반전	OK	
2	정상 패스워드 입력	화면에 보이지 않음	OK	
3	“로그인” 버튼을 클릭	로그인 실패창이 뜬	OK	실패 메시지의 “확인”버튼을 클릭하면, 정상적인 입력 데이터 형식을 보임
4	“확인” 버튼 클릭	로그인 화면으로 리턴	OK	

(표 5-5)는 사용자의 입력을 받아서 시스템의 인터페이스를 테스트하기 위해서 블랙박스 테스트 기법을 적용하였다. 아래의 (표 5-5)는 로그인 처리를 위한 테스트 케이스의 실행 결과이다.

5.2.4 테스트케이스 실행 오류처리

(표 5-6)은 테스트 시나리오별로 테스트를 실시한 결과의 한 예로써, 오류 처리에 대한 결과 보고서를 작성하여 결함에 대한 원인을 분석하고 수정여부를 확인하는 표를 나타낸다.

(표 5-6) 테스트 오류처리 결과 보고서

테스트 케이스ID	테스트케이스	결함원인	검토결과	비고
TS-04-TC-02	강의 수강후 시험 문제 리스트 출력	일반강의 수강후 A형 문제출력 안됨.	코드수정 후 확인-정상출력	일반강의 테스트: A타입문제 출력, 보충강의 테스트: B타입문제
TS-04-TC-04	시험평가 완료	해당 강의에 대한 시험응시가 반복됨.	에러코드 수정 후 정상출력	해당강의 수강 후 일반/보충 시험은 각 1번에 제한되어야함
TS-05-TC-01	강의등록	3주차 강의등록 에러	에러코드 수정 정상	교수자는 해당의를 등록/수정/삭제할 수 있다

5.3 시스템의 평가 및 활용방안

웹을 기반으로 한 통합 개발과정에서 필연적으로 발생되어지는 복잡함과

개발의 어려움은 시스템의 유지보수를 더욱 어렵게 한다. 특히 웹기반 교육 시스템에 대한 효율적인 개발과 유지보수 및 확장을 위해서는 시스템 개발의 생명주기에 따라 적절한 테스트가 이루어져야 한다.

본 논문에서는 사용자의 요구분석을 바탕으로 시스템간의 상호 작용을 정의하고 있는 시스템의 단위기능 즉, 유스케이스를 이용하여 시스템의 기능 테스트를 실시하였으며, 시퀀스 다이어그램을 이용하여 객체들 사이에서 주고받는 메시지의 순서를 보여주고, 시스템 실행 시 특정한 순간에 발생할 수 있는 객체들 사이의 상호작용을 보여주었다. 또한 객체들 사이의 동적인 관계를 분석할 수 있도록 하고 이를 바탕으로 프로그램을 실행 즉, 화이트박스 테스트기법을 이용한 모듈테스트를 실시하여 실행결과를 보였다. 마지막으로 기능적인 요구 사항에 집중하는 테스트 방법으로써, 화이트박스 테스트를 보완하는 블랙박스 테스트를 실시하였다.

현재 웹을 기반으로 한 시스템 구축에 있어서 더욱 제대로 된 테스트가 요구되어지는 이유는 앞서도 살펴보았듯이 웹 개발 환경의 변화에서 주된 원인을 찾을 수 있다. 특히 웹은 시스템 구조상 다양한 컴포넌트와 네트워크의 복합적 결합으로 작동되며, 충분히 검증되지 않은 신기술의 적용이 많기 때문에 그 성능을 이론적으로 예측하기가 매우 어렵다.

본 논문에서 기술한 테스트 기법과 함께 제한적이긴 하지만 웹 자동 테스트 툴의 활용을 고려 할 수 있다. 웹 자동 테스트 툴은 시스템의 주요기능과 특성에 대한 수동 테스트를 보완하기 위해 사용한다. 실제로 유용한 특정 영역은 스트레스 테스트와 부하 테스트, 회귀 테스트, 형상 테스트이다.

제 6 장 결론

본 논문은 더 복잡해지고 다양해지는 웹을 기반으로 한 교육 시스템의 효율성, 신뢰성, 정확성, 기능성들을 객체지향 테스트를 통해서 검증하고 그 중요함을 보였다. 이러한 웹 기반 시스템에 대해 효율적인 개발과 유지보수, 확장에서 좋은 품질을 유지하기 위해서는 시스템 개발에 따른 적합한 테스트 방법을 선정해야 한다. 이를 위해 논문에서는 웹 테스트를 위해서 C언어 기초교육을 위한 시스템을 설계하고 객체지향 테스트를 위해 UML 다이어그램에 기반한 테스트 케이스를 구축하였다. 특히 시스템의 요구사항을 기술한 시나리오를 바탕으로 유스케이스와 시퀀스 다이어그램을 이용하여 실제 테스트 케이스 생성과 테스트에 적용했다.

테스트 케이스를 유도하는 과정에서 사용된 유스케이스 및 시퀀스 다이어그램은 교과과정 및 학습 플로우의 내용이 충분히 반영되도록 설계하였다. 이것은 웹 기반 학습시스템의 테스트가 다른 일반 응용시스템의 테스트와 차별되어야 하는 중요한 사항이다. 개발자들은 단순한 기능의 테스트 뿐 아니라 학습자가 WBI를 통해 학습해나가는 교과과정 및 그 결과에 대한 평가와 학습플로우가 시스템에 충분히 반영되었는지 여부를 정확히 검증해야만 한다.

웹 기반 교육 시스템의 테스트가 갖는 중요성을 고려할 때 이 분야의 국내 연구는 아직 활성화되지 않은 상태이다. 기존의 테스트 자동화 툴도 그 기능에 제한이 존재한다. 본 논문에서 추출한 정보들은 웹 테스트를 위한 기초적인 정보가 될 수 있으며, 실제 테스트에 적용하는 것을 기술한다.

향후 연구로서 보다 발전된 웹 테스트를 위해서 사용자 또는 사용자 환경에 대한 패턴 및 경향 분석을 테스트에 고려하는 문제를 연구할 필요가 있다. 이를 위해서는 실제 상황에 근접한 테스트 케이스 작성 및 정확한 테스트 데이터 추출과 웹 자동 테스트 툴에 대한 연구가 진행되어야 하겠다.

참고 문헌

- [1] 방성준 · 이성근, “웹을 활용한 자기주도적 학습방법과 학습과제 유형이 학업성취도 및 학습태도에 미치는 영향”, 컴퓨터교육학회논문지 5(2), 2002.
- [2] Badrul H. Khan, “Web-Based Instruction(WBI) : What is it and why is it?”, Web-based instruction”, NJ : Educational Technology Publications, pp. 5-18, 1997.
- [3] 장상필 · 이영민, “웹 기반 학습시스템에 적합한 형성 평가 방법에 관한 연구”, 컴퓨터교육학회논문지 3(1), 2000.
- [4] 나일주, “웹기반 교육”, 교육과학사, 1999.
- [5] 백영균, “웹기반 학습의 설계”, 양서원, 1999.
- [6] John Vargo, “Evaluating the effectiveness of Internet delivered course work”, the Third Australian World Wide Web Conference(AusWeb97), 1997.
- [7] 최은만, “웹 기반 소프트웨어의 시험 및 검증 기술”, 정보과학회지 Vol. 4, No. 3, pp. 139-150, 2001.
- [8] Cornelia Boldyfeff, Kewish R., “Reverse engineering to achieve maintainable WWW site”, Reverse Engineering, Eighth Working Conference. pp. 249-257, 2001.
- [9] Roger Fournier, “웹 어플리케이션 개발 방법론”, 이한출판사, pp. 605 2002.

- [10] Rhonda Dibachi, "Testing e-commerce: Reducing your company's risk of doing business on the Web", *Software Testing & Quality Engineering Magazine*, pp. 57-62, Mar 1999.
- [11] Roger Fournier, "웹 어플리케이션 개발 방법론", 이한출판사, pp. 532-536, 2002.
- [12] J. Conallen, "Building Web Application Testing", *StickyMinds.com*, Feb 2001.
- [13] 정선미 · 최은만, "웹 테스트 자동화를 위한 테스트 스크립트 생성방법", *정보처리학회논문집*, 9권, 1호, 2002.
- [14] B.M Subraya, S.V. Subrahmanya, "Object driven performance testing of Web applications", *Proceedings. First Asia-Pacific Conference on Quality Software*, pp. 17-26, 2000.
- [15] C. Kallepalli, J. Tian, "Usage measurement for statistical web testing and reliability analysis", *Software Metrics Symposium, METRICS 2001, Proceedings. Seventh International*, pp. 148-158, 2001.
- [16] D. Kung, Chein-Hung Liu, Pei Hsia, "An Object-Oriented Web Test Model for Testing Web Applications", *Proceedings. First Asia-Pacific Conference on Quality Software*, pp. 111-120, 2000.
- [17] 강재성 · 윤광식 · 오승욱 · 권용래, "웹의 상태기반 기능시험 기법", *정보화학회 봄 학술발표논문집 Vol. 27, No. 1*. pp. 501-503, 2000.

[18] 권영호 · 최은만, “웹 기반 소프트웨어 테스트 모델에 관한 연구”, 정보처리학회 춘계 학술발표논문집 제8권 제1호, pp.197-200, 2001.

[19] 권영호 · 최은만 “웹 어플리케이션의 사용자 인터페이스 테스트 자동화 기법”, 정보처리논문지(D), Vol. 10-D, No. 02, pp. 293-300, 2003.

[20] 한동수 · 정인상 공저, “소프트웨어 테스트 입문”, 한국소프트웨어공학협회/소프트웨어테스팅분과, 2004.

[21] Myers, G., “The Art of Software Testing”, Wkley & Sons, 1979.

[22] 김은주, “시나리오를 이용한 객체 지향 시스템의 통합 테스트”, 동국대학교, 1997.

ABSTRACT

Design and Implementation of a Web-Based Instruction System using Object-oriented Testing Methodology

Yi, Hye Lee

Major in Computer Science Education

Graduate School of Education

Sungshin Women's University

With rapid growth of the web based information technology, the legacy systems are being integrated on the web. This is also the case in education systems. The Web based Instruction(WBI) should be seen as a new form of training methods going beyond the existing electronic training.

An effective form of education should be performed by mutual effort between instructor and learner accordingly. Also, such requests should be precisely reflected to the system. An effective development and maintenance of the system must be in support. In this trend, an appropriate test is required for the development of reliable systems.

In this thesis, designing the test cases which are appropriate for developing web based instruction systems reflecting more complicated

requests from the users are provided. A system for the basic training of the C language has been implemented. The UML design artifacts including use case and sequence diagrams were used on the actual testing. The information abstracted in this thesis can be the basic information for web testing and the information which can be applied to the actual testing is described.