

목차

1. 요구사항 명세서

2. E-R Modeling

1) Entity Set

2) Relationship Set

3) E-R diagram

3. 웹 애플리케이션 시스템

1) 사용자 기능

2) 관리자 기능

4. 트랜잭션 개념 적용 예

1. 요구사항 명세서

기술사시험위원회는 회원들이 필기, 실기 시험을 통과해 자격증을 취득하기까지의 모든 과정을 주관한다. 이러한 관리에 필요한 시스템을 구축할 것이다.

이 시스템을 이용하기 위해서는 회원가입해야 한다. 회원에는 관리자와 이용자가 있으며, 둘은 접근할 수 있는 페이지들이 다르다. 회원이 아닌 사람은 시험에 대한 어떤 정보도 볼 수 없다. 모든 서비스 신청은 온라인 상에서 이루어질 수 있다. 이 시스템은 이용자에게 24시간 서비스를 제공한다. 서비스는 크게 회원관리, 시험정보관리, 응시관리, 지부관리로 나뉜다.

<회원관리>

회원관리 기능은 가입 회원을 체계적으로 관리하는 기능이다. 각 회원은 고유의 username을 갖는다. 회원가입은 웹상에서 이루어지고, 가입 시 회원은 username, 비밀번호, 이름, 생년월일, 연락처, 이메일 주소를 입력하고 관리자 여부를 선택한다. 이메일 주소와 관리자 여부 외에는 반드시 입력해야 하는 사항이다. 관리자 여부는 기본값이 NO이며, 회원이 선택하지 않는다면 NO로 저장된다.

(관리자) 관리자는 가입된 모든 회원의 정보를 열람할 수 있다. 관리자가 수정 가능한 회원의 정보는 연락처와 이메일 주소뿐이다. 회원의 비밀번호는 관리자도 알 수 없다. 관리자는 회원의 username으로 회원을 탈퇴시킬 수 있다. 회원 탈퇴 시 회원의 모든 정보가 시스템 상에서 완전히 삭제된다.

(이용자) 회원은 가입 후 본인의 계정 정보를 조회할 수 있으며, 비밀번호, 연락처, 이메일 주소를 수정할 수 있고, 원하면 탈퇴할 수 있다. 단, 탈퇴 시 시스템에 저장된 모든 정보가 사라진다. 회원은 본인이 응시 신청한 시험의 목록을 모두 열람할 수 있으며, 성적이 이미 공개된 경우 이까지도 열람할 수 있다. 회원은 직접 성적을 입력하거나 수정/삭제할 수 없다.

<시험정보관리>

시험정보관리 기능에서는 시험과목과 시험일정을 관리한다. 시험과목에 대해서 과목코드, 과목 이름, 분야, 필기/실기 시험 응시료, 시험에 대한 기본정보를 저장할 수 있다. 시험은 정해진 시험일정에만 치러지는데, 시험일정은 고유한 시험일정코드로 구분한다. 시험일정코드에 따라 시험의 필기/실기 시험 응시 신청 가능 날짜, 필기/실기 시험 날짜, 필기/실기 시험 결과 발표 날짜가 정해지고, 같은 시험일정코드를 여러 시험이 공유할 수 있다. 각 과목은 시험일정코드를 부여 받아야 비로소 회원에 의해 시험이 신청되고 응시될 수 있다. 각 과목은 특정 시험일정에만 시험이 치러지므로 어떤 시험은 1년에 한 번, 어떤 시험은 매달 시험이 있을 수 있다.

(관리자) 관리자는 시험과목을 열람, 추가, 수정, 삭제할 수 있다. 과목 추가 시 과목코드, 과목 이름, 분야, 필기/실기 시험 응시료를 반드시 입력해야 한다. 관리자는 또 시험과목 정보 중 일부를 수정할 수 있는데, 응시료, 시험과목 정보는 수정 가능하지만, 과목의 핵심 정보인 과목코드, 과목 이름, 과목 분야는 수정할 수 없다. 과목 삭제는 가능하다. 또한 관리자는 시험일정을 관리한다. 시험일정코드에 따라 시험일정이 결정되는데, 관리자는 이 시험일정코드를 추가, 삭제할 수 있다. 관리자는 과목코드에 시험일정코드를 배정해 유효한 시험을 만들어내고, 경우에 따라 삭제도 할 수 있다.

(이용자) 회원은 시험과목 설명을 열람할 수 있지만 입력, 수정, 삭제는 불가능하다. 회원은 현재 신청 가능한, 즉 관리자가 시험일정코드를 배정했고, 그 시험일정코드의 시험 신청 날짜가 현재 날짜를 포함하고 있는 시험들을 확인할 수 있다. 동시에 시험일정코드에 대한 정보도 같이 열람할 수 있다. 따라서 필기시험을 신청한다면 해당 과목에 대한 실기시험이 언제 있을 예정인지도 확인 가능하다.

<응시관리>

응시관리 기능은 원서접수, 시험장 안내, 합격자 발표 등 시험 응시에 필요한 직접적인 절차를 관리한다. 원서접수 내역은 접수번호로 구분하며 접수번호에는 고유한 숫자가 부여된다. 시험 접수는 시험에 배정된 시험일정코드의 시험 신청 기간 내에만 가능하다. 실기 시험 접수는 필기 시험을 합격한 후에만 응시할 수 있다. 시험 응시 연도와 시험일정코드가 같은 시험은 중복해서 신청할 수 없다. 합격자 발표는 홈페이지에서 공개적으로 하지 않으며 개인이 응시한 시험에 대해서만 합격 여부를 확인 가능하다.

(관리자) 관리자는 회원들이 시험 신청한 내역을 모두 열람할 수 있다. 관리자 임의로 이 내용은 수정/삭제할 수 없다. 관리자는 회원들이 응시한 시험의 점수 및 합격 여부를 열람할 수 있다. 그리고 합격자 발표날 응시자의 점수와 합격 여부를 입력해준다.

(이용자) 원서 접수 시 접수 가능한 시험을 확인하여 온라인 폼에 시험과목, 시험일정코드, 필기/실기, 시험장을 정확하게 입력한다. 문자 서비스를 선택하면 시험 전날 문자 메시지로 알림을 받을 수 있다. 합격자 발표일 이후에 본인의 시험 신청 내역에서 점수와 합격 여부를 확인할 수 있다.

<지부관리>

본 위원회는 네 개의 지부를 가지고 있고 각 지부는 1-2개의 시험장을 관리하고 있다. 모든 시험은 모든 시험장에서 치러진다.

(관리자) 관리자는 모든 지부와 시험장에 대한 정보, 즉 지부코드, 시험장코드, 시험장 이름, 수용 인원을 조회할 수 있다.

(이용자) 이용자는 원서 접수 시 시험장 이름만 확인 가능하다.

2. E-R Modeling

1) Entity Set

Entity는 branch, exam, examarea, examlist, examperiodlist, examresult, member, subject, testregister로 총 9개이다.

branch				
Attribute	Description	Data type	PK	Not Null
branchcode	지부의 고유코드	CHAR(3)	✓	✓
branchname	지부 이름	CHAR(10)		✓
representative	지부 대표 이름	CHAR(20)		

exam				
Attribute	Description	Data type	PK	Not Null
subejctid	고유한 과목코드	CHAR(5)	✓	✓
examtype	필기/실기	ENUM (written', 'performance')	✓	✓
registrationfee	응시료	INT		

examarea				
Attribute	Description	Data type	PK	Not Null
examareaaid	시험장의 고유코드	CHAR(5)	✓	✓
examareaname	시험장 이름	CHAR(20)		✓
branchcode	소속 지부	CHAR(10)		✓
capacity	시험장의 최대 수용 가능 인원	INT		

examlist				
Attribute	Description	Data type	PK	Not Null
subjectid	고유한 과목코드	CHAR(5)	✓	✓
examyyear	시험이 치러지는 연도	YEAR	✓	✓
examperiod	시험일정코드	CHAR(1)	✓	✓
examtype	필기/실기	ENUM (‘written’, ‘performance’)	✓	✓

examperiodlist				
Attribute	Description	Data type	PK	Not Null
examperiod	시험일정의 고유코드	CHAR(1)	✓	✓
wregstart	필기시험 원서접수 시작 날짜	DATE		✓
wregend	필기시험 원서접수 마지막 날짜	DATE		✓
writtentest	필기시험 날짜	DATE		✓
wresult	필기시험 합격발표 날짜	DATE		✓
pregstart	실기시험 원서접수 시작 날짜	DATE		✓
pregend	실기시험 원서접수 마지막 날짜	DATE		✓
performancetest	실기시험 날짜	DATE		✓
finalresult	최종 합격 발표 날짜	DATE		✓

examresult				
Attribute	Description	Data type	PK	Not Null
registrationnum	원서접수신청서의 고유번호	INT	✓	✓
examdate	시험 응시 날짜	DATE		
score	시험에서 얻은 점수	INT		
passorfail	합격/불합격	ENUM (‘pass’,‘fail’)		

member				
Attribute	Description	Data type	PK	Not Null
username	고유한 회원ID	CHAR(12)	✓	✓
member_name	회원 이름	CHAR(20)		✓
dob	회원의 생년월일	DATE		✓
phone	회원의 핸드폰번호	CHAR(13)		✓
email	회원의 이메일 주소	CHAR(20)		
password	회원이 설정한 비밀번호	CHAR(100)		✓
admin	관리자 여부 (default : NO)	CHAR(3)		✓

subject				
Attribute	Description	Data type	PK	Not Null
subjectid	고유한 과목코드	CHAR(5)	✓	✓
subjectname	과목명	CHAR(30)		✓
subjectarea	과목의 상위 분야	CHAR(30)		✓
description	과목에 대한 기본정보, 출제 경향, 우대현황 등	CHAR(200)		

testregister				
Attribute	Description	Data type	PK	Not Null
registrationnum	원서접수신청서의 고유번호 (auto_increment)	INT	✓	✓
username	고유한 회원ID	CHAR(12)		✓
date	원서접수신청서 작성 날짜	DATE		✓
subjectid	고유한 과목코드	CHAR(5)		✓
examyyear	시험이 치러지는 연도	YEAR		✓
examperiod	시험일정코드	CHAR(1)		✓
examtype	필기/실기	ENUM (‘written’, ‘performance’)		✓
examareaid	시험장의 고유코드	CHAR(5)		✓
textservice	시험 관련 알림 메시지 수신 여부	CHAR(3)		

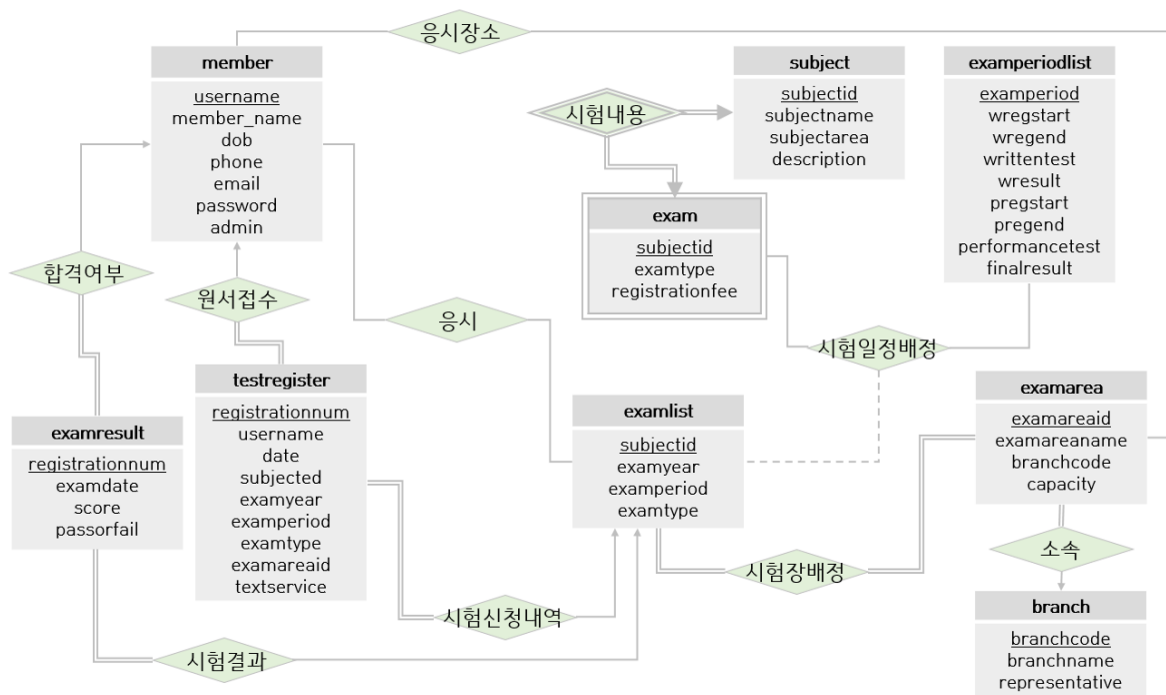
2) Relationship Set

Relationship은 응시, 합격여부, 원서접수, 시험신청내역, 시험결과, 응시장소, 시험일정배정, 시험내용, 시험장배정, 소속으로 총 10개이다.

Relationship	Description	Related entities	Cardinality		Attribute
			최소	최대	
응시	<ul style="list-style-type: none"> 한 회원은 0개 이상의 시험에 응시한다. 한 시험은 0명 이상의 회원에 의해 응시된다. 	member	0	n	
		examlist	0	n	
합격여부	<ul style="list-style-type: none"> 한 회원은 0개 이상의 합격내역을 가지고 있다. 합격내역은 한 회원의 합격내역을 표시한다. 	member	0	n	
		examresult	1	1	
원서접수	<ul style="list-style-type: none"> 한 회원은 0개 이상의 원서접수신청서를 작성한다. 하나의 원서접수신청서는 오직 한 명의 회원에 의해서만 작성된다. 	member	0	n	
		testregister	1	1	
시험신청내역	<ul style="list-style-type: none"> 한 원서접수신청서는 하나의 시험에 접수한다. 한 시험은 여러 원서접수신청서를 통해 접수된다. 	testregister	1	1	
		examlist	0	n	
시험결과	<ul style="list-style-type: none"> 한 시험은 0개 이상의 시험결과를 갖는다. 시험결과는 1개의 시험의 결과만 가지고 있다. 	examlist	0	n	
		examresult	1	1	

Relationship	Description	Related entities	Cardinality		Attribute
			최소	최대	
응시장소	<ul style="list-style-type: none"> - 회원은 0개 이상의 시험장에서 시험에 응시한다. - 한 시험장에서 0명 이상의 회원이 응시한다. 	member	0	n	
		examarea	0	n	
시험일정배정	<ul style="list-style-type: none"> - 어떤 과목의 시험은 0개 이상의 시험일정을 갖는다. - 시험일정은 0개 이상의 시험에 배정된다. 	exam	0	n	examlist
		examperiodlist	0	n	
시험내용	<ul style="list-style-type: none"> - 한 과목에 대한 시험은 한 개이다. - 한 시험은 한 개의 과목에만 관련된다. 	subject	1	1	
		exam	1	1	
시험장배정	<ul style="list-style-type: none"> - 한 시험은 1개 이상의 시험장을 갖는다. - 한 시험장에서 1개 이상의 시험이 치러진다. 	examlist	1	n	
		examarea	1	n	
소속	<ul style="list-style-type: none"> - 한 시험장은 지부 하나의 소속이다. - 한 지부는 0개 이상의 시험장을 가진다. 	examarea	1	1	
		branch	0	n	

3) E-R diagram

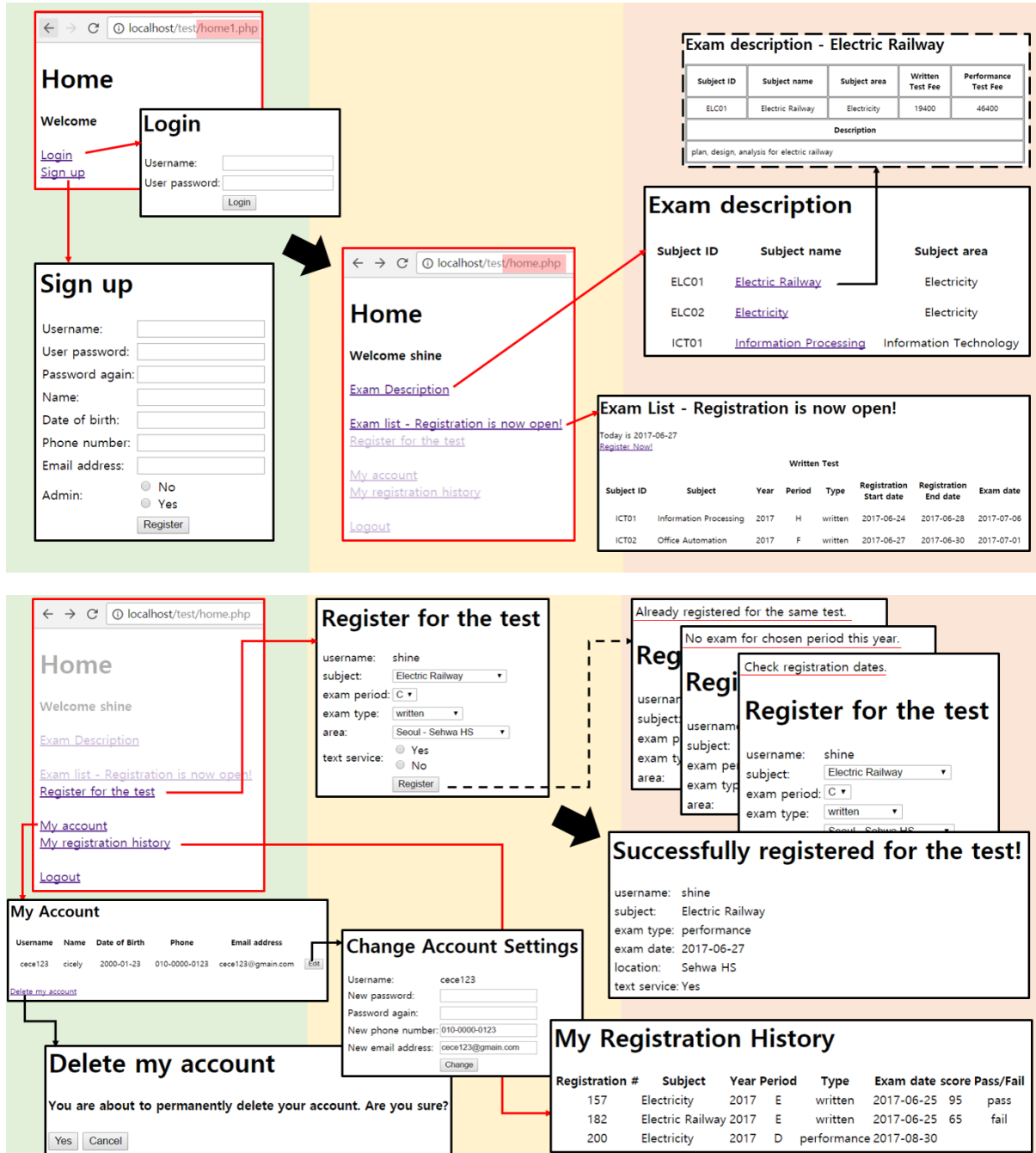


3. 웹 애플리케이션 시스템

시스템의 홈 화면은 로그인이 되어 있지 않을 때 home1.php, 로그인이 되어 있을 때 home.php이다. home.php로 이동하려고 해도 로그인이 되어 있지 않다면 자동으로 home1.php로 이동하고, 로그인이 되어 있다면 home.php로 이동한다. 시스템을 사용하다가 로그아웃하면 역시 home1.php으로 자동으로 이동한다. 이것은 로그인할 때 저장한 \$_SESSION['username']이 isset인지 코드 상단 부에서 확인함으로써 자동으로 이루어진다.

본 시스템은 일반사용자와 관리자가 접근할 수 있는 페이지가 다르다. 관리자만 접근할 수 있는 페이지는 admin.php, adminmemberlist.php 등 admin으로 시작한다. 관리자는 일반사용자가 접근할 수 있는 모든 페이지에 접근할 수 있다. 이렇게 분리된 페이지에서 다른 기능을 제공한다.

1) 일반사용자 기능 (동영상 참고)



2) 관리자 기능 (동영상 참고)

localhost/test/admin.php

Admin page

Welcome admin

Member list

Exam

View all registration

View all exam results

Branch

Logout

Member List

Username	Name	Date of Birth	Phone	Email address	Admin
admin	admin	1900-01-01	010-0000-0000	admin@	
admin999	admin999	2000-01-01	010-0000-9999	admin999@	
cece123	cicely	2000-01-23	010-0000-0123	cece123@	
shine	Ayoung	1900-01-01	010-0000-4261	kim0100	

Edit

Delete

Admin Edit Member

Username	Name	Date of Birth	Phone	Email address	Admin
admin	admin	1900-01-01	010-0000-0000	admin@gmail.com	YES

Username:

New phone number:

New email address:

Change

Member Delete

Username to delete:

Delete

Admin branch

Anyang(ANY, Choi)

ANY01	Gwiin MS	150
ANY02	Sinki MS	210

Busan(BSN, Lee)

BSN01	Haeundae MS	180
-------	-------------	-----

Exam Results

Registration #	username	Subject ID	Year	Exam period	Exam type	Exam date	score	Pass/Fail
136	shine	ELC01	2017	D	written	2017-06-24	90	pass
151	flora512	ELC01	2017	E	written	2017-06-25	80	pass
230	shine	ELC01	2017	E				

Update result

Registration # :

score :

Pass or Fail :

Update

Cancel

localhost/test/admin.php

Admin page

Welcome admin

Member list

Exam

View all registration

View all exam results

Branch

Logout

Admin Exam

Welcome admin

Admin Exam List

Admin Exam Description

Admin Exam Period

Admin Exam List

Subject ID	Exam Year	Period
ELC01	2017	C
ELC01	2017	D
MTL02	2017	G

Add

Delete

Subject ID :

Exam year :

Exam period :

Add Exam

Back to Exam List

Admin Exam Period

Ignore the year.

Exam Period	Registration Start date(w)	Registration End date(w)	Exam date(w)	result(w)	Registration Start date(p)	Registration End date(p)	Exam date(p)	Final Result
C	2000-06-21	2000-06-25	2000-06-25	2000-06-25	2000-06-25	2000-06-26	2000-06-26	2000-06-26
D	2000-06-22	2000-06-24	2000-06-24	2000-06-24	2000-06-25	2000-06-25	2000-08-30	2000-09-15

H

20

Add

Delete

H

2000-06-24

2000-06-28

2000-07-06

2000-07-0

Add

Cancel

H

2000-06-24

2000-06-28

2000-07-06

2000

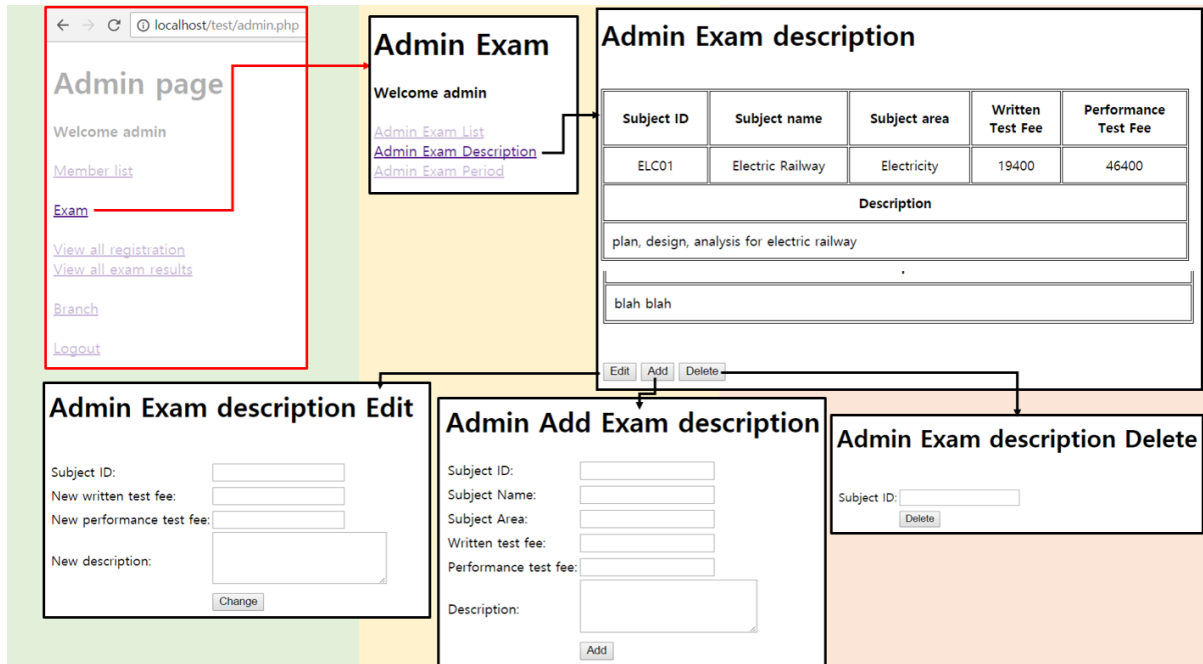
Insert exam period to delete :

Delete

Cancel

Exam Registration

Registration #	username	Registration date	Subject	Exam year	Exam period	Exam type	Area	Text service
136	shine	2017-06-24	ELC01	2017	D	written	SEL02	Yes
151	flora512	2017-06-25	ELC01	2017	E	written	DGU01	Yes



4. 트랜잭션 개념 적용 예

둘 이상의 쿼리를 처리할 때 flag를 두고 하나라도 오류가 발생하면 모든 쿼리가 rollback되게 했다. 예를 들어 adminexamdescadd.php는 관리자가 시험 과목을 추가하는 코드이다. 과목코드, 과목 이름, 과목 상위 분야, 설명, 필기시험 응시료, 실기시험 응시료를 웹 상에서 입력하고, 이것을 DB에 삽입하는 데 세 번의 쿼리가 필요하다. 코드 상단에서 autocommit을 해제하고 에러가 없을 시에만, 즉 flag가 true로 남아 있을 때만 비로소 commit하게 한다.

```
mysqli_autocommit($db, false);
$flag = true;
```

```
$sql1 = "INSERT INTO subject VALUES('".$subjectid."','".$subjectname."','".$subjectarea."','".$description."";
$sql2 = "INSERT INTO exam VALUES('".$subjectid."','".$written."','".$wtfee."')";
$sql3 = "INSERT INTO exam VALUES('".$subjectid."','".$performance."','".$ptfee."')";
$result = mysqli_query($db, $sql1);
if(!$result){
    $flag = false;
    echo "Error details : ".mysqli_error($db).".";
}
$result = mysqli_query($db, $sql2);
if(!$result){
    $flag = false;
    echo "Error details : ".mysqli_error($db).".";
}
$result = mysqli_query($db, $sql3);
if(!$result){
    $flag = false;
    echo "Error details : ".mysqli_error($db).".";
}

if($flag){
    mysqli_commit($db);
    echo "Saved successfully.";
} else{
    mysqli_rollback($db);
    echo "All queries were rolled back";
}
```

이외에도 adminexamdescdelete.php에서 삭제하는 두 쿼리에 대해, adminexamdescedit.php에서 UPDATE하는 세 쿼리에 대해 트랜잭션 개념을 적용하였다. 그러나 examlist.php처럼 SELECT만 여러 번 나오는 코드에서는 굳이 트랜잭션을 고려하지 않았다.

page1.php은 응시 신청하는 페이지로, 이 시스템에서 가장 핵심적인 부분 중 하나인데, 여기서는 특별히 flag를 0에서 4까지 쓰면서 각 경우에 대해 대비하고 경우에 따라 다른 메시지를 표시했다. 시험을 정상적으로 신청했을 때는 flag가 1로 남아 모든 쿼리가 처리되지만, 0, 2, 3, 4로 변경된 경우에는 모든 쿼리가 실행되지 않는다. 현재 날짜가 시험 신청 기간이 아닐 경우 flag가 0으로, 이미 같은 시험에 응시 신청한 경우 flag가 2로, written test를 pass 통보 받지 않고 해당 과목 performance test에 응시 신청했을 때 flag가 3으로, 시험일정과 과목을 잘못 매칭시켜 시험을 신청한 경우 flag가 4로 바뀌어 이 네 경우에는 코드의 모든 쿼리들이 rollback된다.