

문제정의서(연구계획서)

과제명

SW 자율학습 강의 Docker+웹 서비스 개발

조	야간자율학습 조
지도교수	이영석 교수님 (서명)
조원	201701972 고태완 201702002 김지혜 201702080 최수연

1. 연구의 필요성

현재 세계 이러닝 시장이 성장세를 이루면서 SW 자율학습 플랫폼 또한 증가하고 있다. 흔히 자주 쓰이는 SW 자율학습 플랫폼으로 **udemy**, **coursera**와 같은 MOOC와, **codecademy**, **inflex**, 생활코딩 등이 있다. 이러한 SW 자율학습 플랫폼들로 인해 많은 사람들이 인터넷으로도 쉽고 간편하게 SW 학습을 할 수 있게 되었다. 하지만 이를 포함한 많은 SW 자율학습 플랫폼들에서 몇 가지 한계점이 드러난다.

먼저, SW 학습의 상당 부분이 실습 위주인 것에도 불구하고 실습 환경을 제공하는 서비스가 드물다는 것이다. 그래서 많은 사용자들이 SW 실습 강의를 따라가기도 전에 환경 설정에서 막히는 경험을 한다. 학습 강사가 환경 설정에 대한 강의를 해 주는 경우도 있으나, 사용자마다 자신의 개발 환경 상태가 다를 것이므로 환경 설정 강의로는 한계가 있다. 또한 사용자가 자신의 수준을 인지하기 어려워서 수준에 맞지 않는 학습을 할 수 있다. 강의의 수준과 자신의 수준을 정확히 파악하지 못하고 강의를 학습하는 것에 어려움을 겪는 경우도 있다. 이렇게 사용자의 자율학습을 돕는 것에도 한계가 드러난다. 자율학습을 돕는 것에서 뿐만 아니라, 서비스 자체적으로도 한계점이 있다. 사용자가 강의를 직접 업로드 함으로써 서비스 사용자들이 자체적으로 서비스를 업데이트하는 기능 또한 모든 SW 자율학습 플랫폼들이 갖고 있진 않다. 이렇게 되면 서비스 운영자가 직접 서비스를 업데이트해야 하는 데, 그것 만으로는 지속적으로 업데이트를 하는 데 한계가 있다.

4차산업혁명이 현실화가 되어가면서 세계 각국에서 SW 역량을 갖춘 인재 양성에 힘쓰고 있다. 국내에서도 코딩 교육의 중요성을 인지하여 코딩 교육을 의무화하고 있다. 많은 재능 있는 SW 인재를 육성하는 것이 사회의 발전에 기여하는 것 뿐만 아니라, 더 나아가 국가 경쟁력 또한 확보 할 수 있다. 우리는 이러한 현재 SW 시장 흐름에 맞춰 기존에 SW 자율학습 플랫폼들의 한계점을 개선하여 SW 자율학습을 돕고 경쟁력 있는 SW 인재를 육성하는 SW 자율학습 플랫폼 서비스를 개발할 것이다.

2. 연구의 목표 및 내용

본 연구는 궁극적으로 SW 자율학습 웹 서비스인 "Code at Once"의 개발을 목표로 한다. 본 웹 서비스는 오픈형 플랫폼으로 사용자가 자율학습을 진행할 수 있도록 양과 질이 보장된 콘텐츠를 지속적으로 업데이트하는 것을 지향하며 꾸준한 사용자의 유입으로 서비스 유지를 목표로 한다.

본 연구는 사용자가 강의를 제작하여 SW 수강 과목과 제공된 테스트 폼을 이용하여 테스트를 업로드할 수 있고 이를 수강한 이용자는 평점과 리뷰, 질의 응답을 할 수 있도록 하는 오픈형 플랫폼이다. 이를 통해 고정적이거나 무분별한 콘텐츠를 배제할 수 있으며 콘텐츠가 지속적으로 갱신될 수 있다.

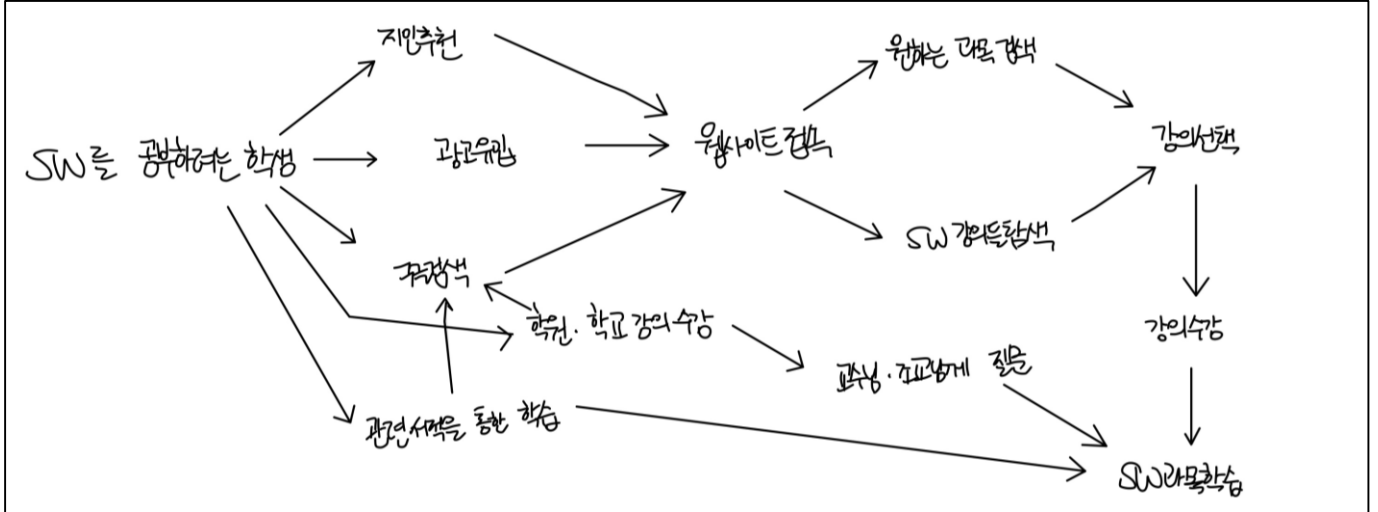
본 연구에서는 SW 과목을 공부하고자 하는 사용자가 자신의 수준을 인지하지 못해 최초 학습 시작 시 자가 수준 측정이 어려운 경우를 고려하여 **Level Test** 기능을 구현한다. 이 기능을 통해 서비스 이용자간의 평균 대비 자신의 수준을 인지할 수 있게끔 할 뿐만 아니라 사용자의 수준에 적절한 강의 리스트를 제공한다. 이론 뿐만 아니라 **Docker**를 이용하여 SW의 각 과목 실습 수강이 가능한 환경을 구축하여 코딩 실습 진행 시 환경 설정의 불편함을 해소한다. 과목 수강 완료 시에는 복습 테스트 기능을 제공하여 수강자가 강의에 대한 이해도를 평가하는데 도움을 준다.

자율학습으로 인한 학습 나태로 웹 서비스 재방문을 감소를 방지하기 위하여 사용자 계정에 주기적으로 이메일로 알림 학습 진도율을 전송한다. 뿐만 아니라 서비스 내 사용자 정보 관리 페이지인 '마이페이지'에서 수강 중인 강좌에 대한 학습 진도율, 테스트의 오답 문항 확인, 나의 질의 응답 내역, 강좌의 특정 부분으로 바로 이동이 가능한 기능을 구현하여 편의성을 제공한다.

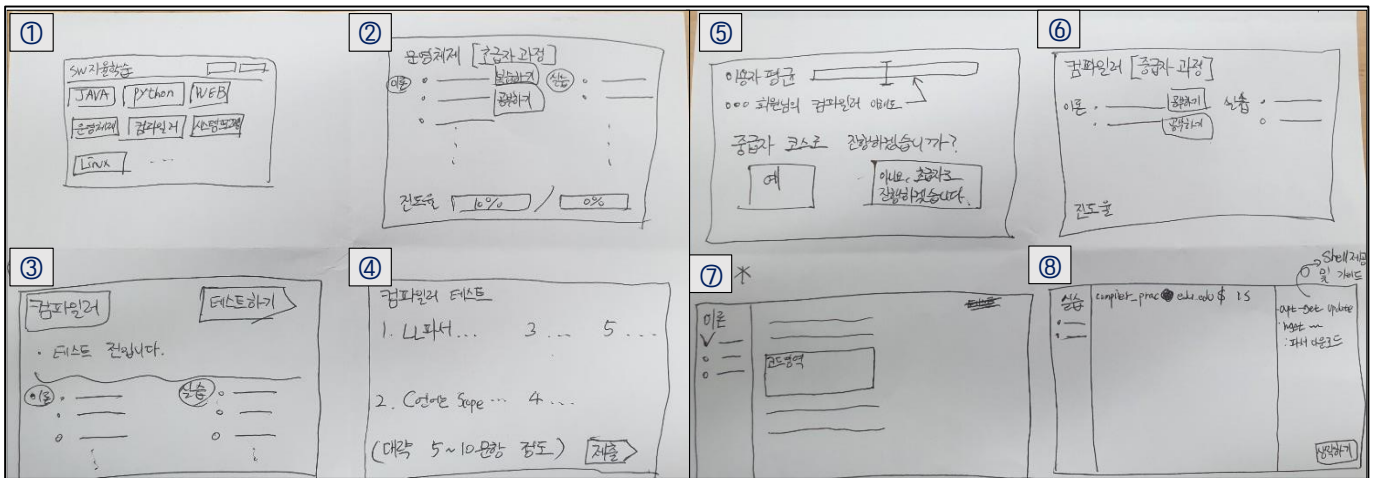
3. 연구의 추진전략 및 방법

구글에서 사용하는 의사 결정 방식인 Design Sprint를 이용한다.

1일차: Map, 지도를 그려서 문제를 정의하고 타깃을 정한다.



2일차: Sketch, crazy 8's를 통해 각자 솔루션을 생성한다.

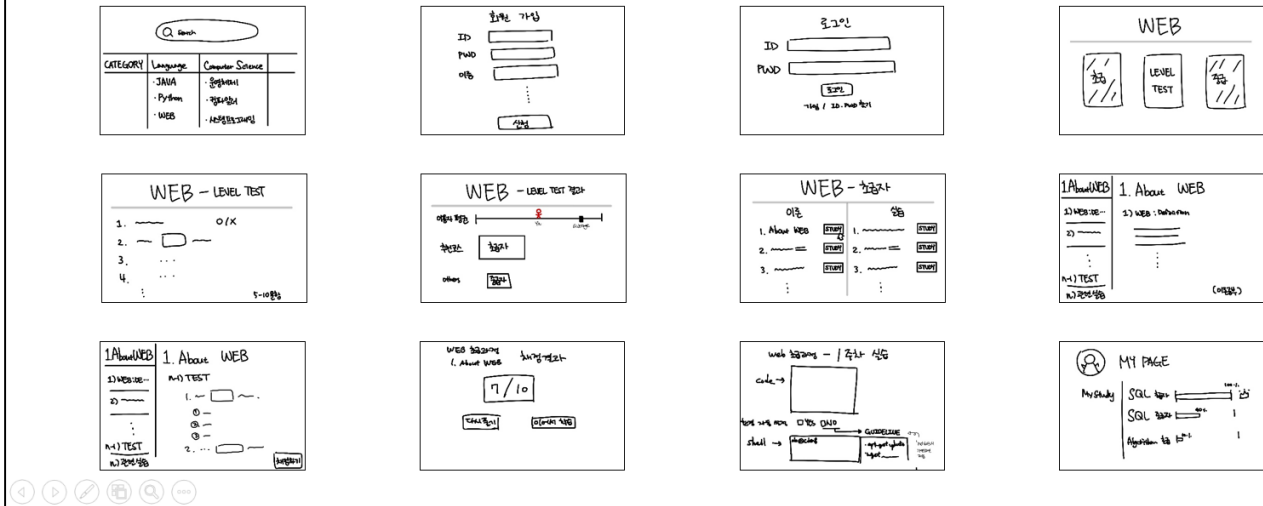


3일차: Decide, 가장 좋은 솔루션을 도출한다.

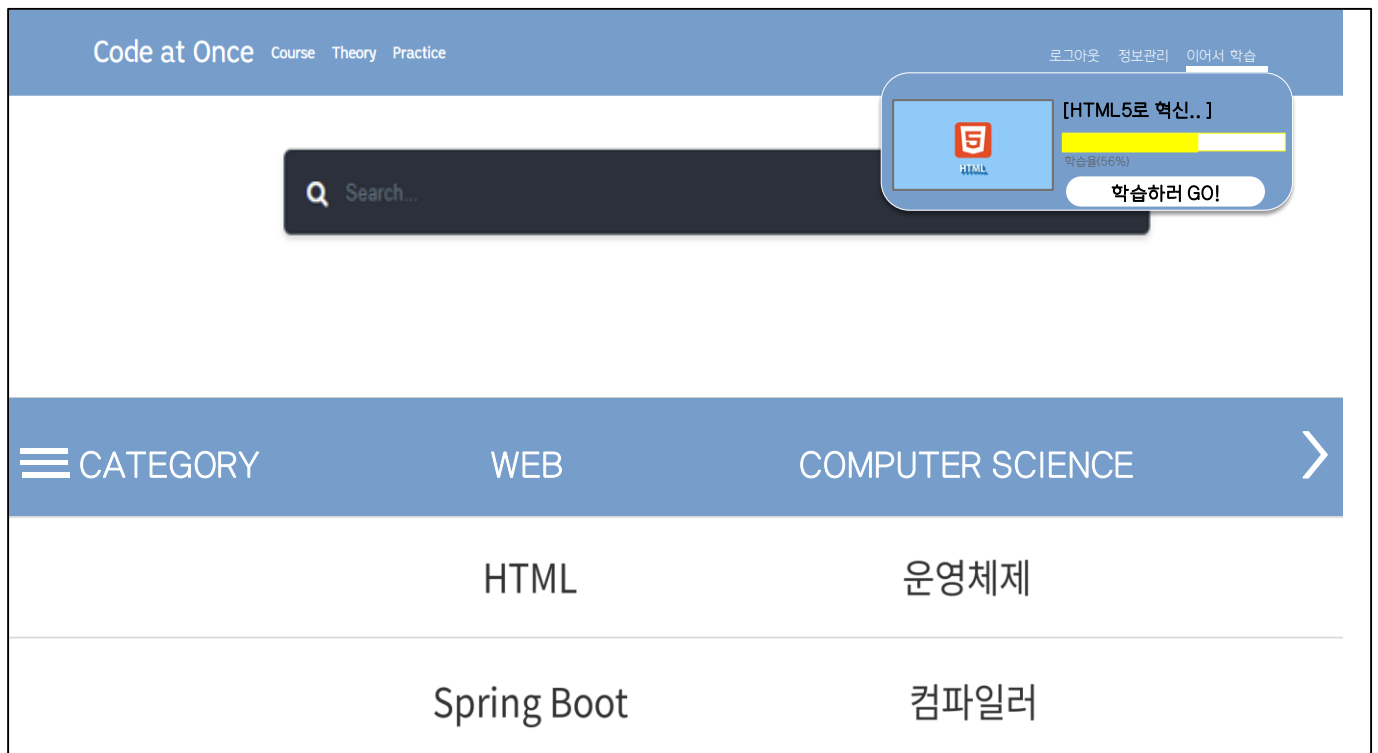
완성된 스토리보드

12슬라이드, 3표, 2표를 받았던 Sketch 반영

그 외에 필요하다고 생각하는 부분들(로그인, 회원가입 등)을 협의를 통해 추가



4일차: Prototype, 3일차에서 선택한 솔루션의 프로토타입을 만든다.



5일차: Test, Feedback을 통해 부족한 점을 보완하고 강점을 강화한다.

질문					
1-1. 기존 sw 학습 사이트 (예: 인프런, codecademy) 을 이용해본 경우가 있는가?	예	예	예	예	예
1-2. 위의 질문(1-1)에 '예' 라고 대답한 경우, 이용한 학습 사이트와 비교하여 어떤 점이 장점 혹은 단점이라고 생각하는가?	딱히 단점은 없지만 그렇다고 이 프로그램을 사용할 특별한 차별점을 느끼지 못하였다	장점:강의 수강 전 나의 레벨을 먼저 테스트 해볼수 있는 점이 좋은것같다. 이론과 실습을 따로 나누었지만 또 하단에 관련실습을 띄어주어 편한것 같다. 단점:내가 듣고 싶은 강의 목록을 저장 해두는것이 없는 것같고, 테스트 후 점수만 알려줘서 틀린부분을 알고 싶으면 불편할 것 같다.	장점 : 레벨 테스트를 통해 자신의 수준을 테스트할 수 있다는 점이 좋다. 또한 각 학습이 끝나고 테스트를 해 볼 수 있어서 좋다.	장점: level test를 통한 자기의 수준에 맞춘 강의가 좋은것 같습니다. 다른 사이트에서는 초급은 뭐다 해서 들었을 때 너무 쉬워서 중급으로 간 적도 있었는데 좋습니다. 그리고 퀴즈를 넣은 것도 좋습니다. 단점은 로드맵? 학습 로드맵이 인프런엔 있는데 '야간자율 학습'팀의 발표 자료에서는 없는 것 같습니다. 이 점이 아쉽습니다.	level- test 기능
2-1. 영상에서 소개된 프로토타입에 나와있는 기능 외에 추가로 있으면 더 좋을것 같은 기능이 있는가?	예	예	예	예	예
2-2.위의 질문(2-1)에 '예'라고 대답한 경우, 추가적으로 있으면 좋을 것 같은 기능은 무엇인가?	QnA 기능	1.아이디와 이름이 중복된 경우 중복 여부 방지 기능이 있으며 좋을 것 같다. 2.듣고싶은 강의 목록 저장 기능(매번 검색만 해서 들어야 하는것인지 아니면 my page의 학습 현황을 누르면 이어서 들을 수 있는 건지 영상만 봐선 잘 모르겠다) 3.강의 책갈피 기능(강의 중간에 이해가 잘 안가서 나중에 보고싶은 구간을 책갈피로 체크해둘수 있었음 좋겠다) 4.test 채점결과만 보여주지 말고 답도 알려줬음 좋겠다.	메인 페이지에서 '이어서 학습하기'버튼이 있어서, 누르면 최근에 들었던 강의를 이어 들을 수 있게 또는 최근에 학습했던 페이지로 바로 가는 기능이 있으면 좋을 것 같다.	실습 퀴즈에 그냥 코딩할 수 있는 환경만 구비되어 있는 것 같은데 퀴즈도 똑같이 있으면 좋겠습니다.	해당 수강수업에 대해서 매일 아침 10시에 폰으로 알림이 오는 기능도 좋을 거 같음 아니면 저녁 10시에 "오늘 수강하셨나요?" 이런 느낌으로 매일 또는 꾸준히 주기적으로 강의를 수강하게끔 도와주는 역할
3. 프로토타입에 나와있는 기능에서 개선해야 한다고 생각되는 부분이 있다면 어떤 기능이며 어떻게 대체하면 좋다고 생각하는가?	화팅!	만약 동영상 강의라면 속도조절이나 자막 기능이 있으면 좋겠다.	초급과 중급을 나누는 기준을 '초급'과 '중급'을 선택하는 페이지에 간략하게나마 적어주었으면 좋겠다. 레벨테스트를 할 필요는 없지만, 어떤 기준으로 나누었는지를 알고 선택하게 할 수 있으면 좋겠다. 예를 들면, 강의에서 가르는 목적을 적어둔다면지...?	1-2의 단점과, 2-2의 추가적으로 있으면 좋을 것 같은 기능을 참고해 주셨으면 좋겠습니다. 김지혜씨의 목소리가 쏘 서웠 ~ 합니다. 또 듣고 싶군요!	딱히 없음

Design Sprint를 통한 의사 결정이 끝난 후, SW 공학적인 계획을 수립하여 목표 기간을 설정한다. 이 때 목표한 기능을 완수할 때 마다 마일 스톤을 세워 계획을 점검한다.

담당 교수 1명, 참여 인원 3명 그리고 담당 멘토 1명과 함께 지속적인 미팅 및 진척 상황 공유를 통해 협업한다. 이 때 협업은 Slack, Github, Zoom과 같은 협업 도구를 이용한다.

4. 연구 팀의 구성 및 과제 추진 일정

프로젝트의 구성 인원은 다음과 같다.

- 이영석: 담당 지도 교수
- 고태완: 클라이언트 개발 담당
- 김지혜: 서버 개발 담당
- 최수연: 데이터 크롤링, 전처리 및 분석
- 하석재: 담당 기업 멘토, 멘토링

추진 일정은 다음과 같다.

Code at Once 프로젝트 일정표								
	05.04 05.18	05.18 06.01	06.01 06.15	06.15 06.29	06.29 07.13	07.13 08.03	07.27 08.10	08.10 08.24
Client Develop	Javascript Environment Setting	UI Develop		Component Test	Connect with server	Release on live server	Integration Test	Final Test
Server Develop	Python Environment Setting	Network configuration	Database design	Endpoint construct & Server develop	Connect with client	Release client program	Integration Test	
Data Crawling, Organize and Storing	Python Environment Setting	Collecting software learning data			Data Analyze & Data Refining	Integration Test		

- 참고문헌(Reference)