

석사학위논문

프로그래밍 교육을 위한 온라인 실습
환경 시스템 설계 및 구현

Design and Implementation of
Online Practice Environment System for
Programming Education

2021년 6월

배재대학교 대학원

컴퓨터공학과 컴퓨터공학전공

송 하 빈

프로그래밍 교육을 위한 온라인 실습 환경 설계 및 구현

지도교수 정 회 경

송하빈의 석사학위논문을 제출함.

배재대학교 대학원

컴퓨터공학과 컴퓨터공학전공

송 하 빈

2021년 6월

송하빈의 석사학위논문을 인준함.

심사위원장

김 창 수



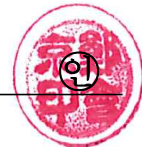
심사위원

이 성 옥



심사위원

정 회 경



2021년 6월

배재대학교 대학원

프로그래밍 교육을 위한 온라인 기반 실습 환경 설계 및 구현

송 하 빈

지도교수 정 회 경

배재대학교 대학원 컴퓨터공학과

최근 프로그래밍, 인공지능에 관한 관심이 높아지면서 초등학교에서부터 소프트웨어 의무 교육을 실시하고 있다. 프로그래밍 교육이란 문제를 컴퓨터로 해결할 수 있도록 문제해결 능력을 배양할 수 있는 교육이다. 교육부는 이에 따라 교육과정을 재구성하고 일부 선도학교를 운영 중이며 지역 대학과 초·중등학교 간 소프트웨어 교육에 대한 협력과 연계도 늘어나는 추세다.

이러한 프로그래밍 교육의 목표를 달성하기 위해 기본적으로 학생과 교사에게 맞는 실습실 환경을 구축해야 한다. 하지만 실습실 환경은 비용 문제로 인한 성능 문제가 있고 특히, 학생들이 동시에 접속해 문제해결 능력을 평가하고자 하는 알고리즘 경진대회 프로그램을 시행하면서 기존의 온라인에 출시된 플랫폼을 구매해 사용해야 하는 문제가 있다. 모두 경제적 비용이 매우 크게 들고 있는 실정이다.

따라서 본 논문에서는 이러한 문제를 해결하고자 웹 기반의 온라인 실습 환경과 알고리즘 경진대회 채점 시스템을 제안한다. 이 시스템은 React, Spring boot를 통해 구현하였다. 학생 관점에서 접속한 온라인 실습 환경과 교사 관점에서 학생들을 관리할 수 있으며 특히, 알고리즘 경진대회 진행 시 학생들이 접속해서

해결한 문제에 대한 소스 코드를 일괄적으로 볼 수 있도록 하고 이에 대한 채점을 진행할 수 있도록 한다. 이를 통해 기존 온라인 플랫폼을 이용하는 방법이 아닌 교내에서 간단히 이를 구축하여 비용을 줄이고 교육을 원활히 진행할 수 있을 것으로 사료된다.

주요어 : Software Education, Online Programming, Online Monitoring System, Spring boot, React

목 차

국문초록	i
목 차	iii
그림목차	v
도표목차	vi
I. 서 론	1
1.1 연구배경 및 목적	1
1.2 연구내용 및 범위	2
1.3 논문의 구성	3
II. 관련 연구	4
2.1 컴퓨팅 기반 사고	4
2.2 한국 소프트웨어 교육 현황	6
2.3 해외 소프트웨어 교육 사례	7
2.4 학교 실습 환경	9
2.5 온라인 프로그래밍 플랫폼 사례	11
2.6 React와 Spring boot	12
III. 시스템 설계	14
3.1 요구사항 분석	14
3.2 데이터베이스 설계	17
3.3 시스템 구조	19
IV. 시스템 구현	21
4.1 개발 환경	21
4.2 모듈 구현	22

4.2.1 회원 인증 모듈	22
4.2.2 UI 컴포넌트 구성	25
4.2.3 문제 관리와 소스 코드 관리 모듈	26
4.3 구현 결과	28
 V. 결론 및 향후연구	 30
 참고문헌	 31
영문초록	35
감사의 글(Acknowledgement)	37

그 립 목 차

그림 1. Usecase diagram	14
그림 2. Activity diagram	15
그림 3. Activity diagram for student	16
그림 4. Table schema for saving problems and source code	17
그림 5. Table schema for saving membership and authentication	18
그림 6. The example of data in the ROLES table	19
그림 7. System architecture	19
그림 8. System module configuration diagram	20
그림 9. Sequence diagram for membership authentication module	23
그림 10. Configuration diagram of member authentication module	24
그림 11. React-based UI Components and communication module configuration diagram	26
그림 12. Problem creating page	27
그림 13. Source code editing page	28
그림 14. Problem list page	29
그림 15. Source code output page for each student	29

표 목 차

표 1. Computational thinking component	5
표 2. Content system (2015 Education revision supplement 10)	7
표 3. The example of SW competency training in major countires around the world	8
표 4. Overseas and domestic SW training hours	9
표 5. Elementary and middle school computer rooms and expansion plans	10
표 6. Status of old PCs for elementary, middel and high school students in 2019	10
표 7. Development environment	21

I. 서 론

1.1 연구배경 및 목적

교육부는 초등학교와 중학교 전체를 대상으로 소프트웨어 교육을 필수화하고 있으며 이에 따라 교육과정을 재구성하고 이를 성공적으로 안착하기 위해 SW선도학교를 운영 중이다[1]. 이 외에 해외 사례를 살펴보면 미국의 경우 2016년 오바마 대통령의 연설로 소프트웨어 교육 정책을 추진하겠다고 발표하였으며 영국은 2014년 학생들을 대상으로 필수로 소프트웨어 교육을 하고 있다[2, 3].

과거에 소프트웨어 교육은 한글, Microsoft PowerPoint, Excel과 같은 워드프로세서, 프레젠테이션 툴 등과 같은 오피스 프로그램이나 Adobe Photoshop, Illustrator, Premiere와 같은 그래픽, 영상 편집 프로그램과 같은 상용 프로그램을 얼마나 잘 활용하여 주어진 과제나 학생이 처한 문제에 대해 올바른 결과물을 출력하는 것에 더욱 치중되어 있었다.

하지만 최근 소프트웨어 교육은 프로그래밍 능력에 초점을 두고 있으며 MIT Media Lab에서 개발한 Scratch와 같은 그래픽 환경에서 프로그래밍 교육에 활용할 수 있는 플랫폼을 활용하고 있고, 머신러닝(Machine Learning), 이미지 인식(Image Recognition) 등 다양한 분야에서 활용하고 있는 Python 라이브러리가 활용되고 있다[4].

이러한 소프트웨어 교육은 다양한 패러다임과 개념이 제시되고 있으며 특히 “컴퓨터 기반 사고”라는 개념이 주목받게 되었다. 소프트웨어 교육은 소프트웨어 개발자를 양성하는 것이 목표가 아니라 “컴퓨터 기반 사고”에 따라 컴퓨터를 이용해 학생 자신이 직면한 문제를 해결하기 위해 알고리즘을 설계하고 이를 프로그래밍하는 것이라고 할 수 있다[5].

이러한 변화에 비춰 보면 학교에서 진행하는 소프트웨어 교육이 주어진 문제를 해결하기 위해 창의적으로 알고리즘을 설계하고 이를 프로그래밍 언어로 구현하는 데 초점을 두고 있다고 볼 수 있다.

최근에는 대학에서 지역 내 중고등학생, 본교 전공자와 비전공자를 대상으로 다양한 프로그램을 수행하여 소프트웨어의 가치를 확산시키고 있다. 특히 전공자를 대상으로 한 알고리즘 경진대회와 중고등학교 컴퓨터 실습실에서 다양한 프로그래밍 교육을 수행하고 있어 교육 환경에 대한 개선이 필요한 실정이다. 학교마다 실습실 컴퓨터 환경이 다르고 교육 프로그램과 프로그래밍 언어도 다르므로 이에 맞는 환경을 구축하기란 쉽지 않다. 따라서, 이러한 문제를 해결하고자 학생과 강사가 실습실 컴퓨터 환경에 구애받지 않고 웹 브라우저만 설치되어 있으면 이용할 수 있는 시스템을 제안하고자 한다.

1.2 연구내용 및 범위

본 논문에서는 소프트웨어 교육과 관련한 우리나라와 해외에서 교육과정을 살펴보고 현재 소프트웨어 교육의 현황과 실습 환경에 대해 논할 것이다. 그리고 이러한 대응 방안 중 하나로 온라인 실습 환경을 제안하며 이를 설계하고 구현한 결과에 대해 논의할 것이다. 연구내용과 범위는 아래와 같다.

● 국내외 소프트웨어 교육 현황

국내외에서 진행되고 있는 소프트웨어 교육 현황에 대해 분석하고 이러한 교육을 수행하기 위해 필요한 자원과 문제점을 제시한다.

- 시스템 구현을 위한 프레임워크

본 논문에서 제안하는 전체 시스템을 설계 및 구현하기 위하여 필요한 웹 서버 프레임워크와 클라이언트에서 UI를 담당하는 프레임워크에 대해서술한다.

- 요구사항 분석을 통한 시스템 설계

제안하는 시스템을 구현하기 전에 수행한 사용자 요구사항 분석을 논하고 이에 대한 서버 모듈과 데이터베이스 설계를 통해 실제 구현 결과를 서술한다.

1.3 논문의 구성

제2장에서는 소프트웨어 교육에 대한 전반적인 내용을 기술하고, 우리나라와 해외 사례를 간략히 분석하여 구현에 필요한 기술들에 대해서 간략히 기술한다. 제3장에서는 소프트웨어 교육에 활용되는 사용자 요구사항을 바탕으로 한 웹 기반 실습 환경을 설계한다. 제4장에서는 구현을 통해 구현 결과를 보인다. 마지막으로 제5장에서는 결론 및 향후 연구내용을 제시한다.

II. 관련연구

세계 주요나라에서는 초중고등학교 소프트웨어 교육에 대한 중요성을 인식하여, 정규 교육과정에 컴퓨팅 사고력 능력을 갖추기 위한 교육 시스템을 도입하고 있다. 우리나라의 경우 IT 교육을 조기에 도입하여 IT 강국으로 크게 성장하는데 기여 하였지만, 그동안의 SW 교육 과정이 IT에 대한 활용 위주로 구성되어 있었으므로, 컴퓨팅 사고력에 기반한 소프트웨어 교육은 부족하였다[6]. 이를 극복하기 위해 교육부는 2015 개정 교육과정을 통해 컴퓨팅 사고력을 갖춘 인재 육성을 목표로 하였다.

2.1 컴퓨팅 기반 사고

컴퓨팅 사고력(Computing Thinking)에 대한 정의에 대해 국내외 여러 학자들에 의해 다양하게 정의되고 있으나, 1980년 Seymour Papert에 의해 처음 소개되었고, 그 후 2006년 카네기 멜런 대학교의 교수였던 Wing이 재정립하였다[7].

Wing의 표현에 의하면 컴퓨팅 사고력이란 프로그래밍을 뜻하는 것이 아니라 개념화에 관한 것이며, 근본적인 기술이라고 표현하고 있다. 또한 수학적 사고와 공학적 사고를 보완하고 결합하며, 누구에게 어디에서나 필요하다고 정의하고 있다. Wing은 많은 사람들이 컴퓨터 과학을 컴퓨터 프로그래밍과 같은 개념으로 오해하고 있고 컴퓨터 과학을 공부하게 되면 선택할 수 있는 직종이 한정되어 있다고 생각하지만 이보다는 21세기를 살아가는 모든 사람이 컴퓨팅적 방법론과 모델을 교육적 측면에서 가르쳐야 한다고 주장하고 있다[8].

CSTA & ISTE(2011)에서는 컴퓨팅 사고력의 문제 해결을 위한 요소들로 자료 수집, 자료 분석, 자료 표현, 문제 분해, 추상화, 알고리즘, 자동화, 시뮬레이션, 병렬화 9가지로 아래의 표 1과 같이 구분하였다. 이 9가지 요소들은 우선순위가 존재하지 않으며, 항상 모두 포함되는 것은 아니라고 설명하고 있다[9].

표 1. Computational thinking component

CT	Definition
Data Collection	The process of gathering appropriate information.
Data Analysis	Making sense of data, finding patterns, and drawing conclusions.
Data Representation	Depicting and organizing data in appropriate graphs, charts, words, or images.
Problem Decomposition	Breaking down tasks into smaller, manageable parts.
Abstraction	Reducing complexity to define main idea.
Algorithms& Procedures	Series of ordered steps taken to solve a problem or achieve some end.
Automation	Having computers or machines do repetitive or tedious tasks.
Simulation	Representation or model of a process. Simulation also involves running experiments using models.
Parallelization	Organize resources to simultaneously carry out tasks to reach a common goal.

2.2 한국 소프트웨어 교육 현황

교육부에서 2019년 7월에 발표한 미래세대 과학교육표준 자료에 따르면, 과학적 소양을 3개의 차원으로 제시하였다. 그 중 역량에 해당하는 영역 중 과학적 사고력 영역에 컴퓨팅 사고력을 새롭게 포함시켜 6개 단계로 학년별로 나뉘었다. 5~8학년에 해당하는 3, 4 단계에선 문제 해결에 컴퓨터를 활용하고 수리적 규칙성을 이해하는 것을 제시하였다. 9~12학년에 해당하는 5, 6단계에선 컴퓨터를 활용하여 복잡한 문제의 규칙을 찾거나 복잡한 자연현상을 알고리즘으로 나타내는 수행기대(performance expectation)를 제시하였다[10].

2015년 개정 교육과정에선 소프트웨어 교육 강화를 위해 기존 초등학교 교과(실과)에 ICT를 활용한 교육 단원이 포함되었지만, 이를 SW 기초 소양교육으로 개편하였으며, 중학교 교육과정에서 선택 교과였던 ‘정보’ 교과목을 ‘가정/정보’ 교과목으로 신설하였다.

또한, 고등학교 심화선택 과목이었던 ‘정보’ 과목을 일반선택으로 전환하였고, SW 중심으로 개편하였다. 이러한 변경 사항들은 2020년 3월 1일부로 초·중·고 전체 교육과정에 반영되었다. 또한 교육부에서 발표한 정보 과목의 내용 체계에 포함되는 ‘문제 해결과 프로그래밍’ 영역은 표 2와 같다. 표 2는 2015년 교육개정안 별책 10의 일부이다[11]. 해당 영역에는 ‘추상화’, ‘알고리즘’, ‘프로그래밍’ 등의 핵심 개념이 포함되어 있다.

표 2. Content system (2015 Education revision supplement 10)

영역	핵심 개념	일반화된 지식	내용 요소	기능
문제 해결과 프로그래밍	추상화	추상화는 문제를 이해하고 분석하여 문제 해결을 위해 불필요한 요소를 제거하거나 작은 문제로 나누는 과정	문제 이해 핵심요소 추출	<ul style="list-style-type: none"> - 비교하기 - 분석하기 - 핵심요소 추출하기 - 표현하기 - 프로그래밍하기 - 구현하기 - 협력하기
	알고리즘	알고리즘은 문제 해결을 위한 효율적인 방법과 절차	알고리즘 이해 알고리즘 표현	
	프로그래밍	프로그래밍은 문제의 해결책을 프로그래밍 언어로 구현하여 자동화하는 과정	입력과 출력 변수와 연산 제어 구조 프로그래밍 운용	

SW 교육을 자리잡기 위해 2015년부터 지정하여 운영한 SW 교육 선도학교는 꾸준히 늘어나고 있는데, 2019년 기준 1,834개교, 2020년엔 2,011개교로 운영되고 있다[1].

2.3 해외 사례

세계 주요 나라들에서는 4차 산업혁명을 선도하기 위해 SW 인재 양성을 위한 교육을 진행하고 있다. 아래의 표 3과 같이 여러 나라들이 활발히 SW 교육을 정규 교과목에 포함하고 있다[12].

표 3. The example of SW competency training in major countries around the world

국가	주요 내용	국가	주요 내용
미국	16년 1월부터 컴퓨터 과학 교육을 위한 ‘Computer Science for All’ 추진	프랑스	16년 9월부터 SW를 중학교 정규 과목화
영국	14.9월부터 초·중등학교 정규 교과과정에 SW 필수교육 포함	이스라엘	94년부터 SW과목을 정규과목에 포함

미국의 경우 ‘Computer Science for All’ 은 유치원부터 고등학교까지 모든 학생들이 컴퓨터 과학을 배우고 소비자의 역할만 하는 것이 아닌 창조까지 할 수 있는 역량을 키우기 위한 운동을 의미한다. 영국은 2014년 소프트웨어 활용 능력 강화에 집중되어 있던 ‘활용’ 교육에서 소프트웨어를 직접 만드는 방법인 코딩 교육으로 전환되었다. 이는 G20에 해당하는 국가 중 최초로 초·중등 학생들에게 프로그래밍 및 코딩 교육을 필수 교육과정으로 의무화한 사례이다[13].

반면 프랑스는 1970년대 초반 ‘정보학’을 통하여 프로그래밍 언어를 가르치는 교육이 있었지만 교육 커리큘럼이 개정되며 2012년 정보학은 ‘수학’ 교과목의 일환으로 교육되었다. 이후 2016년부터 중학교 교육과정에 ‘컴퓨터과학 및 프로그래밍(ISN)’ 과목을 포함하였다[14, 15]. 또한 미래 IT 인재를 육성하기 위해 ‘에꼴42’와 같은 IT 학교도 설립되어 운영하고 있다. 이 학교는 강사도 없고, 학비도 필요 없는 코딩 교육을 통해 미래 인재 육성하기 위한 것으로 최근 우리나라에서도 에꼴42의 교육과정을 수입해 ‘42 서울’이라는 SW 교육 프로그램이 도입되었다.

아래의 표 4는 국내 및 해외의 SW 교육 시간이다. 해외국가들의 경우 국내 교육보다 더 많은 시간을 SW 교육에 투자하고 있는 것을 볼 수 있다[12]. 본문에서 소개한 국가들 외에도 중국, 일본, 인도, 덴마크, 에스토니아 등 세계 각국에서 SW 교육에 많은 투자를 진행하고 있다.

표 4. Overseas and domestic SW training hours

구분	대상	시간		
		초	중	고
영국	1~11학년 전학년 필수	180	90	60
이스라엘	고등학교 이과 필수	-	-	270
인도	1~12학년 의무 교육단계 필수	180	180	180
중국	초등3학년 - 중학교 필수	70	70	72
대한민국	초등5~6학년 - 중학교 필수	17	34	68 (선택)

2.4 학교 실습 환경

코딩이나 프로그래밍을 통한 소프트웨어 교육이 원활하게 진행되기 위해서는 교육 수준에 맞는 기본적인 사양을 갖춘 PC와 네트워크 환경 구축이 필수적이다.

2018년 8월 31일 기준 전국 초·중고등학교의 컴퓨터실 확보 현황은 표 5와 같다. 초·중·고등학교 모두 확보율이 99% 이상이므로 대부분의 학교는 컴퓨터실을 보유하고 있는 것으로 확인된다. 컴퓨터실의 유무를 판단하는 기준은 학년 당 1학급 이하의 소규모 학교는 1실당 20대를 기준으로 하였고, 2학급 이상인 학교는 1실당 40대를 기준으로 하였다 [16].

표 5. Elementary and middle school computer rooms and expansion plans

학교급	학교 수	컴퓨터 실 미확보 교	컴퓨터실 신규 설치 예정교				대체시설, 기기활용 예정교	확보율 (%)
			2018 9-12월	2019	2020	합계		
초등학교	6,180	60	4	7	5	16	44	99.3
중학교	3,224	34	7	2	0	9	25	99.2
계	9,404	94	11	9	5	25	69	99.3

그러나 노후 컴퓨터, 인터넷망 구축 미비로 인하여 현장 소프트웨어 교육에 어려움은 아직 남아있다. 교육부는 이러한 노후 컴퓨터(연수 6년 초과)에 대해 조사를 실시하고 연차별 계획을 수립하여 교체를 추진할 계획이다. 표 6은 2019년 기준 5년 이상의 PC를 정리한 자료이다[16].

표 6. Status of old PCs for elementary, middle and high school students in 2019

구분	학교수	구입시기별 PC 현황				
		5~6년 이내		6년 초과		5년 초과 비중(%)
		대수	비중%	대수	비중%	
전체	11,906	103,208	10.3	182,681	18.2	28.5
초등학교	6,087	40,947	9.0	68,329	15.0	24.0
중학교	3,214	26,036	12.4	32,664	15.5	27.9
고등학교	2,356	34,605	10.7	79,013	24.4	35.1

2.5 온라인 프로그래밍 플랫폼 사례

온라인 프로그래밍 플랫폼은 대부분 알고리즘 코딩 테스트 형태의 사이트로 국·내외 존재하고 있다. 국내에 존재하는 사이트들은 프로그래머스, 구름, 그리고 백준 온라인 등의 사이트가 존재한다[17-19]. 해외 사이트의 경우에는 Codewars, Codility, 그리고 leetcode 등이 존재한다[20-22].

위에서 언급한 플랫폼들은 공통적으로 알고리즘 코딩 문제를 인터넷 브라우저를 통해 코딩하고 사용자가 제출한 문제의 정답 여부를 알 수 있는 기능이 있다. 알고리즘 공부를 하기 위해 좋은 사이트들이지만, 기관에서 사용하기에는 많은 비용이 들어가며, 실시간으로 강의자와 학생 사이에서의 의사소통이 힘든 부분이 있기 존재하므로 학교에서 교육을 위한 웹 프로그래밍 환경으로 사용하기에 적합하지 않다.

최근에는 머신러닝과 딥러닝 등 AI 관련 분야에서 Python이 많이 사용되는데, Google의 Colab, Kaggle, Microsoft의 Azure Notebooks, Amazon의 Sagemaker 등 Jupyter Notebook 기반의 온라인 플랫폼이 존재한다[23-26].

위와 같은 플랫폼들의 등장으로 사용자가 직접 Python이나 다양한 종속성 패키지들을 설치하지 않아도 웹 브라우저만 있으면 어디서든 손쉽게 개발할 수 있는 장점이 있다. 또한 AI 관련 분야에선 모델을 학습시키기 위해 GPU가 중요하게 작용하는데, 사용자의 컴퓨터의 사양에 관계없이 플랫폼에서 제공하는 GPU를 사용할 수 있는 장점이 있다.

하지만 이러한 플랫폼들의 경우에는 AI 관련 분야에 맞춰져 서비스를 제공하고 있으며, 제공하는 언어의 개수가 한정적으로 여러 가지 프로그래밍 언어를 교육하기엔 적합하지 않다.

2.6 React와 Spring boot

시스템을 설계하고 구현하기 위해 React 프레임워크와 Spring boot 프레임워크를 사용하였다. React 프레임워크는 프론트엔드(Front-end)를 담당하는 Facebook에서 제작한 컴포넌트 기반 프레임워크(component-based framework)이다. Spring 프레임워크는 백엔드(Back-end)를 담당하는 WAS(Web Application Server)로 기존의 Spring 프레임워크를 모듈화하여 경량화된 서버를 제작하기 위해 나온 것이다.

이러한 두 프레임워크를 연결하기 위해 Axios 모듈이 사용된다[27]. Axios 모듈은 JavaScript 문법(ES6 이후 지원) 중 Promise 기반의 HTTP 통신 모듈로서 Node.js나 프론트엔드 모듈로써 사용된다[28]. Promise는 JavaScript 비동기(asynchorous) 처리에 사용되는 객체를 구현할 수 있는 방식으로 대기(pending), 이행(fulfilled), 실패(rejected)의 3가지 상태를 지닌다.

이행 상태는 “if p. then (f. r)” 에서 함수 f가 즉시 큐(queue)에 들어갈 수 있는 상태이고, 실패 상태는 함수 r이 즉시 큐에 들어갈 수 있는 상태를 말한다. 대기 상태는 위 두 가지 상태가 모두 아닌 상태를 말한다. 즉, 어떤 비동기 이벤트 p가 들어오기를 기다리는 상태가 대기 상태이고 이를 처리하는 것이 이행 상태이며 예외나 오류가 발생한 상태를 실패 상태라고 한다. 이러한 기술을 이용해 React와 Spring boot간 JSON 데이터 통신이 이뤄질 수 있다.

로그인, 회원가입 부분은 Spring boot의 Security 모듈(module)을 사용하여 JWT(JSON Web Token) 인증 방식을 사용했다. 토큰 기반 인증 시스템의 구현체로 웹 표준 문서 RFC 7519에 정의되어 있다[29]. 이는 두 개체에서 JSON 객체를 안전하게 전달하기 위한 방식 중 하나로

회원 인증 방식에 많이 이용된다. JWT 토큰은 Security 모듈이 자동으로 인코딩(encoding)과 해싱(hashing) 작업을 한다. 헤더(header)와 내용(payload), 서명(signature)으로 구성되며 해싱 알고리즘은 보통 SHA256, SHA512 등을 사용한다.

III. 시스템 설계

3.1 요구사항 분석

본 논문에서 제안하는 시스템은 강사와 학생이 주로 이용하는 시스템으로 사용자 관점에서 요구사항 분석을 진행하였다. 또한, 시스템을 관리하는 관리자도 함께 요구사항을 분석하였다. 그림 1은 유스케이스 다이어그램(usecase diagram)을 나타낸 것이다.

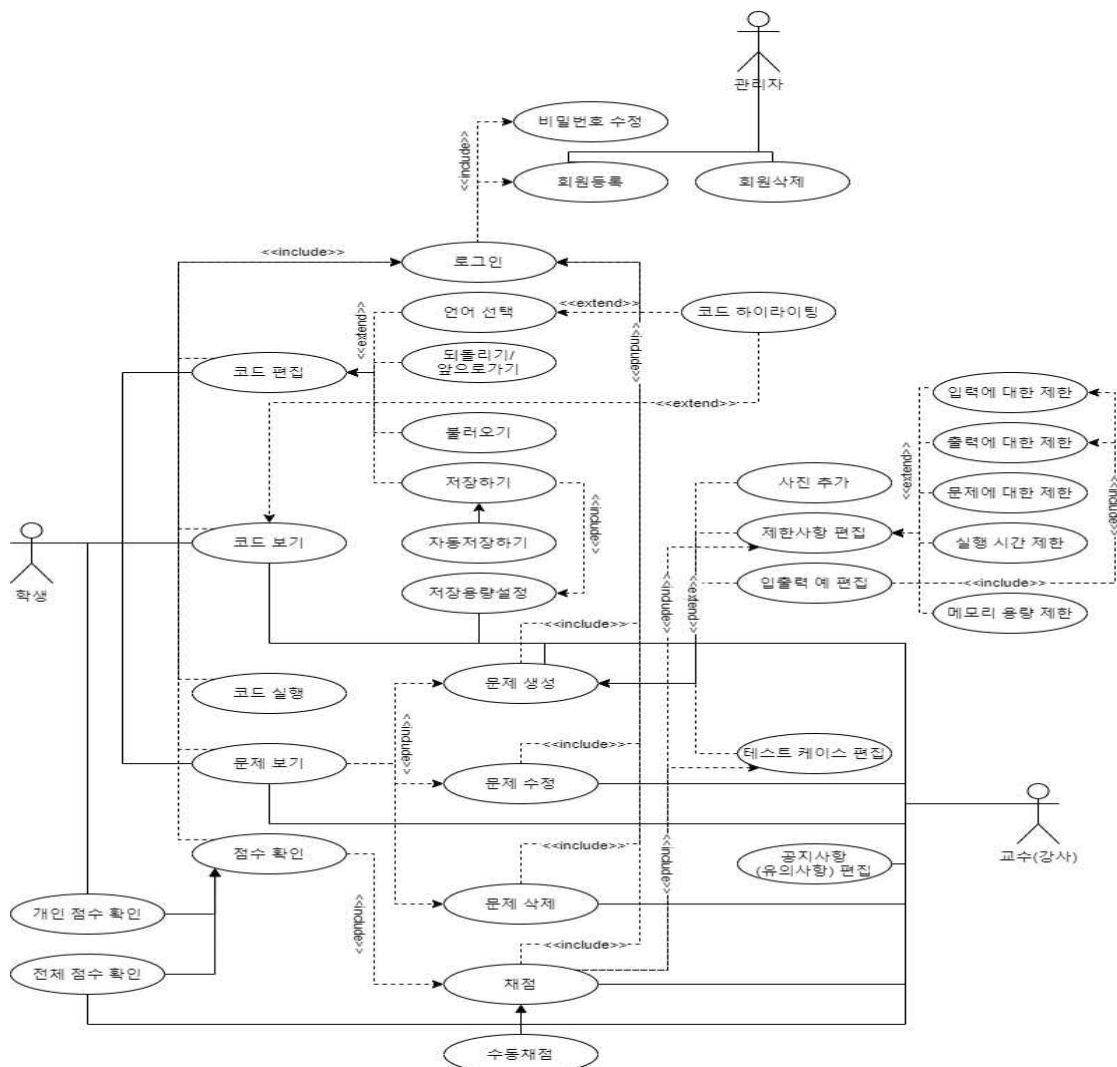


그림 4. Usecase diagram

그림 1에서 학생은 작성한 코드를 입력할 수 있도록 편집 기능과 이를 실행할 수 있도록 해야 한다. 만약 알고리즘 경진대회에서 이를 이용한다면 작성한 코드에 대해 강사가 채점한 점수도 함께 볼 수 있어야 한다. 반면 강사는 학생이 작성한 코드를 통해 채점해야 하고 경진대회를 위해 출제한 문제를 저장하는 기능도 있어야 한다.

코드를 편집하는 기능에서는 기본적인 되돌리기, 앞으로가기, 저장하기 등과 같은 편집기 기능이 필요하며 강사가 문제를 저장하기 위해 이미지를 업로드하고 글자 디자인 편집 기능과 다양한 제약조건을 입력해야 하므로 위지윅(WYSIWIG) 편집기 기능이 필요하다.

위에서 언급한 요구사항을 정리하여 강사 입장에서 서술한 액티비티 다이어그램(activity diagram)은 그림 2와 같으며, 학생 입장에서 서술한 액티비티 다이어그램은 그림 3과 같다.

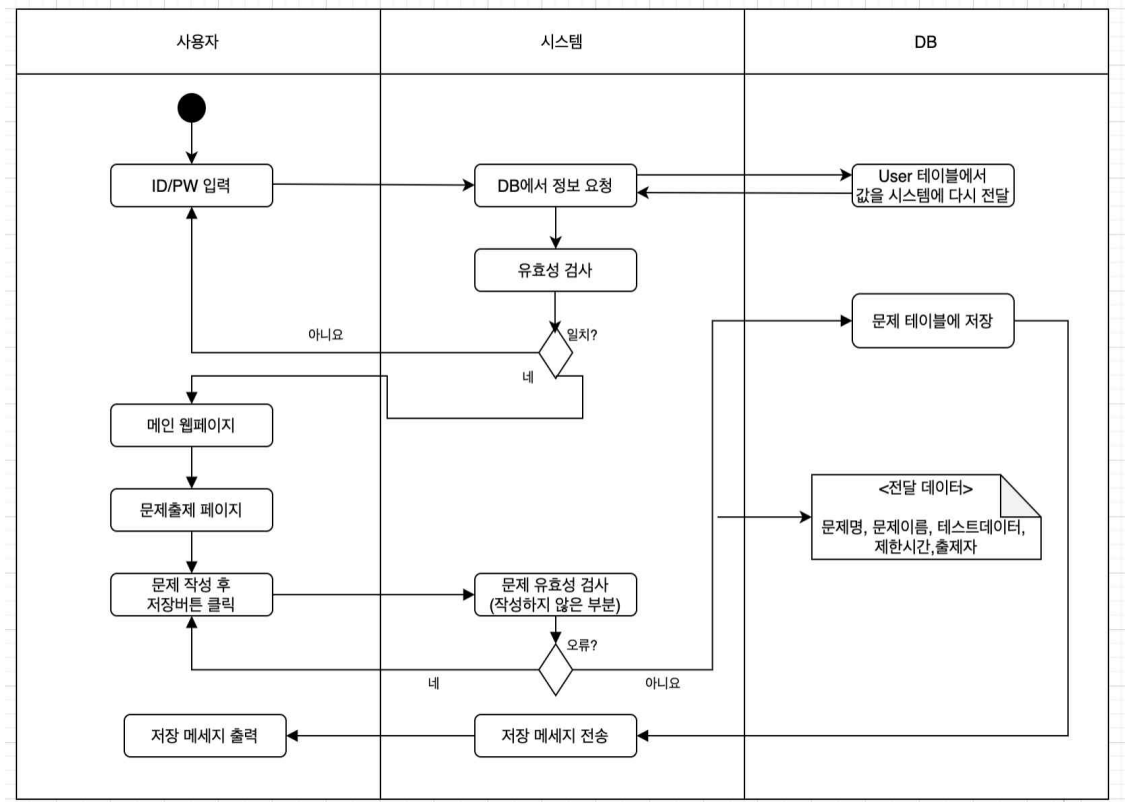


그림 5. Activity diagram for lecturer

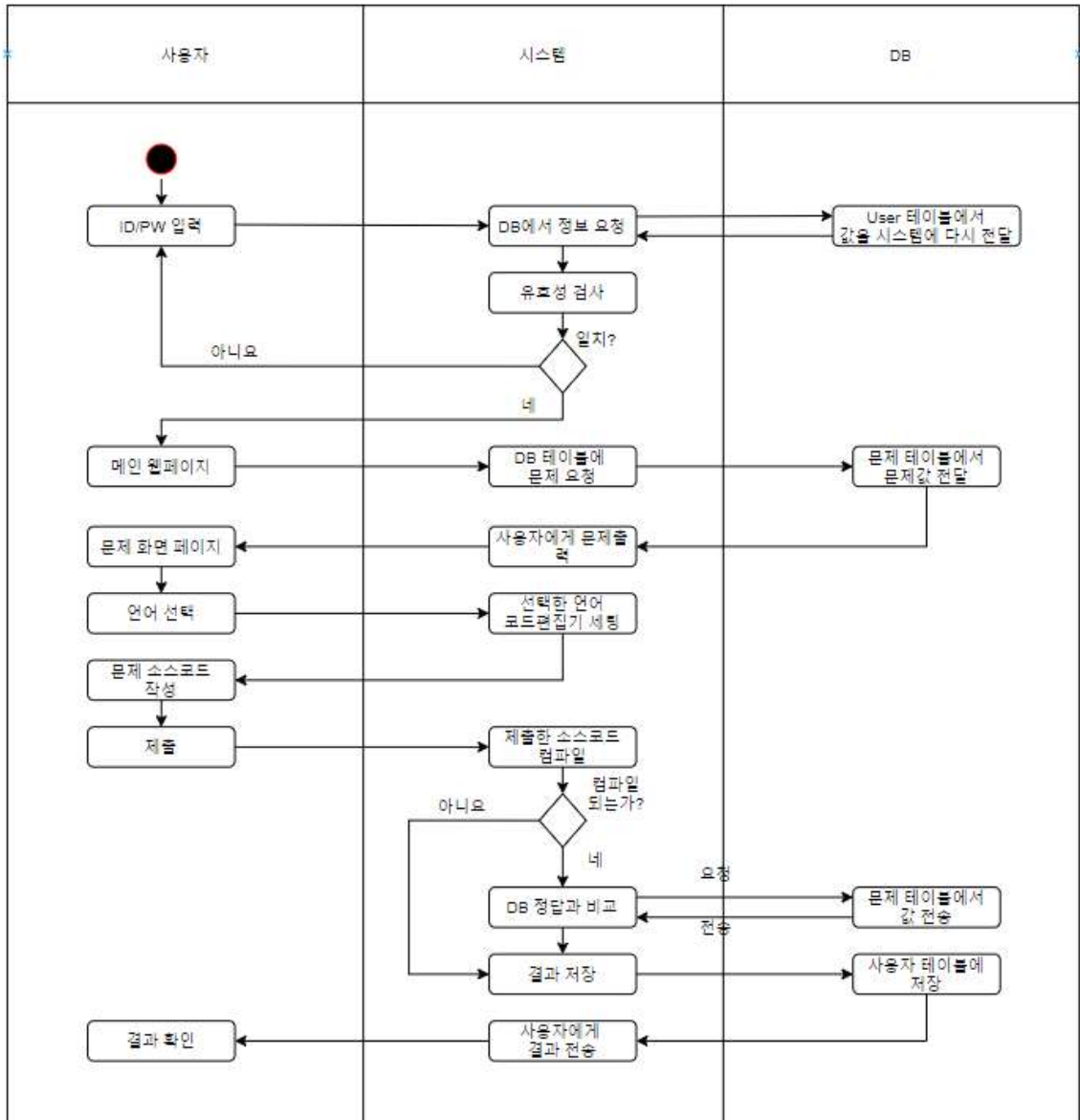


그림 6. Activity diagram for student

강사에 대한 액티비티 다이어그램에서 보는 바와 같이 강사 계정으로 로그인을 수행하면 문제 출제 페이지를 접속할 수 있는 권한을 준다. 강사가 문제 출제 페이지에서 문제를 업로드하면 작성하지 않은 부분에 대한 유효성 검사를 수행하고 최종 저장이 된다. 문제에 대한 데이터는 문제 이름, 테스트 데이터, 문제 내용, 제한 시간 등으로 구성된다.

한편 학생에 대한 액티비티 다이어그램은 학생 계정으로 로그인 후 수행되고 이후에 대한 흐름이다. 학생은 메인 웹페이지에서 강사가 업로드한 문제 리스트가 출력되고 한 문제를 선택하면 그 문제에 대한 소스코드를 작성할 수 있는 페이지로 가게 된다. 소스코드를 작성하기 전에 프로그래밍 언어를 선택하고 모두 작성이 완료됐으면 저장을 클릭하면 소스코드가 서버에 파일 형태로 저장된다. 이 파일을 채점 서버에 보내 출력 결과를 정답과 비교하고 이러한 결과를 저장하게 된다.

3.2 데이터베이스 설계

경진대회를 위해 강사가 저장한 문제를 저장하기 위한 PROBLEMS 테이블 스키마(schema)와 학생이 작성한 소스 코드를 저장하기 위한 SOURCE 테이블 스키마는 그림 4와 같다.

```
MariaDB [oswtf_db]> desc problems;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
userName	varchar(10)	YES		NULL	
problemName	varchar(30)	YES		NULL	
problemContents	varchar(3000)	YES		NULL	
inputData	varchar(300)	YES		NULL	
outputData	varchar(300)	YES		NULL	
limitedTime	varchar(50)	YES		NULL	
level	varchar(20)	YES		NULL	

8 rows in set (0.001 sec)


```
MariaDB [oswtf_db]> desc source;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
userName	varchar(10)	YES		NULL	
problemId	int(11)	YES		NULL	
fileName	varchar(50)	YES		NULL	
date	varchar(50)	YES		NULL	

5 rows in set (0.001 sec)

그림 7. Table schema for saving problems and source code

그림 5는 회원 정보와 권한 정보를 저장하는 테이블 스키마를 표현한 것으로 외래키로 연결되어 있다. USERS 테이블에는 회원 정보를 저장하고, 추가 권한을 설정할 수도 있음을 고려하여 USER_ROLES 테이블에는 회원 역할을 저장한다. 회원 역할에는 그림 6과 같이 학생 (ROLE_STUDENT), 강의자(ROLE_LLECTURER), 관리자(ROLE_ADMIN) 정보를 저장했다. USERS 엔티티(entity)와 ROLES 엔티티는 1:N 관계(relation)로 사상(mapping)한 것이다.

```
MariaDB [oswtf_db]> desc user_roles;
```

Field	Type	Null	Key	Default	Extra
username	varchar(25)	NO	PRI	NULL	
role_id	int(10)	NO	PRI	NULL	

2 rows in set (0.001 sec)

```
MariaDB [oswtf_db]> desc users;
```

Field	Type	Null	Key	Default	Extra
username	varchar(25)	NO	PRI	NULL	
email	varchar(65)	NO		NULL	
password	varchar(100)	NO		NULL	

3 rows in set (0.001 sec)

```
MariaDB [oswtf_db]> desc roles;
```

Field	Type	Null	Key	Default	Extra
id	int(10)	NO	PRI	NULL	auto_increment
name	varchar(20)	NO		NULL	

2 rows in set (0.001 sec)

그림 8. Table schema for saving membership and authentication

```

MariaDB [oswtf_db]> select * from roles;
+-----+-----+
| id | name          |
+-----+-----+
| 1  | ROLE_STUDENT |
| 2  | ROLE_LECTURER|
| 3  | ROLE_ADMIN   |
+-----+-----+
3 rows in set (0.000 sec)

```

그림 9. The example of data in the ROLES table

3.3 시스템 구조

그림 7은 전체적인 시스템 구성도이고, 그림 8은 시스템 모듈 구성도이다. 회원 정보, 문제 정보, 소스 코드 정보를 저장하고 사용자가 요청한 웹 페이지를 처리하는 서버와 소스 코드를 실행하여 결과를 확인하는 채점 서버가 필요하다.



그림 10. System architecture

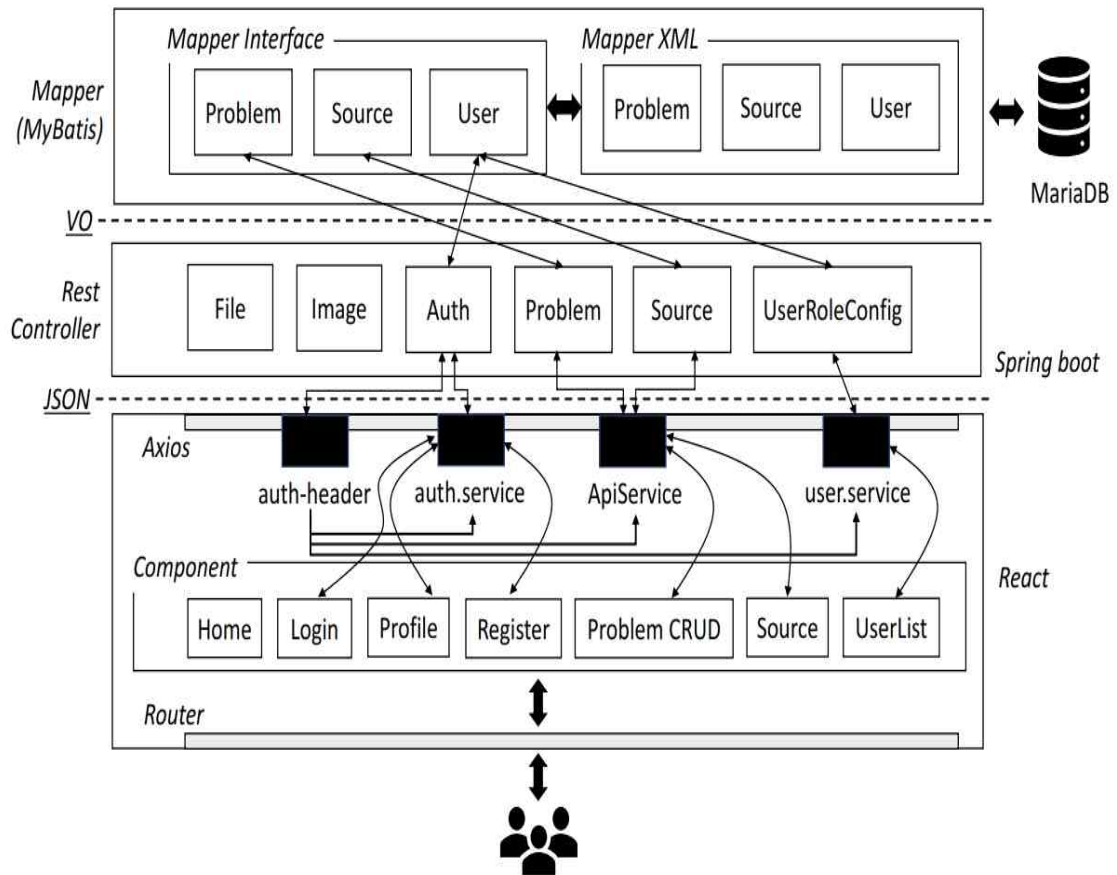


그림 11. System module configuration diagram

그림 8에서 사용자의 요청을 제일 먼저 처리하는 모듈은 React의 Router라는 모듈로 React에서 정의한 UI 컴포넌트로 연결하여 사용자가 요청한 화면을 전환하게 된다. 이 화면에서 입력한 데이터나 서버에서 받은 데이터는 React의 Axios 모듈과 Spring boot 서버 사이에 JSON 데이터로 통신하게 된다. Spring boot는 RESTful API로 요청과 응답을 받을 수 있도록 구성하였다. 인증(Auth), 문제(Problem), 소스 코드(Source), 회원 역할 수정(UserRoleConfig)와 같은 Spring boot 내에 있는 서비스들은 DB에서 CRUD(Create, Read, Update, Delete)를 수행해야 하므로 MyBatis Mapper로 MariaDB와 연결한다.

IV. 시스템 구현

4.1 개발 환경

제안 시스템에 대한 개발 환경 설정은 아래 표 7과 같다.

표 7. Development environment

WAS	Spring boot 2.3.2 (JDK 11)
Server build tool	Gradle
Front-end UI	React 17
Front-end build tool	NPM 12.6.0, Yarn
Deployment server OS	Ubuntu 20.04.1

표 7에서 WAS(Web Application Server)는 Spring boot를 사용하였으며 기존 스프링 프레임워크(Spring framework)를 모듈 기반으로 만든 프레임워크이다. 따라서, 단독 실행이 가능하고 최소한의 모듈 설정으로 경량화 서버를 구축할 수 있다.

Front-end는 Facebook에서 개발한 React를 사용하였으며 컴포넌트(component) 기반으로 사용자 인터페이스(User Interface)를 구축할 수 있는 프레임워크이다. 사용자 실행 환경은 크롬(Chrome) 브라우저에서 안정적으로 동작할 수 있도록 이를 기준으로 하여 구현하였다.

4.2 모듈 구현

서버의 모듈은 크게 3가지로 나눌 수 있다. 로그인, 회원가입, 회원 권한 관리를 위한 회원 인증 모듈과 경진대회에 필요한 문제 CRUD 모듈, 학생이 소스 코드를 저장하고 실행할 수 있는 소스 코드 CRUD 모듈로 구성되어 있다.

4.2.1 회원 인증 모듈

그림 9는 회원 인증 절차를 나타낸 시퀀스 다이어그램(sequence diagram)이다. 먼저 회원가입 절차에서는 `api/auth/signup`으로 사용자가 POST로 요청을 보내면 해당 회원이 중복으로 존재하는지 DB에서 확인한다. 해당 회원이 신규 회원이라면 성공 메시지를 JSON 형태로 보낸다. 로그인 절차에서는 `api/auth/signin`으로 POST로 사용자가 요청하면 올바른 회원 정보이면 JWT 토큰 키(token key)를 문자열 형태로 생성해 이를 타입(type), 회원정보(user info), 권한 정보(role)를 클라이언트에 넘겨줘 로그인에 성공하게 된다. 권한 별 접근은 이 토큰 정보를 이용해 서버에 있는 키와 교차 검증을 수행하여 서버가 권한에 해당하는 올바른 데이터를 제공할 수 있도록 한다.

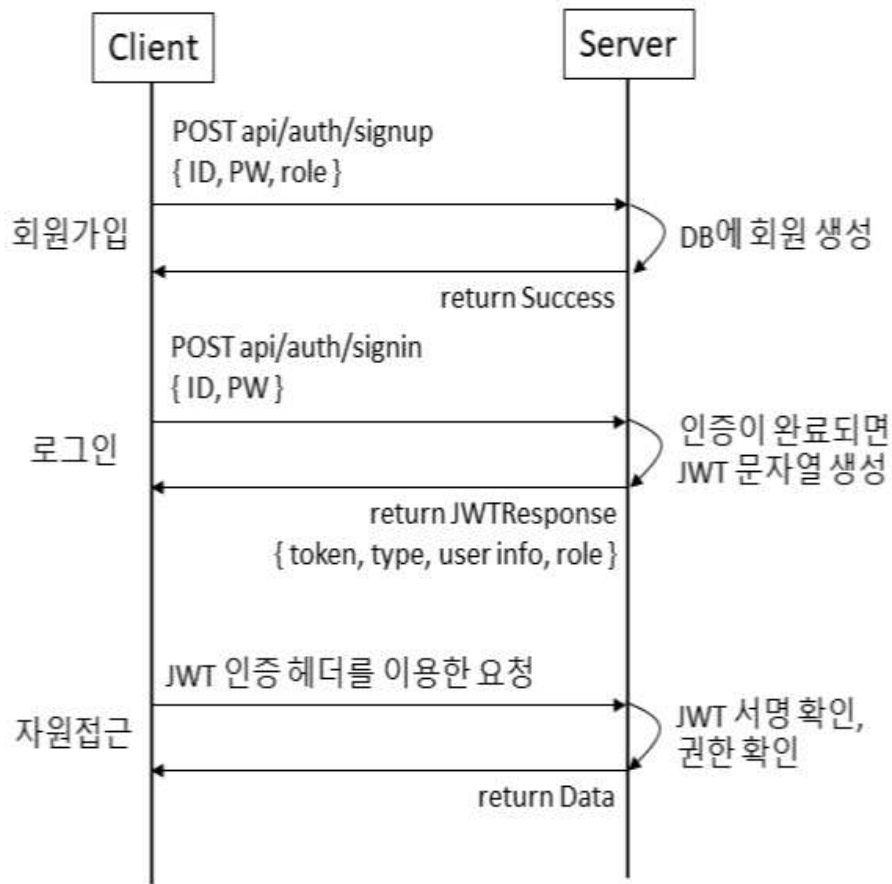


그림 12. Sequence diagram for membership authentication module

회원 인증 모듈에 대한 구성도는 그림 10과 같다. 먼저 로그인, 회원가입, 권한과 같은 회원 인증과 관련된 요청은 Auth Controller를 통해 들어오며, 응답도 이를 통한다(그림 10에서 ❶). 그러면 Auth Controller는 회원 인증과 관련된 처리를 OncePerRequestFilter로 위임한다(그림 10에서 ❷). OncePerRequestFilter는 1명의 사용자 요청에 대해 1번만 실행되는 구조로써 중복 로그인을 막는 객체이다. 그리고 doFilterInternal메서드를 통해 JWT 토큰을 파싱(parsing)하여 검증하는 역할을 수행한다.

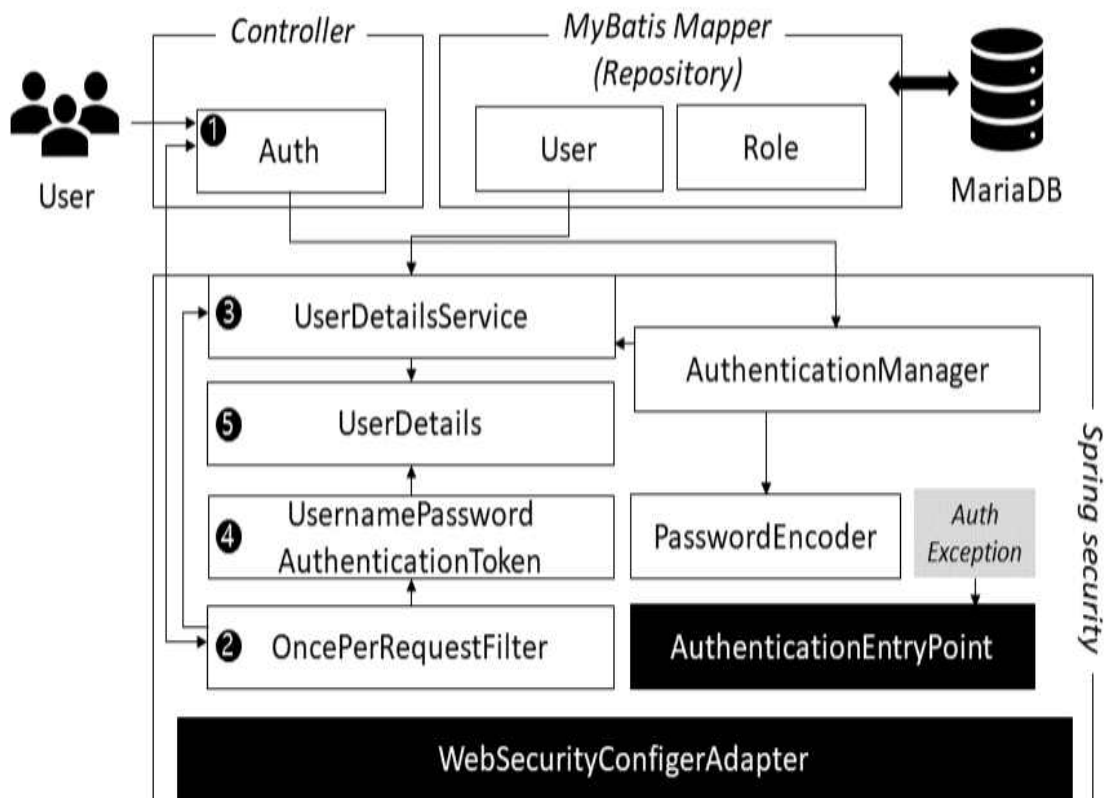


그림 13. Configuration diagram of member authentication module

검증이 완료되면 사용자 정보 객체(UserDetails)를 UserDetailsService 인터페이스를 통해 DB에서 불러온다(그림 10에서 ③). 이 UserDetails 객체를 이용해 UsernamePasswordAuthencationToken을 사용하여 권한을 검사한다(그림 10에서 ④). UsernamePasswordAuthencationToken은 로그인 요청 객체로부터 사용자 ID와 패스워드를 AuthenticationManager에서 유효성을 검증하고 PasswordEncoder를 통해 암호화를 수행한다. 마지막으로 유효성 검증이 완료되면 사용자 ID와 암호화된 패스워드, 권한 정보가 담긴 UserDetails 객체를 Auth Controller를 통해 응답하게 된다(그림 10에서 ⑤).

이러한 회원 인증 모듈은 Spring security를 통해 구현했으며 CORS(Cross Origin Resource Sharing)와 CSRF(Cross-Site Resource Sharing)을 허용하지 않도록 설정했다. 이러한 보안에 전반적인 정책은 WebSecurityConfigurerAdapter가 담당한다. 그리고 허용되지 않은 권한 에러(401 Error)와 같은 예외(exception)가 발생하면 AuthencationEntryPoint가 이를 처리하게 된다.

4.2.2 UI 컴포넌트

그림 11은 React를 기반으로 구성한 React 컴포넌트 구성과 처리 흐름을 나타낸 것이다. React는 앞서 언급했듯이 컴포넌트 기반 구조를 사용해 사용자 인터페이스를 관리하기 때문에 각 페이지마다 컴포넌트를 구성할 수 있고 각 페이지의 일부를 컴포넌트로 구성할 수 있다. 이 시스템에서는 전자의 방법을 사용하였다.

그림 11과 같이 각 컴포넌트를 강사, 학생, 관리자별로 다르게 접근하도록 구성하였다. 먼저 사용자가 원하는 화면을 컴포넌트와 연결시켜주는 역할을 React의 Router 컴포넌트가 수행한다. 로그인이나 회원가입, 프로필, 문제 리스트는 공통 기능으로, 모든 사용자가 이용할 수 있는 컴포넌트는 auth.service 모듈을 거치도록 구현하였으며 나머지 기능들은 api.service 모듈을 거치도록 구현하였다. auth.service와 api.service 모듈은 브라우저의 지역 저장소(local storage)에 저장된 권한 정보를 HTTP 헤더(header)에 삽입하여 Axios로 전달한다. 이 헤더에는 JWT 토큰 값이 들어 있으며 이 헤더가 없거나 유효하지 않은 토큰 값이면 권한 에러가 발생하며 사용자가 요청한 것을 거부한다. 마지막으로 서버와 JSON으로 통신하는 역할을 Axios가 수행한다. 따라서 동적 웹 페이지를

구성하는데 통신과 UI를 구분하여 수행할 수 있게 된다.

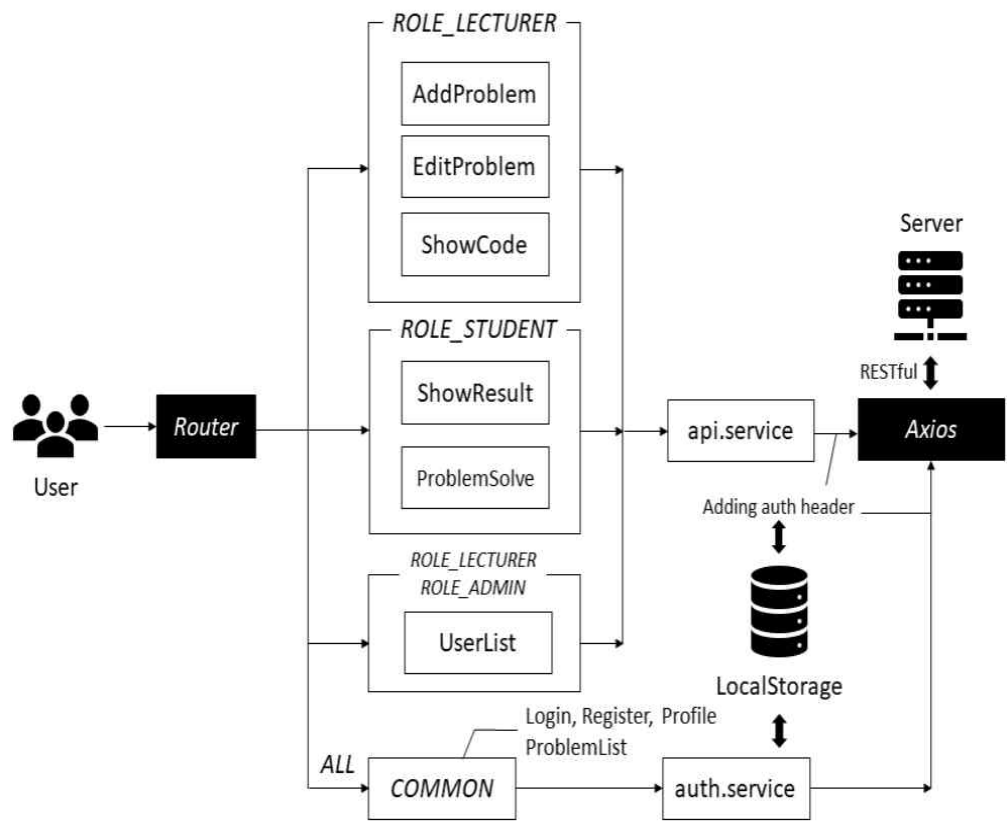


그림 14. React-based UI components and commuication module configuration diagram

4.2.3 문제 관리와 소스 코드 관리 모듈

강사가 문제를 추가하고 수정하기 위해 React-draft-wysiwig와 Draft.js를 기반으로 위지윅 편집기를 구현하였다[30]. 이미지 업로드 부분은 이미지를 서버에 저장해야 하므로 Axios를 거쳐야 한다. 그림 12는 위지윅 편집기를 반영하여 문제를 생성하는 페이지이다.

출제자

lecture

문제명

B ▾

표준 ▾

16 ▾

글꼴 ▾

☰ ▾

≡ ▾

▾

▾

그림 15. Problem creating page

학생들이 소스 코드를 추가하고 수정할 수 있도록 Microsoft Visual Studio Code의 코드 에디터를 웹 기반의 오픈소스로 제공한 Monaco 에디터를 이용해 구현하였다. 하지만, Monaco 에디터는 React 위에서 동작해야 하므로 NPM(Node Package Manager)를 통해 React 기반의 Monaco 에디터를 설치해야만 한다[31]. 그림 13은 소스 코드 편집 페이지이다. 소스 코드는 서버에 파일로 저장된다.

문제이름

최단경로 구하기

문제 내용

error

입력 데이터

error

출력 데이터

error

```
1 function Hello() {  
2  
3     printf("Hello");  
4 }
```

저출

그림 16. Source code editing page

4.3 구현 결과

그림 14는 학생이 접속했을 때 문제 리스트를 보여주는 화면이다. 만약 강사가 이 페이지를 접속하게 되면 문제를 수정하고 삭제할 수 있는 버튼이 활성화된다. 학생이 접속했을 때와 강사나 관리자가 접속했을 때 화면은 모두 같은 UI로 구성하였다.

OSWTF						문제풀기	LOGOUT
Problem List							
번호	출제자	문제명	제한시간	난이도	문제풀기		
23	admin	최단경로 구하기	15분	중			
24	lecture	출구 찾기	30분	상			
27	lecture	하노이 탑 쌓기	12분	하			

그림 17. Problem list page

강사와 관리자 입장에서 학생이 문제에 대한 소스 코드를 작성한 결과를 볼 수 있도록 그림 15와 같이 UI를 구성하였다. 해당 문제를 푼 학생의 ID를 화면 왼쪽에 출력하고 학생의 ID를 선택하면 화면 오른쪽에 소스 코드가 출력된다.







OSWTF

문제관리

LOGOUT

뒤로가기

최단경로 구하기

학생이름	코드보기
student	
student1	
student2	
student4	
student7	
student5	

student

```

1 function () {
2   return ""
3 }
4
5

```

그림 18. Source code output page for each student

V. 결론 및 향후 연구

본 논문에서는 우리나라의 소프트웨어 교육 현황을 우리나라 교육부에서 추진하고 시행되고 있는 소프트웨어 교육과정과 학교 실습 현장에 면에서 노후 컴퓨터 문제와 인터넷 망 문제로 인한 교육부에서 개선하려는 정책도 제시되고 있음을 보였다. 또한, 이미 상업적으로 출시된 플랫폼을 비교함으로써 비용적인 문제, 학생과 강사에게 필요한 기능적 문제를 제시하였다.

이러한 문제를 해결하고자 온라인 실습 환경과 교내 알고리즘 경진대회 환경을 통합해 구축하였다. 이를 통해 기존 상업 플랫폼에서 제공하고 있었던 실습실 컴퓨팅 환경에 구애받지 않고 웹 브라우저만 있으면 어디서든지 접속할 수 있는 기능을 구현하였다. 특히, 학교 실습 환경에서 학생과 강사에게 필요한 요구사항을 수집하여 이를 바탕으로 설계하였다는 점에서 다른 상업 온라인 프로그래밍 환경에 비해 경량화하여 구축할 수 있었다.

하지만 채점 서버는 여러 언어와 버전에 맞는 컴파일러(compiler)가 배치되어 있고 경진 대회나 수업과 같이 동시에 학생이 컴파일하려는 소스 코드를 동시에 요청하게 된다. 따라서 이와 관련된 동시성 테스트나 부하 테스트(stress test)를 추가로 수행해야 할 것으로 보여진다. 또한, 기능적인 면에서 학생과 강사가 만족할 만한 기능을 완전히 구현했는지와 관련된 설문 조사, 만족도 평가를 통해 향후 이를 보완한 연구가 필요할 것으로 보여진다.

참고문헌

- [1] J. H. Park, “An Analysis on the Current Status and Effectiveness of Software Education Leading School,” *Journal of Digital Contents Society*, Vol. 21, No. 10, pp. 1845–1854, Oct. 2020.
- [2] Department for Education. The national curriculum in England [Internet], Available: <https://www.gov.uk/government/publications/national-curriculum-in-england-primary-curriculum>.
- [3] K. S. Kim, “An Implications of Computer Education in Korea from the U.S., U.K. and Germany Computer Curriculums,” *Journal of the Korean Association of Information Education*, Vol. 20, No. 4, pp. 421–432, Aug. 2016.
- [4] MIT Media Lab. Scratch official site [Internet], Available: <http://scratch.mit.edu>.
- [5] S. Y. Jung, S. Y. Yoon and C. S. Park, “A SW Education through making chatbot,” In *Conference of Korea Intelligent Information Systems Society*, pp. 143–145, Aug. 2017.
- [6] Ministry of Science/ICT & Future Planning(Korea) and Ministry of Education(Korea), “Talent training promotion plan for a SW-oriented society,” Jul. 2015.
- [7] Telecommunications Technology Association(TTA). Computational Thinking [Internet]. Available: https://terms.tta.or.kr/dictionary/dictionaryView.do?word_seq=166866-9.

- [8] J. M. Wing, “Computational thinking.” *Communications of the ACM*, Vol. 49, No. 3, pp. 33–35, Mar. 2006.
- [9] CSTA & ISTE(2011). Computational Thinking teacher resources second edition. Available: http://csta.acm.org/Curriculum/sub/CurrFiles/472.11CTTeacherResources_2ed-SP-vF.pdf.
- [10] Ministry of Education(Korea), “Scientific Literacy for All Koreans Korean Science Education Standards for the Next Generation,” 2019.
- [11] Ministry of Education(Korea), “2015-74 Separate book 10 - Practical course(Technology&Home Economics) Computer Science Curriculum,” 2015.
- [12] M. S. Kim, “Current status and implications of domestic and overseas SW education for training SW talent,” National IT Industry Promotion Agency(NIPA), Issue Report 2018-53, Dec. 2018.
- [13] H. Y. Kil, “Case Study on Global Software Education in Schools,” *Journal of the Korea Society of Computer and Information*, Vol. 24, No. 9, pp. 151–160, Sep. 2019.
- [14] French Academy of Sciences, “Teaching computer science in France – Tomorrow can't wait,” May. 2013.
- [15] G.-L. Baron, B. D.-D, M. Grandbastien, F. Tort, “Computer Science Education in French Secondary Schools: Historical and Didactical Perspectives,” *ACM Transactions on Computer Education*, Vol. 14, No. 2, Jun. 2014.
- [16] Y. H. Kim, J. Y. Yu, N. J. Kim, “Elementary and secondary

software education operation status and improvement tasks,” National Assembly Research Service (Korea), Legislative Policy Report No. 34, Dec. 2019.

- [17] Programmers [Internet], Available: <https://programmers.co.kr>
- [18] Goorm IDE [Internet], Available: <https://www.goorm.io>
- [19] BAEKJOON ONLINE JUDGE [Internet], Available: <https://www.acmicpc.net>
- [20] Codewars [Internet], Available: <https://www.codewars.com>
- [21] Codility [Internet], Available: <https://www.codility.com>
- [22] LeetCode [Internet], Available: <https://leetcode.com>
- [23] Google Colab [Internet], Available : <https://colab.research.google.com>
- [24] Kaggle [Internet], Available : <https://www.kaggle.com/notebooks>
- [25] Microsoft Azure Notebook [Internet], Available : <https://notebooks.azure.com/>
- [26] Amazon Sagemaker [Internet], Available : <https://aws.amazon.com/ko/sagemaker/>
- [27] Axios (MIT License). Github Axios [Internet], Available: <https://github.com/axios/axios>.
- [28] ECMA International. ECMAScript(r) 2015 Language Specification (ES6). pp. 483, Jun. 2015.
- [29] Internet Engineering Task Force (IETF). RFC 7519 - JSON Web Token (JWT). May. 2015.
- [30] Jpuri. React-draft-wysiwyg (Github, MIT License) [Interne

t], Available: <https://github.com/jpuri/react-draft-wysiwyg>.

- [31] NPM(Node Package Manager). Monaco-editor/react (MIT License) [Internet], Available: <https://www.npmjs.com/package/@monaco-editor/react>.

Design and Implementation of Online Practice Environment for Programming Education

Habin Song

*Department of Computer Engineering,
Graduate School of Paichai University
Daejeon, Korea*

(supervised by Professor Hoekyung Jung)

Recently, as interest in programming and artificial intelligence has increased, Software education is mandatory from elementary school. Programming education is an education that can cultivate problem-solving skills so that problems can be solved with computers. The Ministry of Education has recognized the curriculum accordingly and is operating some leading schools, and cooperation and connects between local universities and elementary and middle schools in software education are increasing.

In order to achieve this goal of programming education, it is necessary to basically establish a lab environment suitable for students and teachers. However, the lab environment has performance problems due to cost issues, and in particular, there is a problem that students must purchase and use an existing online platform while implementing an algorithm

contest program in which students access simultaneously to evaluate their problem solving ability. All of them have very high economic costs.

Therefore, in this paper, we propose a web-based online practice environment and algorithmic contest scoring system to solve these problems. This system was implemented through React and Spring boot. Students can be managed from the point of view of the online practice environment accessed from the student's point of view and from the point of view of the teacher. In particular, during the algorithm contest, students can access and view the source code of the problem solved collectively and perform scoring for it. To be. Through this, it is believed that it will be possible to reduce costs and facilitate education by simply building it on campus rather than using the existing online platform.

감사의 글(Acknowledgement)

4년간의 학사 과정을 마치고 대학원 석사 과정에 진학 한지 어느덧 2년의 세월이 흘렀습니다. 석사 과정 학술 연구와 더불어 치열했던 취업 전선에 뛰어들어, 저축은행 본사에 입사하여 본업과 함께 학업에 몰두하기 힘들 때도 많았습니다. 그러나 항상 뒤에서 든든하게 응원해주시고 지원해주신 아버지와 어머니께 먼저 감사의 말씀을 올리며, 그동안 저를 뒷받침해 주며 보듬어주신 분들께 직접 감사의 인사를 전해드리지 못하여 송구스럽지만, 지면을 통한 감사의 인사를 드리려 합니다.

먼저 지도교수님이자 인생의 조언자로서 저에게 아낌없는 조언과 격려를 해주신 정회경 교수님께 진심 어린 감사의 인사를 드리며, 학사 과정부터 항상 좋은 말씀과 알찬 수업으로 밝은 분위기를 만들어주셨던 김창수 교수님, 송정영 교수님께 감사의 인사를 드립니다. 또한 학사 과정 동기이며, 군대 선배, 대학원 선배이자 인생의 멘토·멘티로서 함께해오며, 앞으로도 돈독한 우정을 쌓을 전성우에게 감사의 인사를 전합니다.

이 짧은 글을 통해 모두에게 감사의 마음을 전해드리지 못하지만, 앞으로 본업에 충실하여 성공한 모습으로 보답하겠습니다.

마지막으로 저에게 아낌없는 조언과 격려를 해준 대학 동기들과 모든 분들께 감사의 말을 올립니다.

2021년 6월

송하빈