

기사 시험 접수, 취소를 편리하게!

# 기사 시험 신청 시스템

대덕인재개발원 초급 프로젝트



4조 만두네  
(이혜진, 김선민, 정은비)

# 목 차

1. 조원 소개 및 역할 분담
2. 주제 선정 배경
3. 개념/논리/물리 ERD
4. UML 다이어그램
5. 구현 예시
6. 전체 프로세스 흐름도
7. 프로젝트 시연
8. Q&A



## 조원 소개 및 역할 분담

작지만 웅글찬 4조 만두네



**이혜진**

회원가입, 로그인 기능 및 DB 구현



**김선민**

시험 접수 기능 및 DB 구현



**정은비**

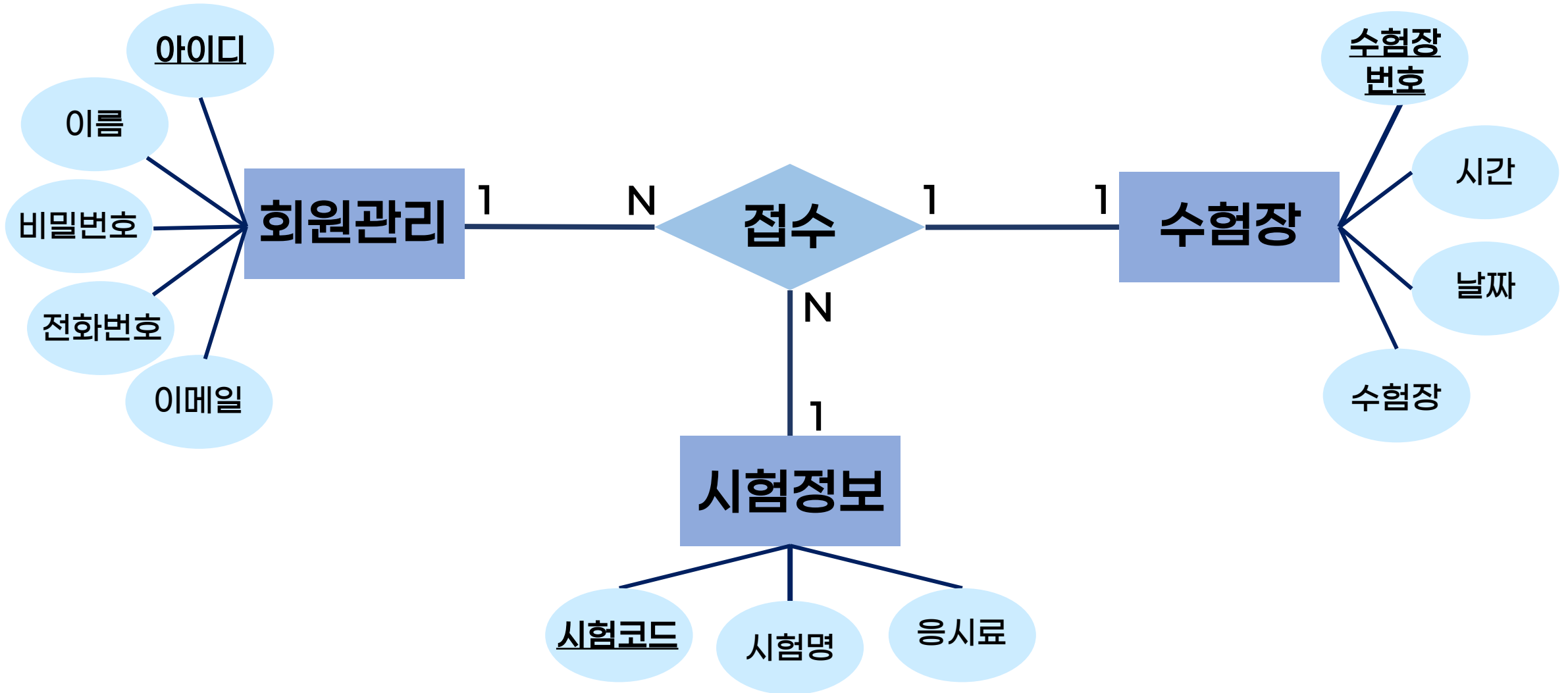
콘솔 UI 기능 및 DB 구현

## 주제 선정 배경



초급 프로젝트의 취지인 **CRUD**를 사용할 수 있는 주제로 시험 접수 시스템을 선정하였고,  
사용자가 직접 회원가입을 하고 시험 과목 및 수험장을 직접 선택하여 능동적인 참여를 할 수 있도록  
**JAVA** 프로젝트와 **DB**를 연동해 프로그램을 구현하였습니다.

# 개념 ERD



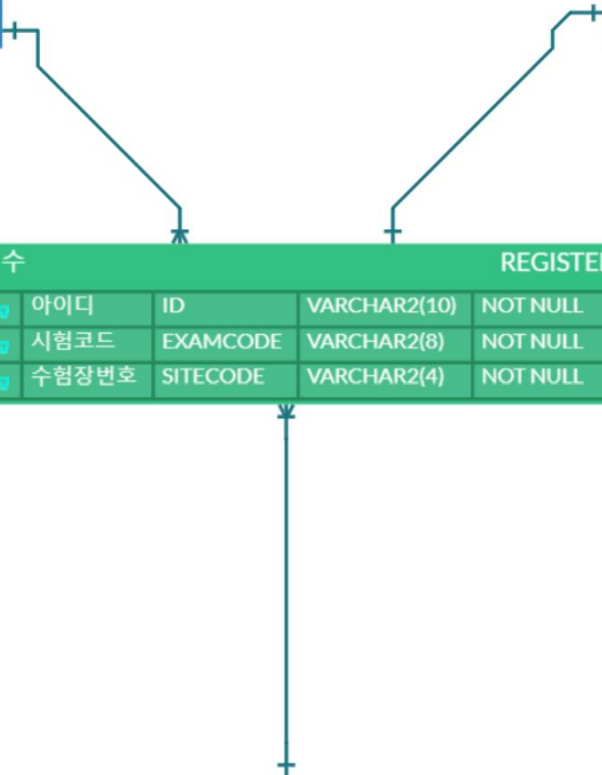
# 논리/물리 ERD

회원관리		EXAMINEE		
아이디	ID	VARCHAR2(40)	NOT NULL	
이름	NAME	VARCHAR2(10)	NOT NULL	
비밀번호	PASSWORD	VARCHAR2(20)	NOT NULL	
전화번호	TELNO	VARCHAR2(11)	NOT NULL	
이메일	EMAIL	VARCHAR2(50)	NOT NULL	

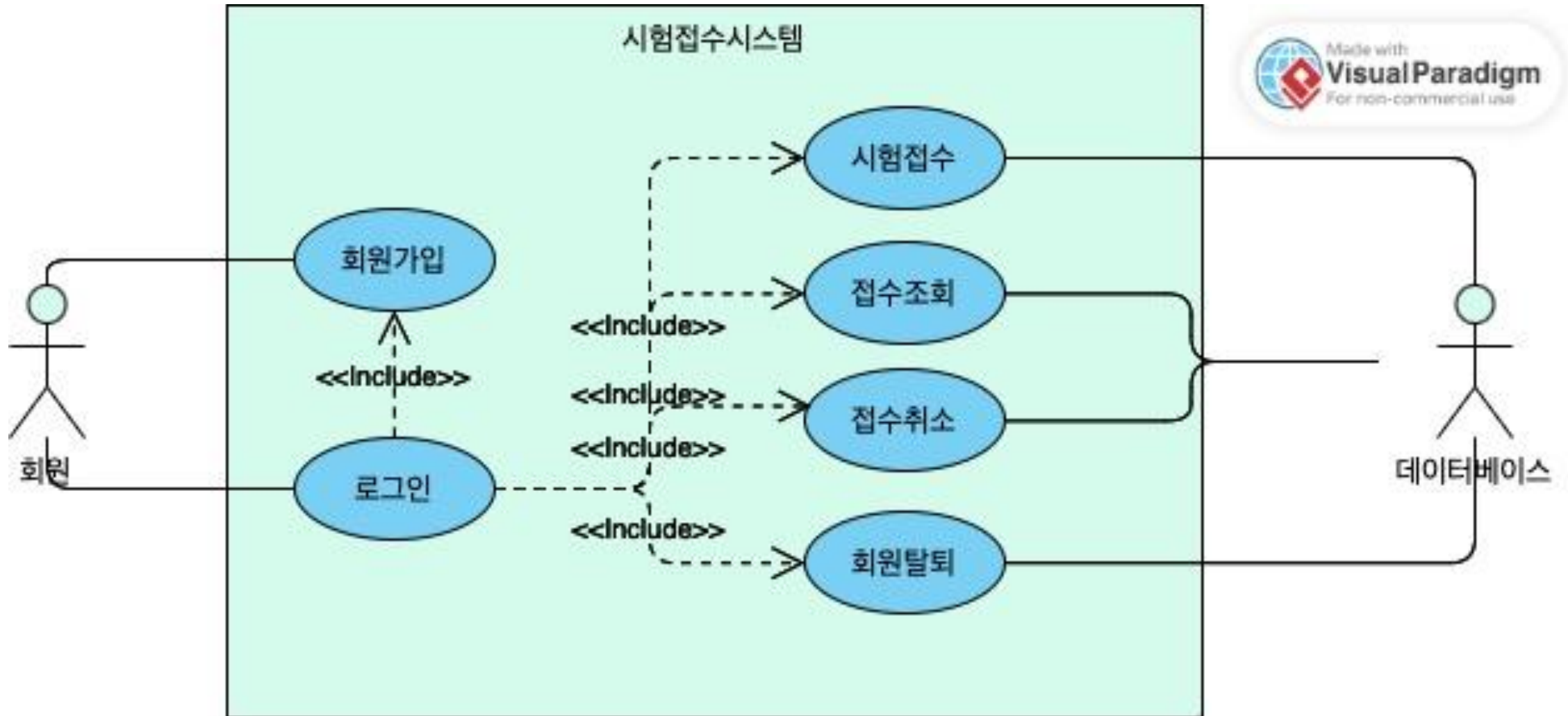
수험장		EXAMSITE		
수험장번호	SITECODE	VARCHAR2(4)	NOT NULL	
시간	EXAMTIME	VARCHAR2(10)	NOT NULL	
날짜	EXAMDATE	DATE	NOT NULL	
수험장	SITENAME	VARCHAR2(20)	NOT NULL	

접수		REGISTER		
아이디	ID	VARCHAR2(10)	NOT NULL	
시험코드	EXAMCODE	VARCHAR2(8)	NOT NULL	
수험장번호	SITECODE	VARCHAR2(4)	NOT NULL	

시험 정보		EXAM		
시험코드	EXAMCODE	VARCHAR2(8)	NOT NULL	
시험명	EXAMNAME	VARCHAR2(40)	NOT NULL	
응시료	FEE	VARCHAR2(15)	NOT NULL	



# 유스케이스 다이어그램



# 클래스 다이어그램

ExamineeVO
- id: String - name: String - password: String - telNo: String - email: String
+ ExamineeVO(String, String) + ExamineeVO(String, String, String) + ExamineeVO(String, String, String, String, String) + getId(): String + setId(): void + getName(): String + setName(): void + getPassword(): String + setPassword(): void + getTelNo(): String + setTelNo(): void + getEmail(): String + setEmail(): void + toString(): String

ExamSiteVO
- siteCode: String - examTime: String - examDate: String - siteName: String
+ ExamSiteVO(String sitecode, String examtime, String examdate, String sitename) + getSiteCode(): String + setSiteCode(String Sitecode): void + getExamTime(): String + setExamTime(String Examtime): void + getExamDate(): String + setExamDate(String Examdate): void + getSiteName(): String + setSiteName(String Sitename): void

RegisterView
- scanner: Scanner
+ welcome(): void + signMenu(): void + insertSignIn(): ExamineeVO + login(): ExamineeVO + registerMenu(): void + registerSeq(): RegisterVO + registerCheckMenu(): void + showRegister(): void + modifyInfoMenu(): void + modifySignInMenu(): ExamineeVO + modifyExamInfoMenu(): RegisterVO + deleteRegister(): RegisterVO + deleteExaminee(): ExamineeVO

RegisterVO
- id: String - examcode: String - sitecode: String
+ getId(): String + setId(String id): void + getExamCode(): String + setExamCode(Exam Vo): void + getSiteCode(): String + setSiteCode(ExamSiteVo): void

ExamVO
- examCode: String - examName: String - fee: String
ExamVO(String examcode, String examname, String fee) getExamCode(): String setExamCode(String Examcode): void getExamName(): String setExamName(String Examname): void getFee(): String setFee(String Fee): void

RegisterService
+ selectRegisterInfos(): List<RegisterVo> + insertRegisterInfo(RegisterVo vo): int + deleteRegisterInfo(RegisterVo vo): int + updateExamCode(RegisterVo vo): int + updateExamDate(RegisterVo vo): int + updateExamSite(RegisterVo vo): int

RegisterDAO
+ selectRegisterInfos(): List<RegisterVo> + insertRegisterInfo(RegisterVo vo): int + deleteRegisterInfo(RegisterVo vo): int + updateExamCode(RegisterVo vo): int + updateExamDate(RegisterVo vo): int + updateExamSite(RegisterVo vo): int

ExamineeDAO
+ insertMyInfo(ExamineeVO vo): int + deleteMyInfo(ExamineeVO vo): int + updateMyInfo(ExamineeVo vo): int

ExamineeService
+ insertMyInfo(ExamineeVO vo): int + deleteMyInfo(ExamineeVO vo): int + updatePassword(ExamineeVo vo): int + updateTelNo(ExamineeVo vo): int + updateEmail(ExamineeVo vo): int



# 구현 예시

(q를 누르면 프로그램을 종료합니다.)

시작화면

1. 회원가입 | 2. 로그인

> 1

회원정보를 입력하세요

아이디:

이름:

비밀번호:

전화번호:

이메일:

회원가입이 완료되었습니다. 첫 페이지로 돌아갑니다.

인사

1. 회원가입 | 2. 로그인

> 2

아이디:

비밀번호:

로그인 성공!

1. 접수하기 | 2. 접수내역 조회

> 1

수험장을 선택하세요.

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11.

> n

대전(DJ)를 선택하셨습니다.

시험 과목을 선택하세요.

1. 2. 3. 4. 5. 6. 7. 8. 9. 10.

> m

시험 회차를 선택하세요.

1. 2. 3.

> o

n지역 m과목 o회차 오전 9시 시험을 선택하셨습니다. 계속 진행하시겠습니까? [y/n]

> y

응시료는 19400원입니다.

결제 방식을 선택하세요.

1. 신용카드 | 2. 계좌이체 | 3. 휴대폰 결제

> 3

접수가 성공적으로 되었습니다.  
돌아감)

// 접수화면

// 2-1. 접수하기 시작

// 2-1. 접수하기 끝(2 로그인 직후로

# 구현 예시

1. 접수하기 | 2. 접수내역 조회

> 2

// 2-2. 접수내역 조회 시작

1. 접수 내역 확인 | 2. 회원 정보 및 접수 내역 수정

> 1

// 2-2-1. 접수 내역 확인 시작

~회원 아이디, 시험장, 과목, 응시 일시~

돌아가시겠습니까?[y/n]

직후로 돌아감)

// 2-2-1. 접수 내역 확인 끝(2 로그인

1. 접수 내역 확인 | 2. 회원 정보 및 접수 내역 수정

> 2

// 2-2-2. 접수 내역 수정 시작

수정할 항목을 선택하세요

1. 회원 정보 | 2. 접수 정보

> 1

// 2-2-2-1. 회원 정보 수정 시작

1. 비밀번호 | 2. 전화번호 | 3. 이메일 | 4. 회원탈퇴

> 1 -> 바꿀 비밀번호를 입력하세요: 비밀번호 update

> 2 -> 전화번호 update

> 3 -> 이메일 update

// 2-2-2-1. 회원 정보 수정 끝

수정할 항목을 선택하세요

1. 회원 정보 | 2. 접수 정보

> 2

1. 시험장 | 2. 과목 | 3. 응시회차 | 4. 접수 취소

> 1 -> 시험장 update

> 2 -> 과목 update

> 3 -> 응시회차 update

// 2-2-2-2. 시험장 수정

// 2-2-2-3. 과목 수정

// 2-2-2-4. 응시장 수정

1. 접수 내역 확인 | 2. 회원 정보 및 접수 내역 수정

> 3

// 2-2-3. 접수 취소 시작

~회원 아이디, 시험장, 과목, 응시 일시~ 접수를 취소하시겠습니까?[y/n]

> y -> delete, 성공적으로 취소되었습니다.

> n -> 접수화면 돌아갑니다.

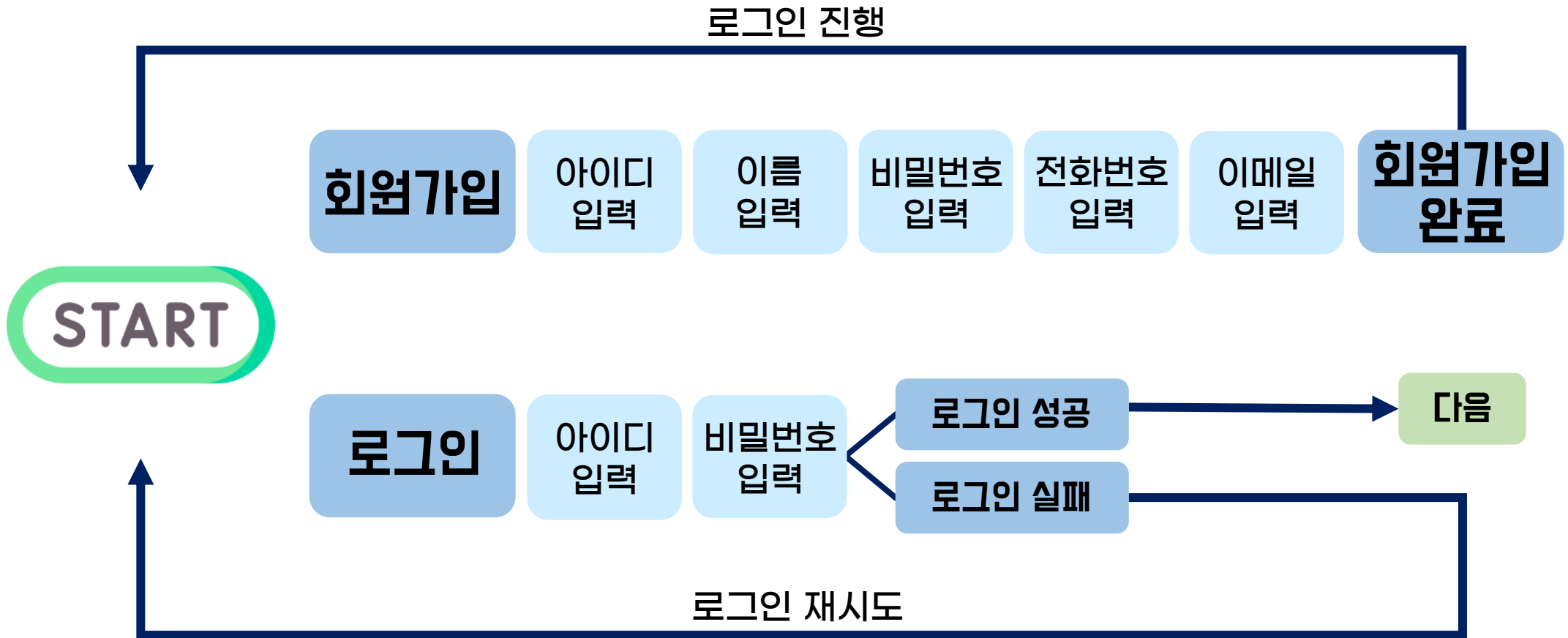
// 2-2-3. 접수취소 끝(2 로그인 직후

로 돌아감)

# 전체 프로세스 흐름도



## 회원가입 및 로그인



# 전체 프로세스 흐름도



## 접수하기

접수내역 조회

접수하기

수험장  
선택

시험과목  
선택

응시일시, 응시회차  
선택

응시료  
결제

시험 접수  
완료

로그인 성공

접수내역  
조회

접수내역 확인

접수내역 출력

회원정보 수정 및 접수 취소

다음

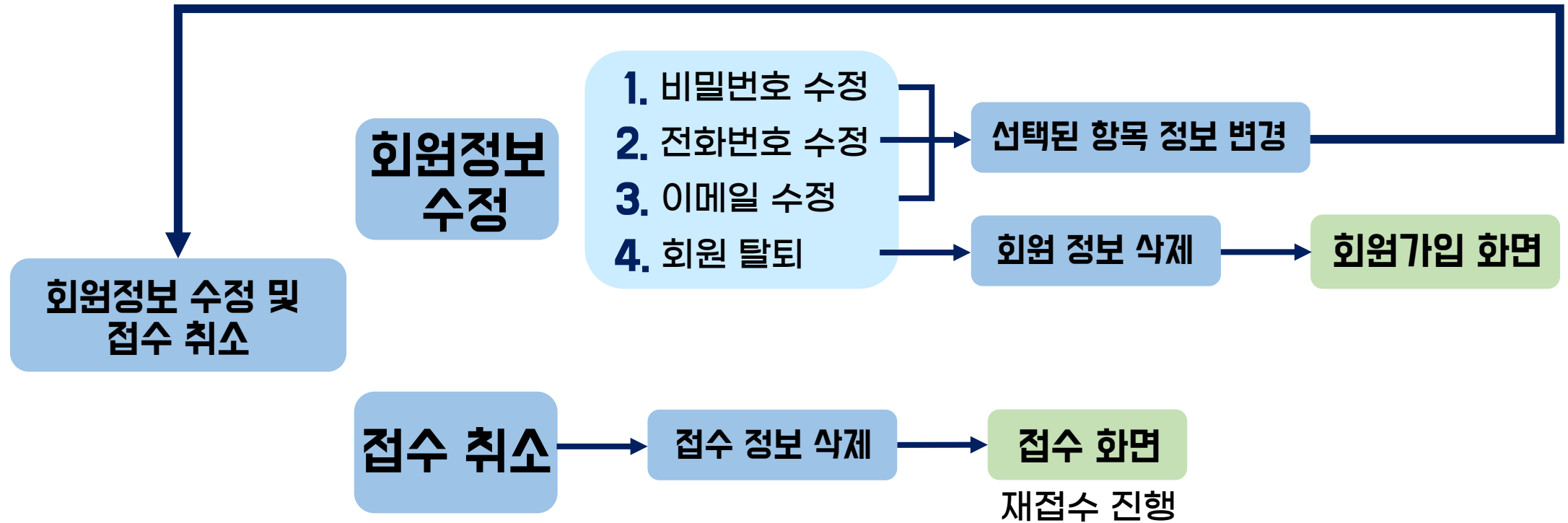
회원정보 수정 및 접수 취소

# 전체 프로세스 흐름도



## 회원정보 수정 및 접수 취소

회원정보 재수정 or 접수 취소



# 프로젝트 시연



# Q&A

