



ONLINE JUDGE

# Agenda

001

## 프로젝트 개요

- 1.1 추진배경
- 1.2 목적/목표

005

## 아키텍처 설계

- 5.1 소요기술
- 5.2 시스템 아키텍처

002

## 팀구성 및 개발일정

- 2.1 팀 구성
- 2.2 프로젝트 일정

006

## 화면 설계

- 6.1 메뉴 구조
- 6.2 주요 화면 레이아웃

003

## 프로세스 설계

- 3.1 프로세스 분할도
- 3.2 프로세스 설계서

007

## 시연

- 7.1 시연 시나리오
- 7.2 동영상 시연

004

## 테이블 설계

- 4.1 테이블 ERD

008

## 질의 응답

- 8.1 Q & A
- 8.2 후기

# WHY

## 프로젝트 개요

---

1.1 추진배경

1.2 목적/목표

---



## 프로젝트 개요

### 1.1 추진배경

### 1.2 목적/목표

## 1. 학습의 막연함

### 문제

정수를 저장하는 큐를 구현한 다음, 입력으로 주어지는 명령을 처리하는 프로그램을 작성하시오.

명령은 총 여섯 가지이다.

- push X: 정수 X를 큐에 넣는 연산이다.
- pop: 큐에서 가장 앞에 있는 정수를 빼고, 그 수를 출력한다. 만약 큐에 들어있는 정수가 없는 경우에는 -1을 출력한다.
- size: 큐에 들어있는 정수의 개수를 출력한다.
- empty: 큐가 비어있으면 1, 아니면 0을 출력한다.
- front: 큐의 가장 앞에 있는 정수를 출력한다. 만약 큐에 들어있는 정수가 없는 경우에는 -1을 출력한다.
- back: 큐의 가장 뒤에 있는 정수를 출력한다. 만약 큐에 들어있는 정수가 없는 경우에는 -1을 출력한다.

### 입력

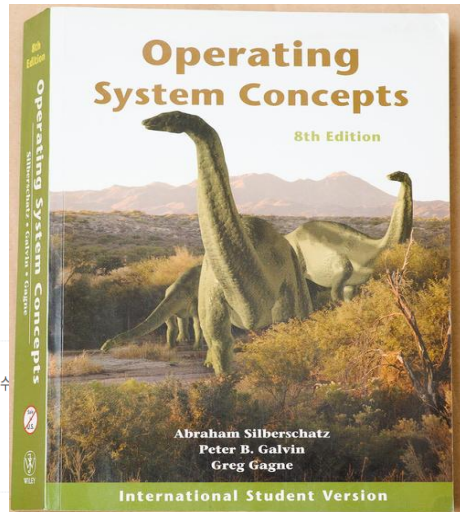
첫째 줄에 주어지는 명령의 수  $N$  ( $1 \leq N \leq 2,000,000$ )이 주어진다. 둘째 줄부터  $N$ 개의 줄에는 명령이 하나씩 주어진다. 주어지는 정수거나 같다. 문제에 나와있지 않은 명령이 주어지는 경우는 없다.

### 출력

출력해야하는 명령이 주어질 때마다, 한 줄에 하나씩 출력한다.

예제 입력 1 복사

예제 출력 1 복사

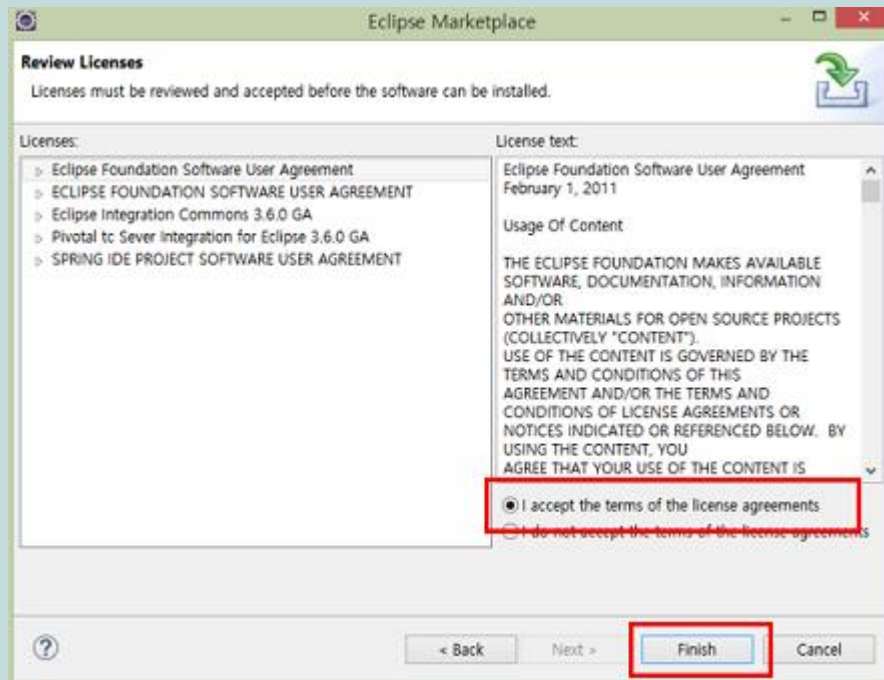


## 프로젝트 개요

### 1.1 추진배경

### 1.2 목적/목표

## 1. 학습의 막연함



## 프로젝트 개요

### 1.1 추진배경

### 1.2 목적/목표

## 1. 접근이 쉬운 코딩이론 콘텐츠

## 2. 온라인 코딩 테스트 실습환경



programmers

대시보드 내 프로젝트 스킵 체크 개발자 채용 월간지 코딩테스트 연습 프로그래밍 강의

## 2020 Dev-Matching

웹 백엔드 개발자 (상반기)

### 2020 Dev-Matching: 웹 백엔드 개발자 (상반기)

접수하기

7일 5시간 48분  
접수 종료까지

정수 20년 03월 30일 12:00 ~ 04월 17일 18:00  
진행 20년 04월 18일 13:00 ~ 04월 18일 15:00

JS Python Java C++

전형 정보 일정 채용 공고 목록

## 프로젝트 개요

### 1.1 추진배경

### 1.2 목적/목표

# 목적

온라인 코딩 실습환경 과  
학습콘텐츠 제공

---

## 프로젝트 개요

### 1.1 추진배경

### 1.2 목적/목표

# 목표

**1** QUIZ 를 통해 스스로  
진도 확인 이 가능한 학습환경

---

**2** ONLINE 에서 코드를 실행한뒤  
결과에 대해 비교가 가능한 척도 제공

---

비교척도 : 정답유무, 실행시간, 메모리 사용량 기준



WHO  
WHEN

## 팀구성 / 개발일정

---

2.1 팀 구성

2.2 개발일정

---



## 팀구성/개발일정

### 2.1 팀구성

### 2.2 개발일정

# 팀구성

## 박용민

업무진행 점검  
이론학습 기능 구현  
프로그램 목록 정의

## 허은열

화면설계  
프론트엔드 구현  
게시판 기능 구현

## 김현욱

프로젝트 기획  
DB설계/프로세스설계  
코드실행 및 검사구현

## 이유동

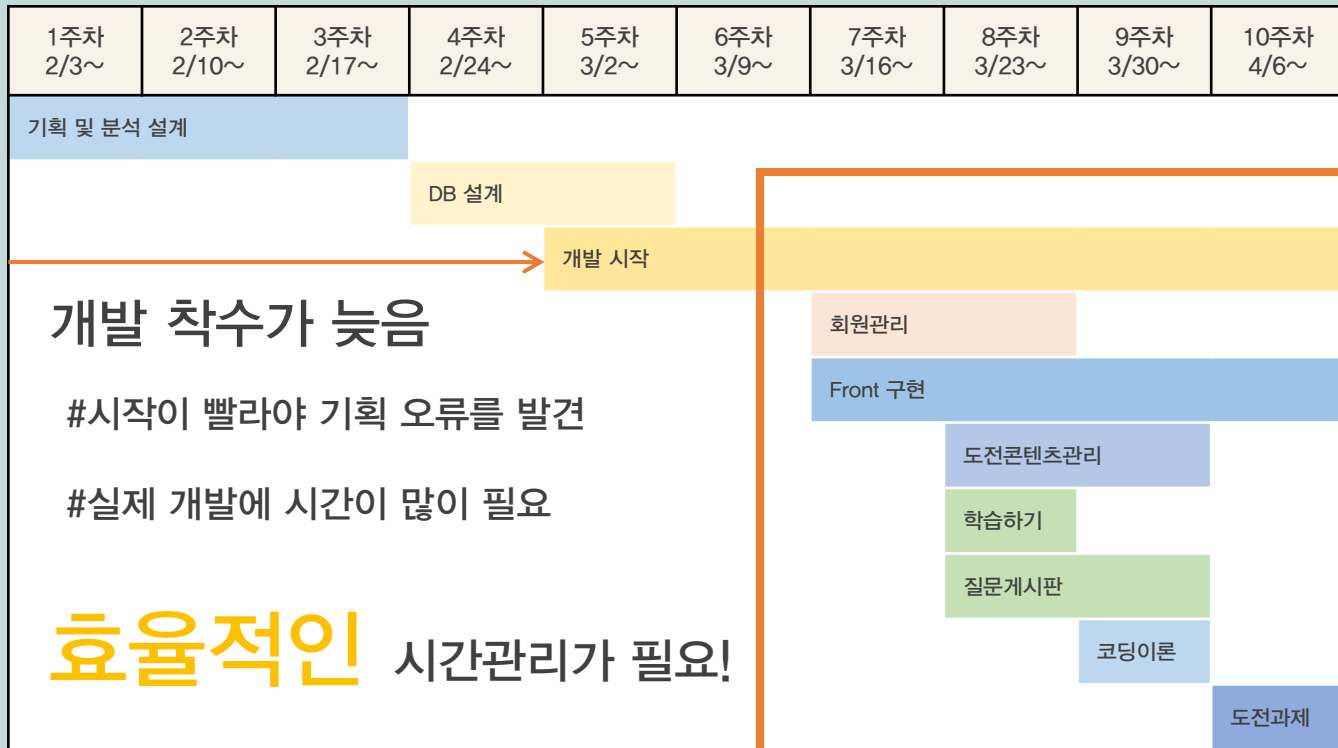
DB설계  
프로세스 설계  
회원 서비스 구현

## 팀구성/개발일정

### 2.1 팀구성

### 2.2 개발일정

## 개발일정



## 팀구성/개발일정

### 2.1 팀구성

### 2.2 개발일정

# 개발일정

**1** 개인의 산출물(소스코드)을  
합치는 과정이 반복

---

**2** 문제 발생 시 원인 추적이 어렵다  
(누가, 언제, 어디서, 무엇을 했는지)

---

## 팀구성/개발일정

### 2.1 팀구성

### 2.2 개발일정

## 개발일정



# GitHub

깃허브를 통한 프로젝트 관리

# HOW

## 프로세스 설계

---

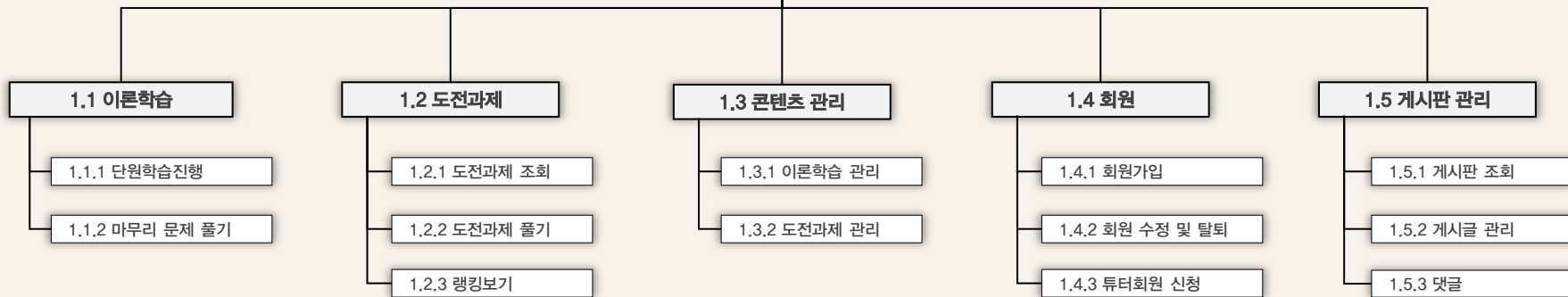
3.1 프로세스 분할도

3.1 프로세스 설계서

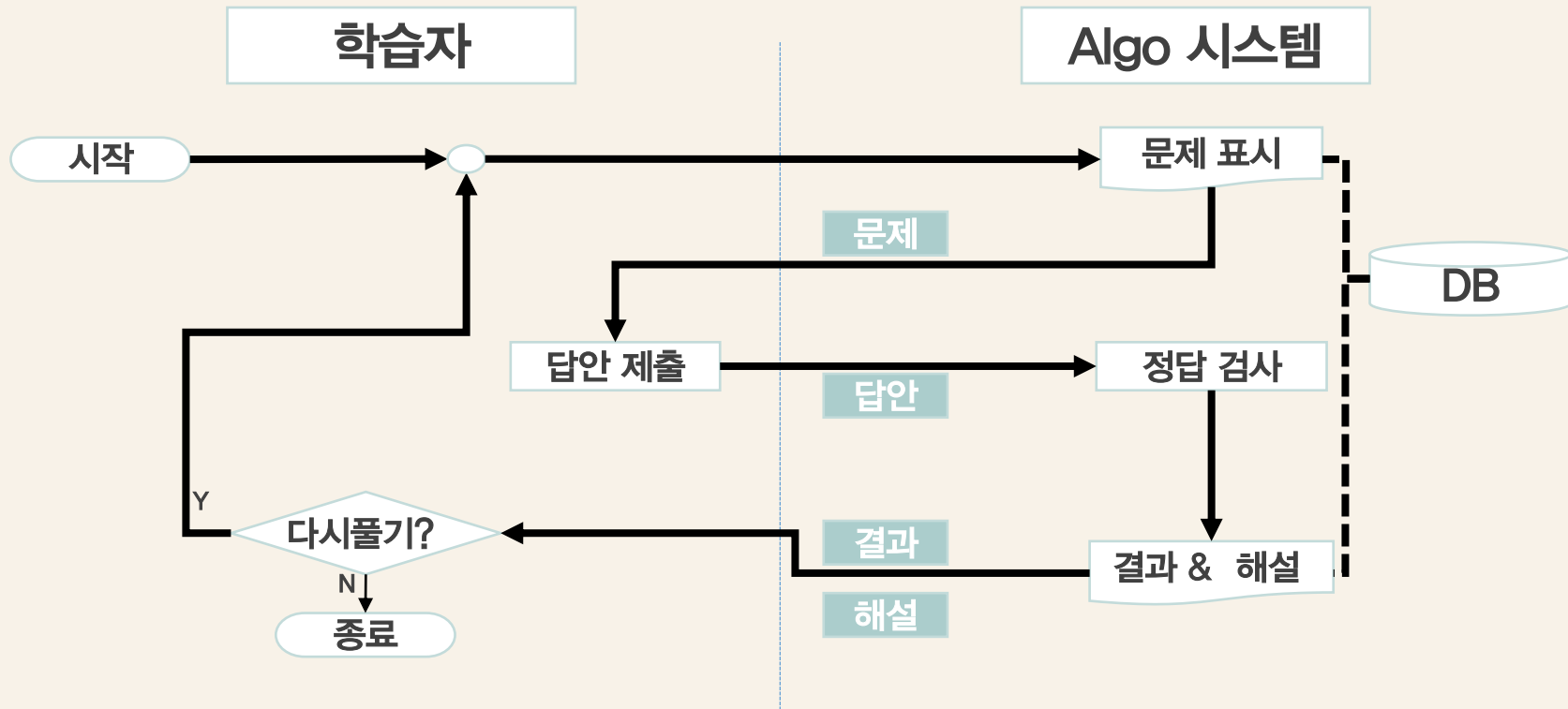
---

## 3.1 프로세스 분할도

# 1. Algo



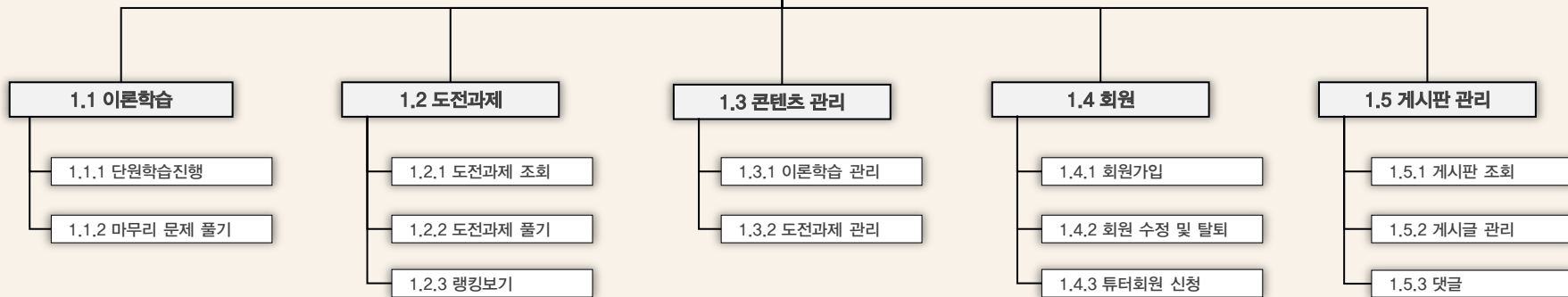
# 단원 마무리문제 풀기



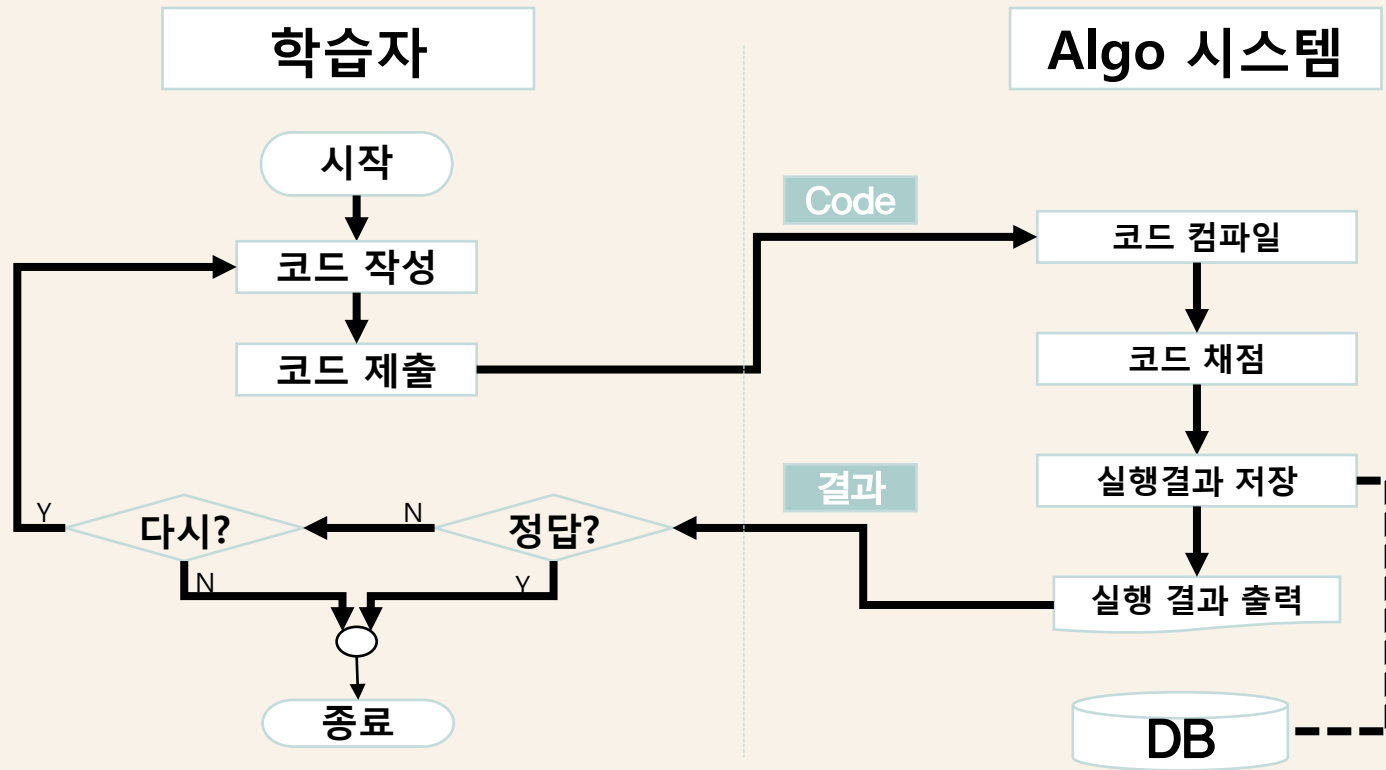


## 3.1 프로세스 분할도

# 1. Algo



# 도전과제 풀기



# HOW

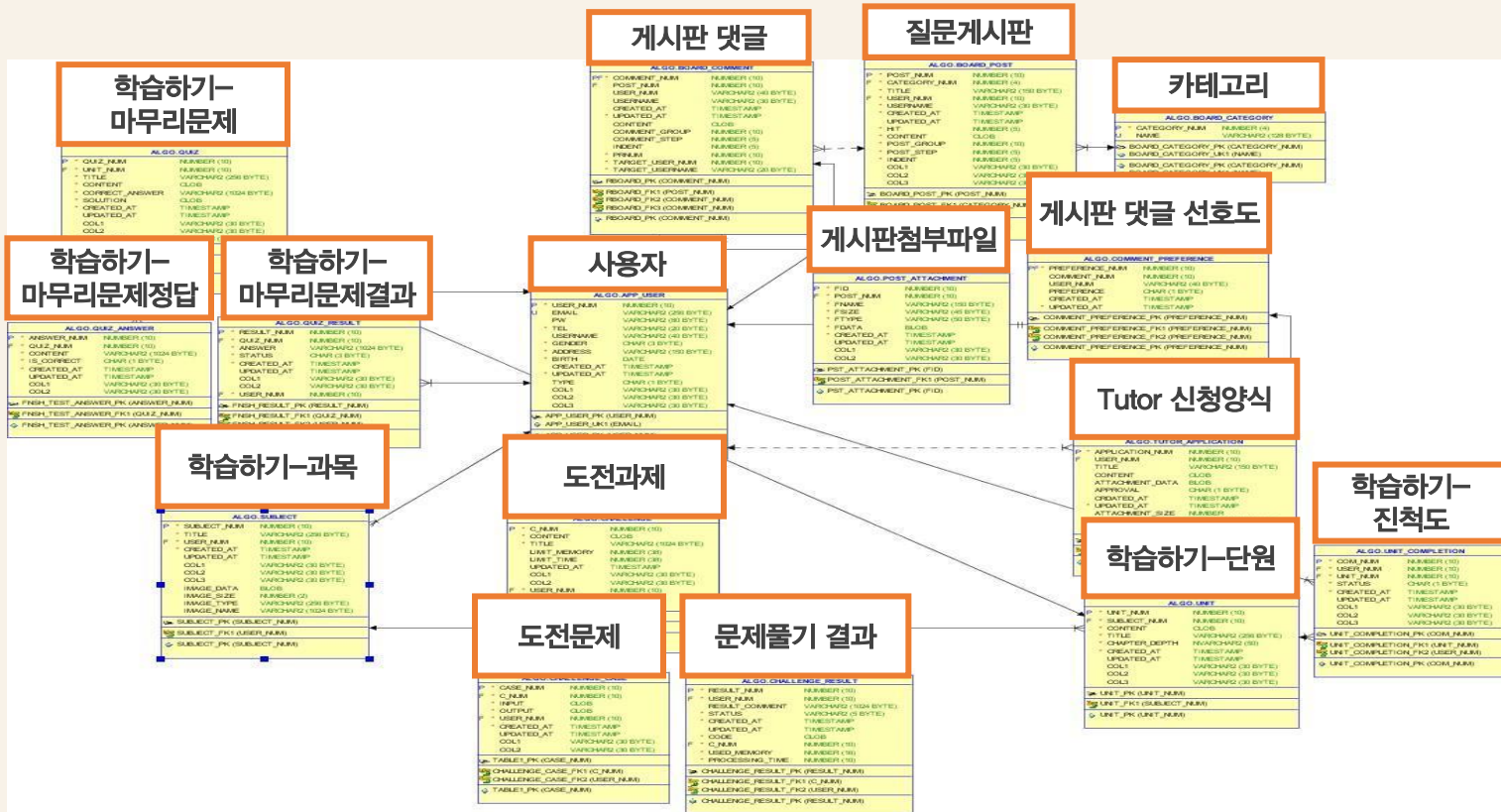
## 테이블 설계

---

### 4.1 테이블 ERD

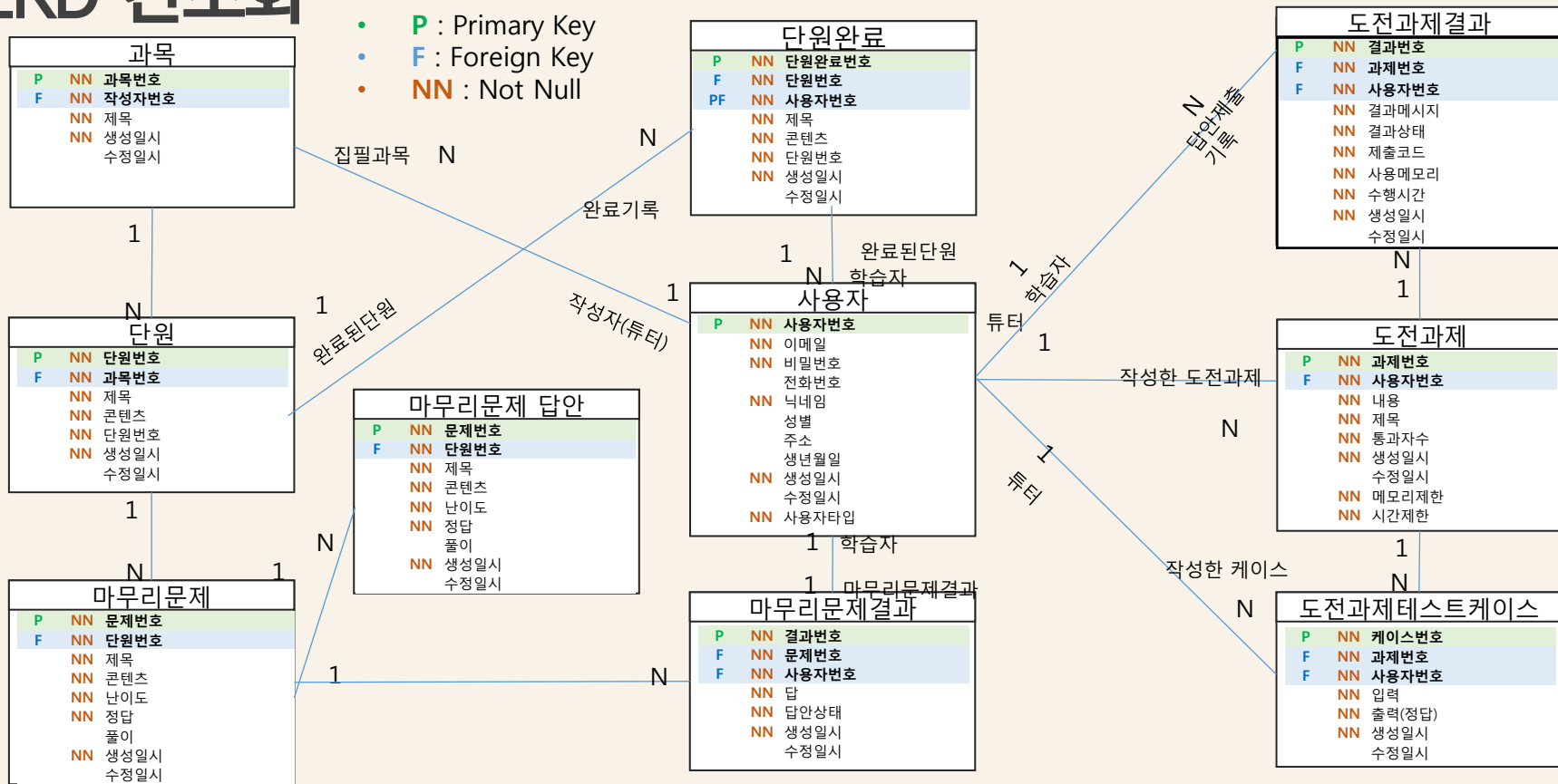
---

# 4



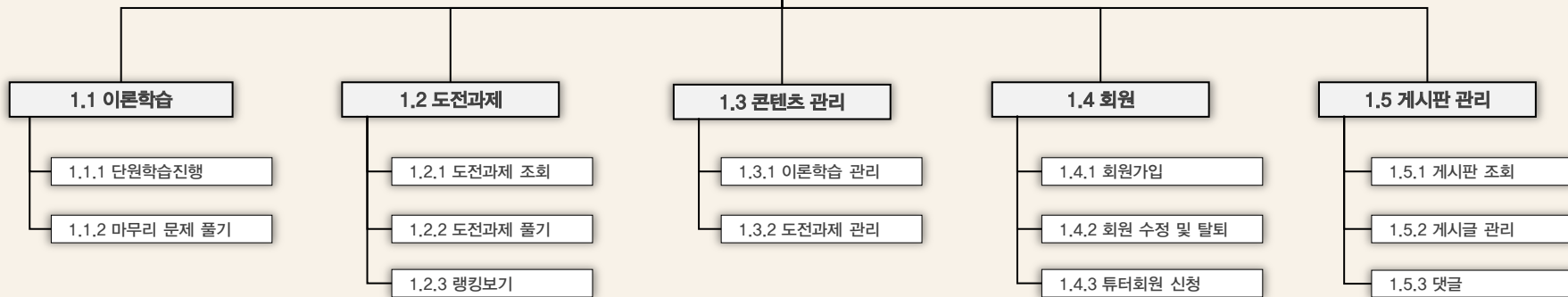
# ERD 간소화

- P : Primary Key
- F : Foreign Key
- NN : Not Null

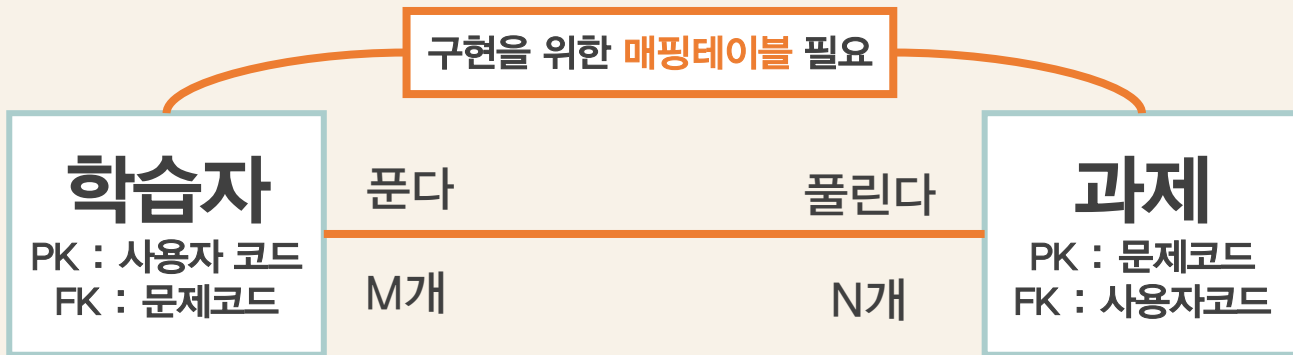


## 3.1 프로세스 분할도

# 1. Algo



# ERD (도전과제:과제풀기)

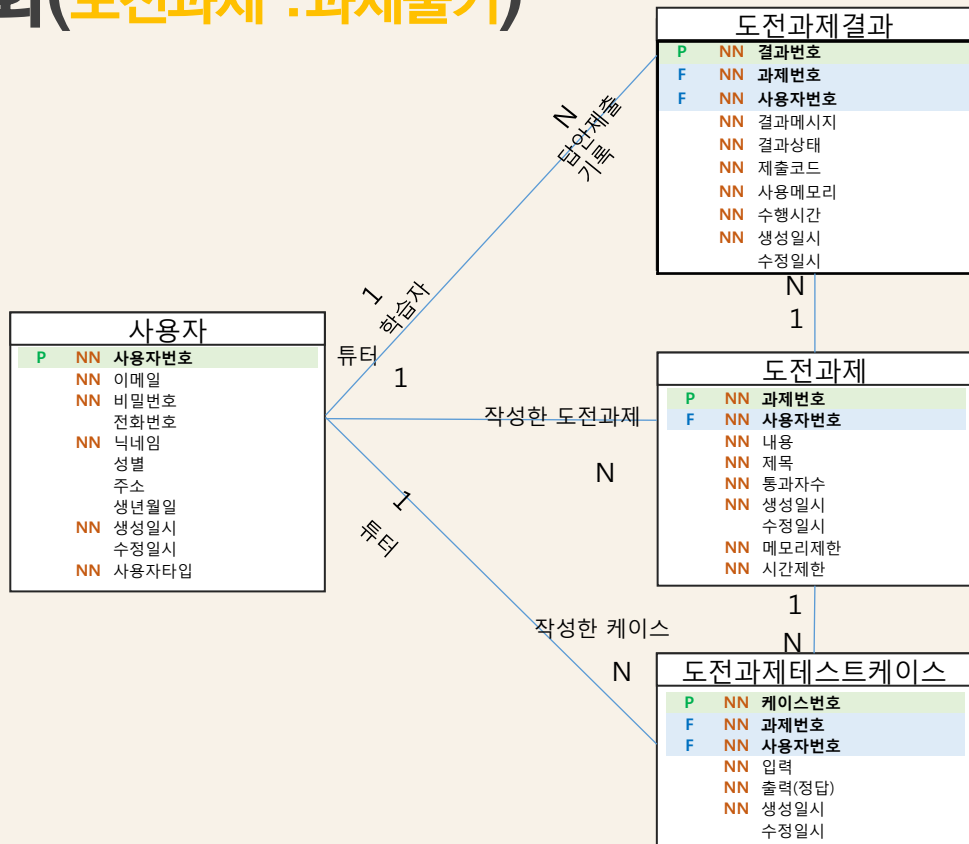


# ERD (도전과제:과제풀기)



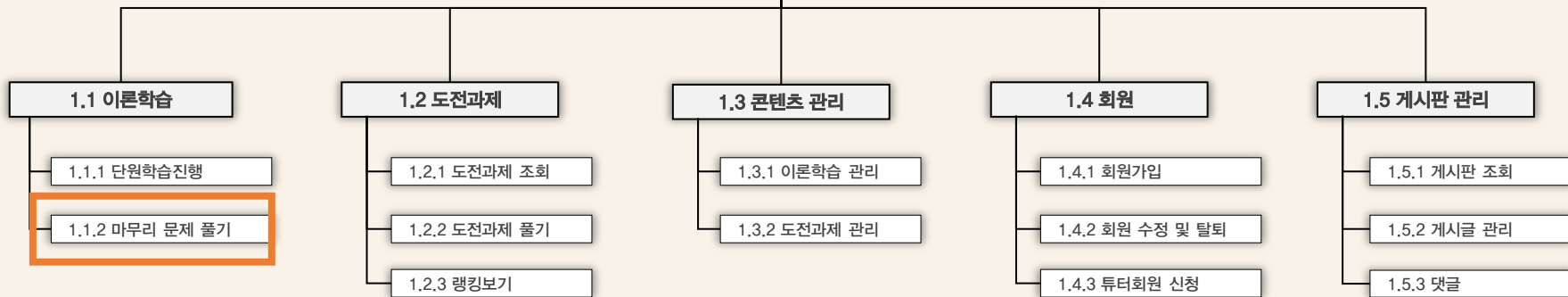


# ERD 간소화(도전과제 : 과제풀기)

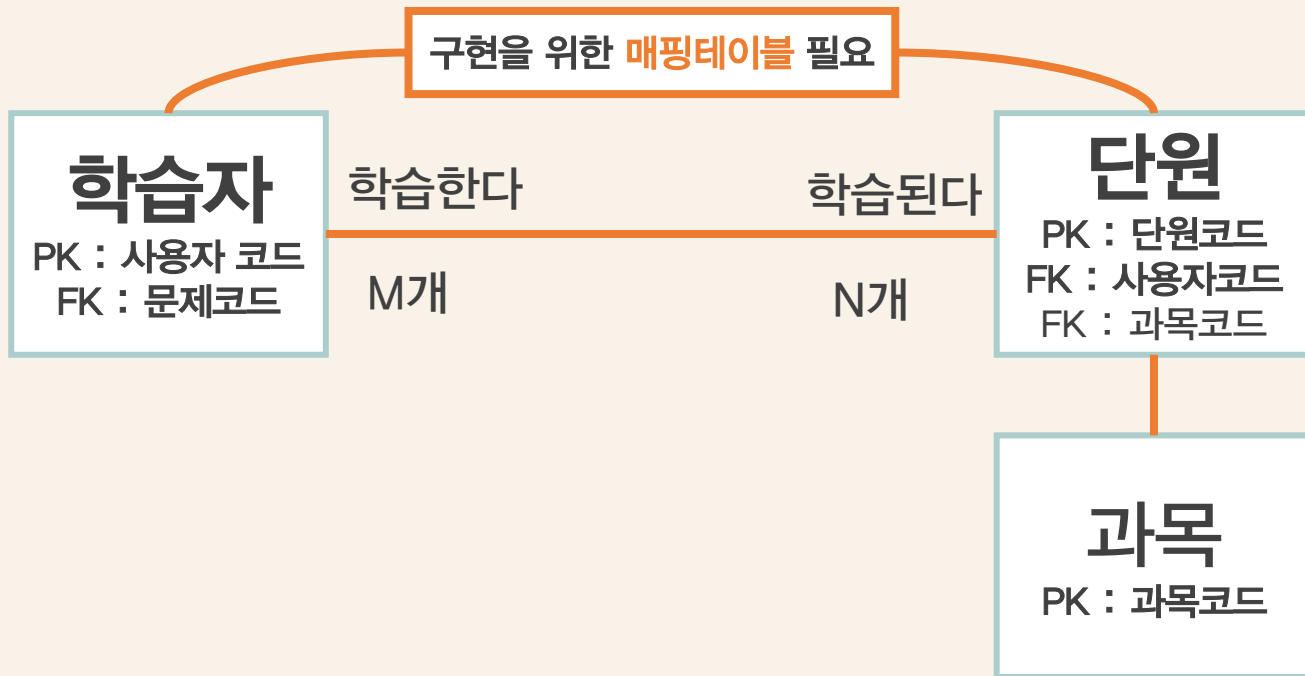


## 3.1 프로세스 분할도

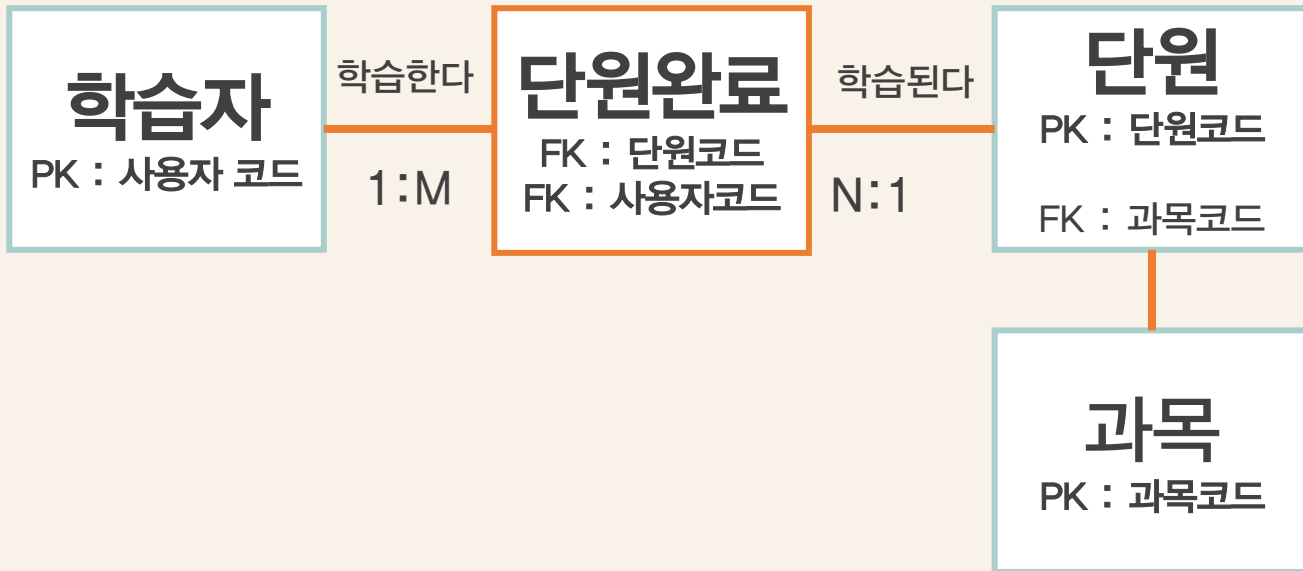
# 1. Algo



# ERD (이론학습)

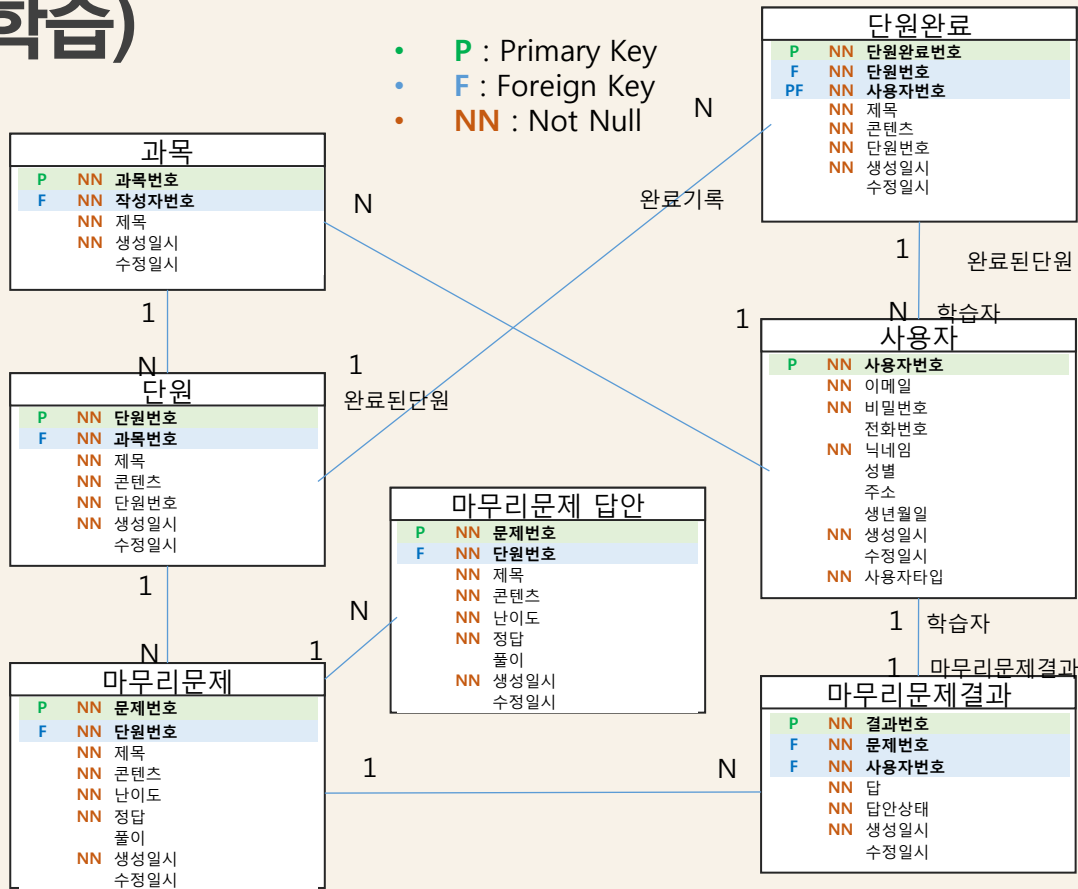


# ERD (이론학습)



# ERD (이론학습)

- **P** : Primary Key
- **F** : Foreign Key
- **NN** : Not Null



# HOW

## 아키텍처 설계

---

5.1 소요기술

5.2 시스템아키텍처

---

# 5

## 아키텍처 설계

### 5.1 소요기술

### 5.2 시스템아키텍처

# 소요기술

	기술	상세
배포 환경	Windows7	팀 역량상 가장 익숙한 환경
사용 언어	JAVA8	서버 로직 구현
	Java Script	프론트엔드 기능 구현 웹 표준
데이터베이스	Oracle 11g express	검증된 무료 소프트웨어
웹표준	HTML5/CSS3	표준에 맞게 페이지 구조와 디자인 분리
	SSE	서버의 변화를 추적하기 위함
	AJAX	XMLHttpRequest(web API) 사용
프레임워크	Spring Framework	MVC, DI, Rest 등 사용
3 <sup>rd</sup> Party Library	JNA	쉽게 기본 공유라이브러리에 접근 가능 Windows API(Win32) 접근 목적
	Lombok	Getter/Setter 및 equal 메소드 자동생성
	Mybatis	Sql매핑
	Code Mirror	코드 편집기, 코드 Assist

## 아키텍처 설계

### 5.1 소요기술

### 5.2 시스템아키텍처

## 소요기술(JNA : Java Native Access)

# 1

Java로 특정 프로세스 메모리 사용량 **측정**이 어려움

---

# 2

Window API 를 사용해야 함

---

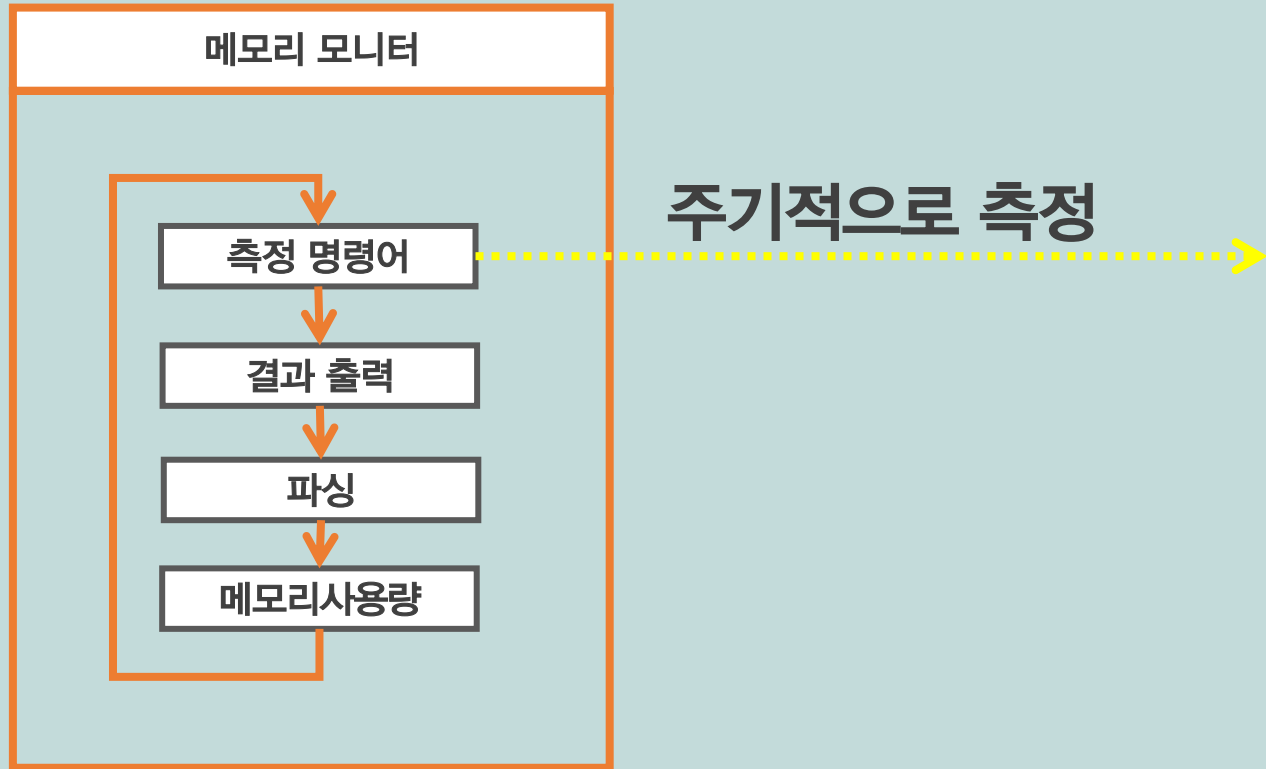


## 아키텍처 설계

### 5.1 소요기술

### 5.2 시스템아키텍처

## 소요기술(JNA : Java Native Access)

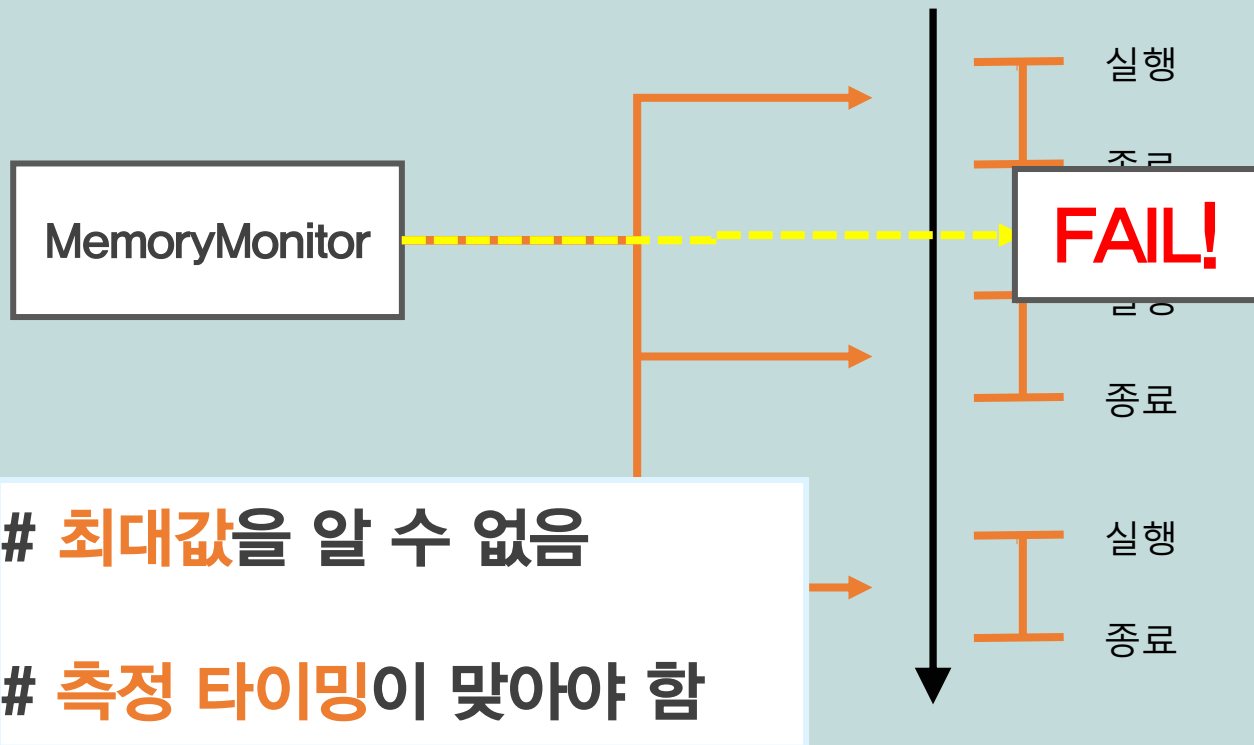


## 아키텍처 설계

### 5.1 소요기술

### 5.2 시스템아키텍처

## 소요기술(JNA : Java Native Access)



## 아키텍처 설계

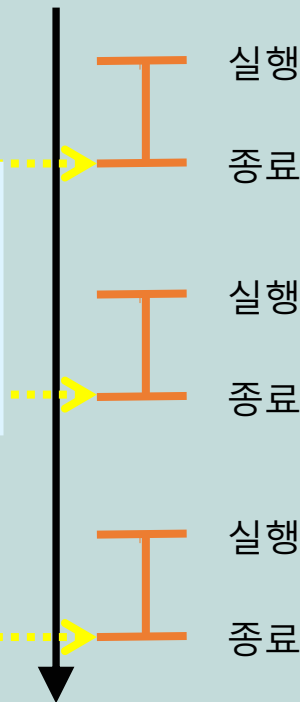
### 5.1 소요기술

### 5.2 시스템아키텍처

## 소요기술(JNA : Java Native Access)

# 최대값 측정 가능

# 종료 시점에도 확인 가능



# HOW

## 아키텍처 설계

---

5.1 소요기술

5.2 시스템아키텍처

---

# 5

## Presentation Layer

1. learningContentManaging.jsp: 과목 관리  
2. unitContentEditor.jsp : 단위 내용 관리  
3. challengeContentManaging.jsp: 도전과제 관리  
4. challengeContentEditor.jsp: 도전과제 내용 관리

1. SubjectList.jsp : 과목 및 단위 보기  
2. unitContent.jsp : 단위 내용 보기  
3. quiz.jsp : 마무리문제 풀기화면

1. challengeList : 도전과제 리스트/검색  
2. challengeSolving.jsp: 도전과제 참여  
3. ranking.jsp : 도전과제 랭킹

1. signup.jsp : 회원가입  
2. signin.jsp : 로그인  
3. profile.jsp : 회원정보 수정  
4. login.jsp : 로그인 화면  
5. tutorApplicationList.jsp : 튜터 신청 관리  
6. tutorApplication.jsp : 튜터신청양식

1. board.jsp : 게시판 메인  
2. posting.jsp : 게시물/작성/수정  
3. postView.jsp : 게시물 내용 열람  
4. Comment.jsp : 댓글 작성

## Controller Layer

## Dispatcher Servlet

## ChallengeController

## RankingController

## LearningController

## ContentController

## AdminController

## UserController

## LoginController

## BoardController

ChallengeResultVO

ChallengeCaseVO

ChallengeVO

QuizResultVO

SubjectVO

QuizVO

UnitCompletionVO

UnitVO

QuizAnswerVO

UserVO

TutorApplicationVO

BoardCategoryVO

BoardPostVO

CommentPreferenceVO

## Business Layer

## ChallengeService

## RankingService

## LearningService

## ContentManagingService

## UserService

## LoginService

## BoardService

## ChallengeManager

## RankingManager

## LearningManager

## ContentManager

## UserManager

## LoginManager

## BoardManager

CodeTester

JavaCodeTester

NativeAccess

PageManager

RowCriteria

ChallengeCaseVO

ChallengeVO

ChallengeResultVO

QuizResultVO

SubjectVO

QuizVO

QuizAnswerVO

UnitCompletionVO

UnitVO

UserVO

TutorApplicationVO

BoardCategoryVO

BoardPostVO

AttachmentVO

CommentVO

CommentPreferenceVO

## Data Access Layer

ChallengeDAO

ChallengeDAOImpl

ChallengeDAO-mapper.xml

LearningDAO

LearningDAOImpl

LearningDAO-mapper.xml

UserDAO

UserDAOImpl

UserDAO-mapper.xml

PostingDAO

PostingDAOImpl

PostingDAO-mapper.xml

CommentDAO

CommentDAOImpl

CommentDAO-mapper.xml

## SqlSession

## Database

UNIT\_COMPLETION

FNSH\_TEST

BOARD\_COMMENT

FNSH\_TEST\_RESULT

APP\_USER

CHALLENGE

BOARD\_POST

TUTOR\_APPLICATION

CHALLENGE\_CASE

CHALLENGE\_RESULT

SUBJECT

UNIT

## 5.2 시스템 아키텍처(도전과제)

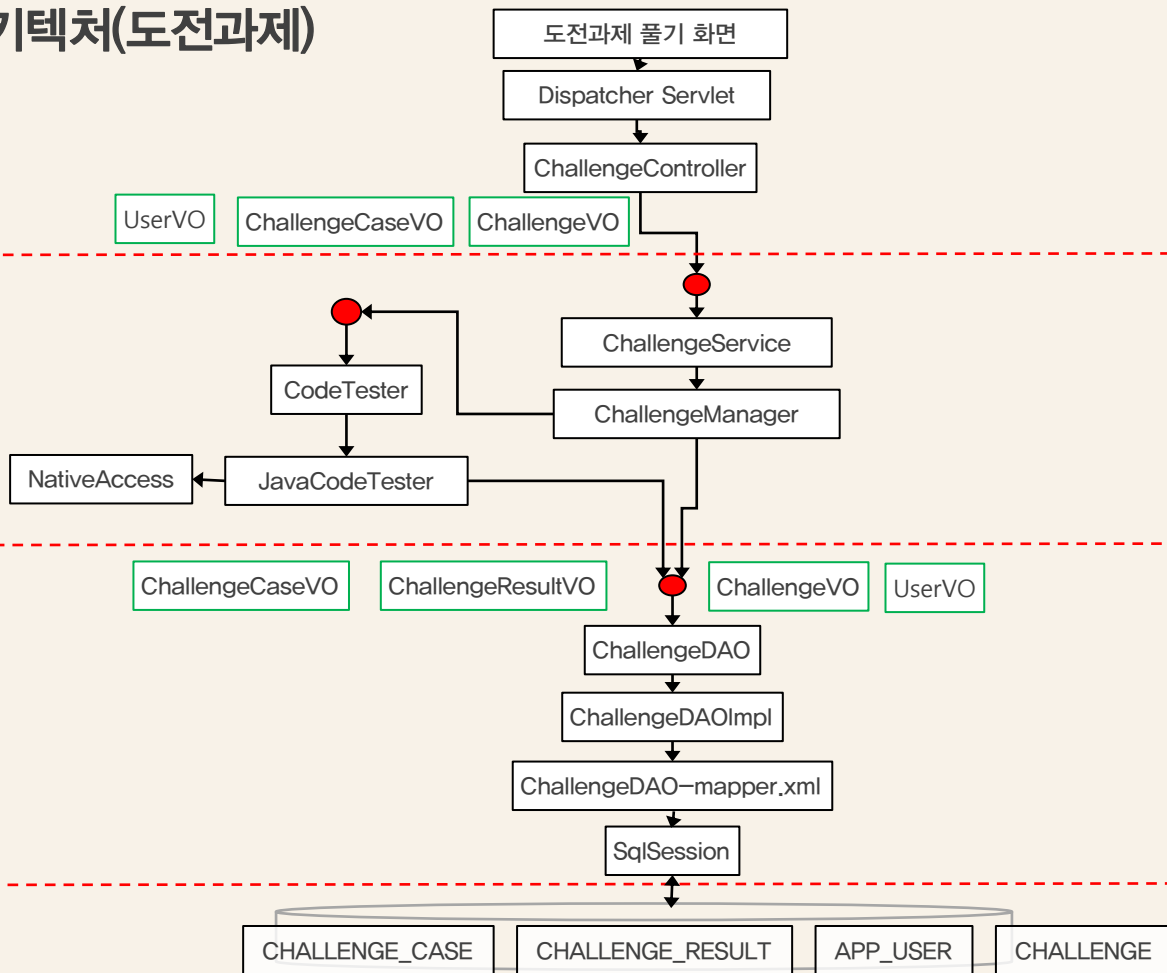
Presentation Layer

Controller Layer

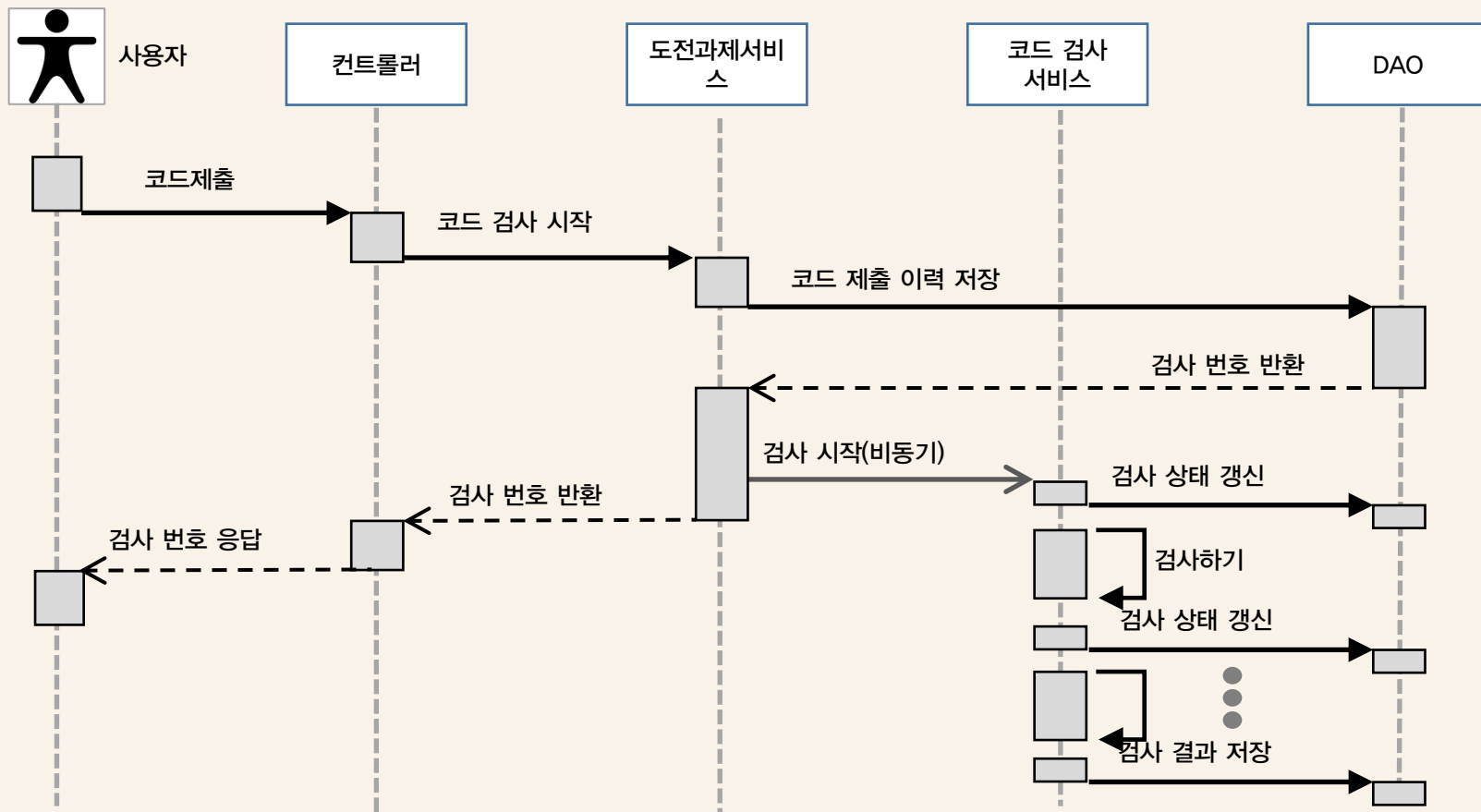
Business Layer

Data Access Layer

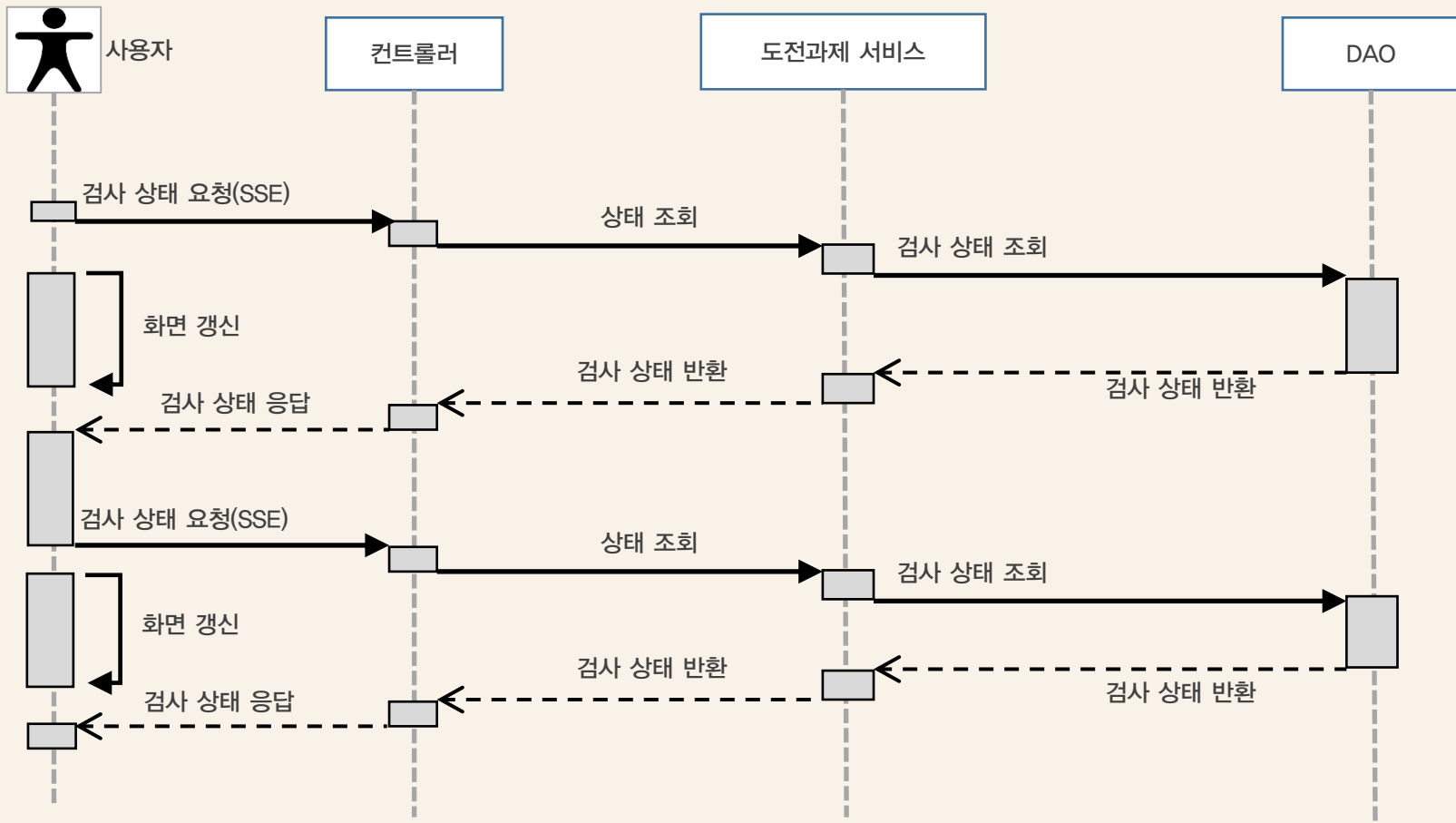
Database



## 시퀀스 다이어그램 : 코드검사



## 시퀀스 다이어그램: 실시간 갱신





## 5.2 시스템 아키텍처(이론학습)

Presentation Layer

Controller Layer

Business Layer

Data Access Layer

Database

1.SubjectList.jsp : 과목 및 단원 보기  
2.unitContent.jsp : 단원 내용 보기  
3.quiz.jsp : 마무리문제 풀기화면

Dispatcher Servlet

LearningController

QuizResultVO

SubjectVO

QuizVO

UnitVO

QuizAnswerVO

UnitCompletionVO

LearningService

LearningManager

QuizResultVO

QuizVO

SubjectVO

UnitCompletionVO

QuizAnswerVO

LearningDAO

LearningDAOImpl

LearningDAO-mapper.xml

SqlSession

UNIT\_COMPLETION

FNSH\_TEST

SUBJECT

FNSH\_TEST\_RESULT

APP\_USER

UNIT

# HOW

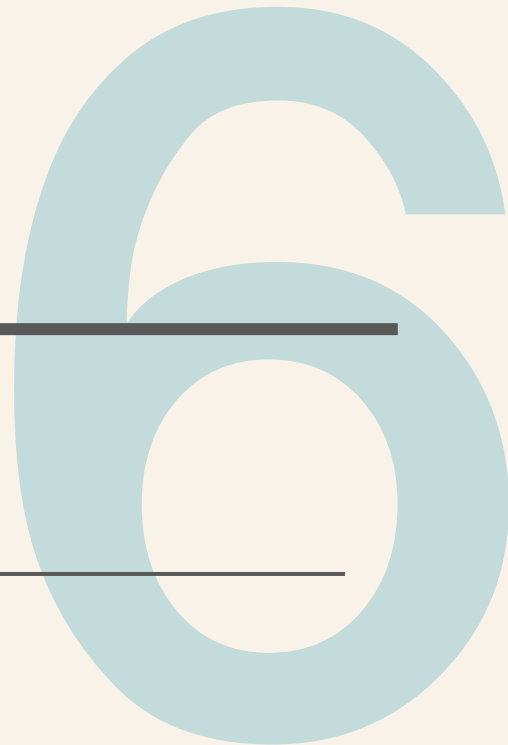
## 화면 설계

---

6.1 메뉴구조

6.2 주요화면 레이아웃

---

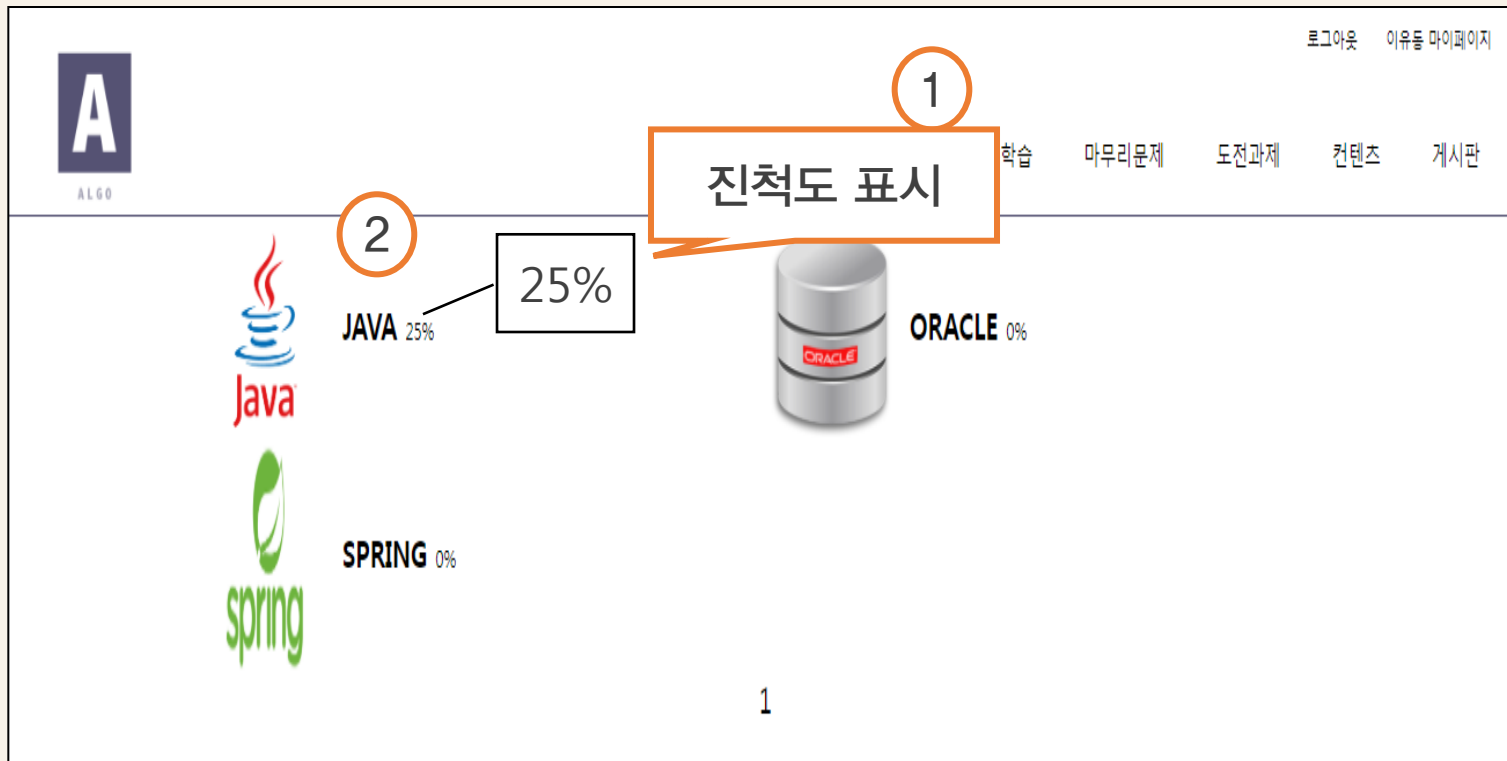


## 6.1 메뉴구조


이론학습	도전과제	컨텐츠 관리	회원관리	
과목 조회	도전과제 조회	이론학습 관리	회원관리	회원정보수정
-단원 내용 열람	-도전과제 풀기	-과목 관리 화면	-튜터신청서 관리	-튜터 신청하기
-단원 마무리문제	-과제 랭킹 보기	-단원내용 편집기		
		도전과제 관리		
		-도전과제 목록		
		-도전과제 편집화면		

메인화면	게시판	Algo 소개	헤더/푸터/네비게이션
메인화면	게시판	소개	로그인
-사용자 랭킹	-글쓰기	연혁	회원가입
-문제리스트	-글 수정		회원정보 수정
	-답글 쓰기		메뉴

## 6.2 주요화면 레이아웃 이론학습



## 6.2 주요화면 레이아웃 이론학습

ALGO

로그아웃   김현욱 마이페이지

ALGO소개   이론학습   마무리문제   도전과제   콘텐츠   게시판

1.JAVA-1 의 마무리문제 **1**

1

2

3

**마무리문제1** **2**

다음 문제를 풀어보세요.

1 보기값 넣어주세요 정답 ☐

2 보기값 넣어주세요 ☐

3 보기값 넣어주세요 ☐

4 보기값 넣어주세요 ☐

**3**

정답확인

## 6.2 주요화면 레이아웃 도전과제

### A와 B의 합

작성자 : test

정답자(도전자) : 3(4)

제한 시간: 1500 ms

메모리제한: 100000000000 byte

내용:

a와 b를 입력하세요

결과 표시

상태 : 성공(S)

메모리 사용량 : 248455168 byte

실행시간 : 1330 ns

랭킹보기

```
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         int a = sc.nextInt();
7         int b = sc.nextInt();
8         try {
9             Thread.sleep(1000);
10        } catch (InterruptedException e) {
11            // TODO Auto-generated catch block
12            e.printStackTrace();
13        }
14        System.out.println(a+b);
15    }
16 }
17 }
```

메모리 사용량 확인  
→ SSE 적용

제출

# HOW

## 시연

---

7.1 시연 시나리오

7.2 동영상 시연

---



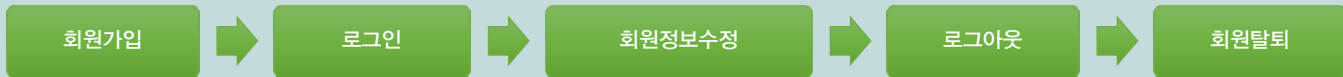
# 시연

## 7.1 시연시나리오

### 7.2 동영상시연

#### 사용자권한

##### 회원관리



##### 이론학습



##### 도전과제



##### 질문게시판



##### 튜터신청





## 시연

### 7.1 시연시나리오

#### 7.2 동영상시연

### 관리자 권한

#### 이론학습 관리



#### 도전과제 관리



#### 튜터 관리



## 시연

7.1 시연시나리오

7.2 동영상시연

<https://www.youtube.com/watch?v=60Q8J9zLT58>

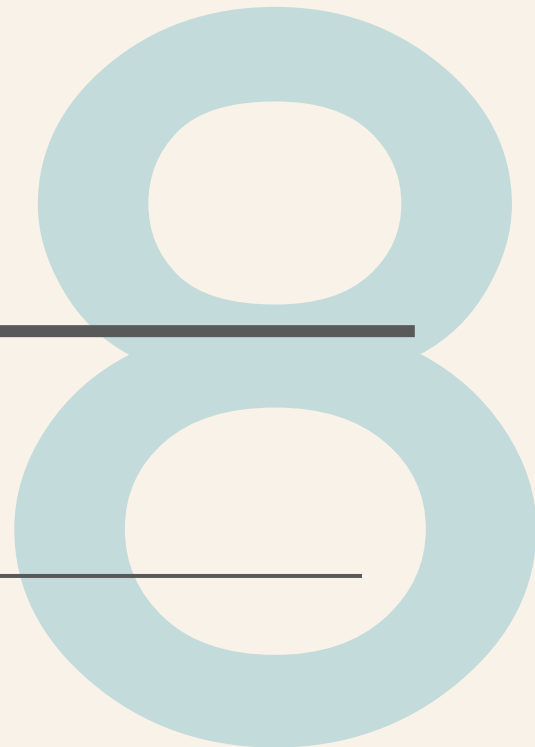
# Algo

## 후기

---

8. 후기

---



## Q&A, 후기

### 8. 후기

허은열

-Team 단위 프로세스를 경험할 수 있어서 좋았습니다

---

이유동

-부족한 부분을 서로 채워주는 것이 Team 이다.

---

김현욱

-Team 과의 협업에는 약속된 절차가 중요한 것을 느꼈습니다

---

박용민

-Team 덕분에 포기하지 않을 수 있어서 고마웠습니다

---



감사합니다