

Requirement Specification

Web Service Program for Chat-bot Education in Python

소프트웨어공학개론 Team4



제출일	22.04.10.	그룹	Team4
과목	소프트웨어공학개론	담당교수	이은석 교수님
이름	양이삭	학번	2016312117
이름	김창현	학번	2015313546
이름	손영훈	학번	2020311323
이름	송예진	학번	2019310847
이름	정동진	학번	2017314786
이름	최가은	학번	2019315437

목차

1. Preface	07
1.1 Objective	07
1.2 Readership	07
1.3 Document Structure	07
A. Preface	07
B. Introduction	07
C. Glossary	07
D. User Requirement Definition	07
E. System Architecture	07
F. System Requirement Specification	08
G. System Models	08
H. System Evolution	08
I. Appendices	08
J. Index	08
2. Introduction	08
2.1 Objective	08
2.2 Needs	08
2.3 Service	10
2.4 Expectations	10
2.4.1 학습자	10
2.4.2 교수자	10
3. Glossary	11
3.1 Objective	11
3.2 Term Definition	11
3.3 Acronyms and Abbreviations	11
4. User Requirement Definition	12
4.1 Objective	12
4.2 Functional Requirement	12
4.2.1 회원 가입	12
4.2.2 로그인	12
4.2.3 메뉴바	13
4.2.4 회원 정보 보기	13
4.2.5 회원 정보 수정하기	13
4.2.6 회원 정보 삭제하기	13
4.2.7 회원 리스트 보기	13

4.2.8 공지사항 작성하기	13
4.2.9 게시물 작성하기	13
4.2.10 공지사항·게시글 보기	13
4.2.11 공지사항·게시글 수정하기	14
4.2.12 공지사항·게시글 삭제하기	14
4.2.13 게시물 답글 달기	14
4.2.14 공지사항·게시글 리스트 보기	14
4.2.15 이론 학습하기	14
4.2.16 연습 문제 풀기	14
4.3 Non-functional Requirement	14
4.3.1 Product Requirement	14
4.3.1.1 Usability Requirement (UI)	14
4.3.1.2 Efficiency Requirement	15
4.3.1.3 Dependability Requirement	15
4.3.1.4 Security Requirement	15
4.3.2 Organization Requirement	15
4.3.2.1 Environmental Requirement	15
4.3.2.2 Operational Requirement	15
4.3.2.3 Development Requirement	15
4.3.3 External Requirement	15
4.3.3.1 Regulatory Requirement / Legislative Requirement	15
4.3.3.2 Ethical Requirement	16
4.3.3.3 Safety / Security Requirement	16
5. System Architecture	16
5.1 Objective	16
5.2 System Architecture	16
5.3 Sub system	17
5.3.1 사용자 시스템	17
5.3.2 Python & Web Editor	17
5.3.3 게시판 시스템	18
6. System Requirement Specification	18
6.1 Objective	18
6.2 Functional Requirement	19
6.2.1 회원 가입	19
6.2.2 로그인	19
6.2.3 이론 학습하기	19

6.2.4 연습 문제 풀기	20
6.2.5 채점 결과 확인하기	20
6.3 Non-functional Requirement	21
6.3.1 Product Requirement	21
6.3.1.1 Usability Requirement (UI)	21
6.3.1.2 Efficiency Requirement	21
6.3.1.3 Dependability Requirement	21
6.3.1.4 Security Requirement	21
6.3.2 Organization Requirement	21
6.3.2.1 Environmental Requirement	21
6.3.2.2 Operational Requirement	22
6.3.2.3 Development Requirement	22
6.3.3 External Requirement	22
6.3.3.1 Regulatory Requirement / Legislative Requirement	22
6.3.3.2 Ethical Requirement	22
6.3.3.3 Safety / Security Requirement	22
6.4 Scenario	22
6.4.1 회원 가입	22
6.4.1.1 로그인	22
6.4.2 이론 학습하기	22
6.4.2.1 Initial Assumption	22
6.4.2.2 Normal Flow of Event	22
6.4.2.3 What can go wrong	23
6.4.2.4 System State on Completion	23
6.4.3 연습문제 풀기	23
6.4.3.1 Initial Assumption	23
6.4.3.2 Normal Flow of Event	23
6.4.3.3 What can go wrong	23
6.4.3.4 System State on Completion	23
6.4.4 채점 결과 확인하기	23
6.4.4.1 Initial Assumption	23
6.4.4.2 Normal Flow of Event	23
6.4.4.3 What can go wrong	24
6.4.4.4 System State on Completion	24
7. Sysetm Model	24
7.1 Objective	24

7.2 Context Models	24
7.2.1 Context Model	24
7.2.2 Process Model	25
7.3 Interaction Models	26
7.3.1 Use Case Model	26
7.3.2 Tabular Description	26
7.3.2.1 회원 가입	26
7.3.2.2 프로필 확인	27
7.3.2.3 로그인/로그아웃	27
7.3.2.4 이론 학습하기	27
7.3.2.5 연습 문제 풀기	27
7.3.2.6 채점하기	27
7.3.2.7 글 작성하기	28
7.3.2.8 공지사항 작성하기	28
7.3.2.9 답글 작성하기	29
7.4 Behavioral Models	29
7.4.1 Event-driven Diagram	29
8. System Requirement Evolution	30
8.1 Objective	30
8.2 Limitation	30
8.2.1 Output이 확인 불가능한 경우	30
8.2.2 국소적인 내용	30
8.2.3 아이디나 비밀번호 찾기 불가능	30
8.3 System Evolution	30
8.3.1 Evolution for Practice checking	31
8.3.2 Evolution for curriculum	31
8.3.3 Evolution for User Database	31
8.3.4 Evolution for Board system	31
9. Appendix	31
9.1 Objective	31
9.2 Database Requirement	31
9.3 Database Detail	31
9.4 Login Details	32
10. Index	32
10.1 Objective	32
10.2 Table Index	32

10.3 Figure Index.....	32
10.4 Diagram Index.....	32
<i>11. Reference</i>	<i>33</i>

1. Preface

1.1 Objective

Preface에서는 이 문서의 예상된 독자와 문서의 전체적인 구조, 각 파트에서 서술하는 세부사항들에 대해 설명한다.

1.2 Readership

User Requirement Readership

User Requirement에서는 본 Software System의 요구사항에 대한 것을 사용자의 관점에서 추상화 및 시각화를 이용하여 정의하고 표현하며 서술한다. 전문 지식이 없는 독자들을 고려하여 자연어와 도표, 그림 등을 이용하여 서술한다. Client manager와 Contractor manager가 주된 독자가 된다.

System Requirement Readership

System Requirement는 시스템을 구성하는 function, service, operational constraint에 대해 상세히 기술한다. 계약에 사용될 수 있으므로 구체적이고 구조화된 문서 양식으로 서술한다. 시스템을 개발하는 Software developer와 System architect들을 비롯한 개발진이 주된 독자가 된다.

1.3 Document Structure

A. Preface

Preface에서는 이 문서의 예상된 독자와 문서의 전체적인 구조, 각 파트에서 서술하는 세부사항들에 대해 설명한다.

B. Introduction

Introduction에서는 본 시스템의 필요성을 실제 사회에서의 트렌드를 분석하여 찾아낸다. 본 시스템을 개괄적으로 소개하고 Stakeholder별로 기대효과를 간략히 기술한다.

C. Glossary

Glossary에서는 문서에 나오는 용어들의 뜻을 설명한다. 배경지식이 없는 독자를 고려하여 문서를 이해하는 데 어려움이 없도록 가능한 모든 용어에 대해 서술한다. 단어를 명확히 정의하여 오해하거나, 의미를 다르게 추정하지 않도록 한다.

D. User Requirement Definition

User Requirement Definition에서는 본 시스템이 제공할 서비스에 대해 사용자의 관점에서 서술한다. 이 때 시스템에 대한 기능적 요구사항과 비기능적 요구사항을 모두 기술한다. 사용자의 이해를 돕기 위해 자연어로 기술하며, 필요에 따라 그림과 도표를 사용한다.

E. System Architecture

System Architecture에서는 본 시스템의 구조에 대해 높은 수준의 추상도로 설명한다. 시스템을 구

성하는 컴포넌트와 이들의 상호작용을 기술한다. 서술의 이해도를 높이기 위해 시각화를 이용한다. 개발자가 시스템의 전체적인 구조를 이해하는 데 도움을 준다.

F. System Requirements Specification

System Requirement Specification에서는 시스템을 구성하는 각 기능들의 기능적 요구사항과 비기능적 요구사항을 기술적인 관점에서 상세히 서술한다. 구조화된 문서 양식을 통해 시스템 내 기능적 부분을 명확하게 설명한다.

G. System Models

System Models에서는 시스템 컴포넌트, 서브시스템, 시스템 간의 관계를 설명한다. Object Model, data flow model 등을 사용하고, Context, Process, Interaction, Behavioral이라는 4가지 관점에서 시각화를 통해 기술한다.

H. System Evolution

System evolution에서는 본 시스템에 사용한 가정과 이에 따라 발생하는 한계점을 설명한다. 또한 비즈니스 환경, 플랫폼, 사용자 요구사항의 변화에 따라 어떻게 본 시스템을 evolution하고 한계점을 보완할 수 있을지 서술한다.

I. Appendices

Appendices에서는 개발되는 시스템에서 사용할 구체적이고 자세한 정보를 제공한다.

J. Index

Index에서는 본 문서에 사용된 용어와 diagram, 기능에 대한 인덱스를 나타낸다.

2. Introduction

2.1 Objective

Introduction에서는 본 시스템의 필요성을 실제 사회에서의 트렌드를 분석하여 찾아낸다. 본 시스템을 개괄적으로 소개하고 Stakeholder별로 기대효과를 간략히 기술한다.

2.2 Needs

4차 산업혁명에 사회, 경제, 산업 구조를 변화를 가지고 왔다. 화이트칼라(White Collar)로 대표되는 기존의 인지적 노동은 소프트웨어와 인공지능이라는 자동화 기술에 의하여 대체되며, 급격한 생산성 증가를 가져왔다. 그렇기에 미래에는 기존의 산업구조가 무너지며, 현재 직업의 50-70%가 사라지고 새로운 형태의 직업이 등장할 것으로 예상하고 있다. 그에 따른 사회와 경제구조 또한 큰 변화를 예상 시가총액 변화 추이를 보면 SW와 인터넷 기반 서비스로 급변하고 있다.

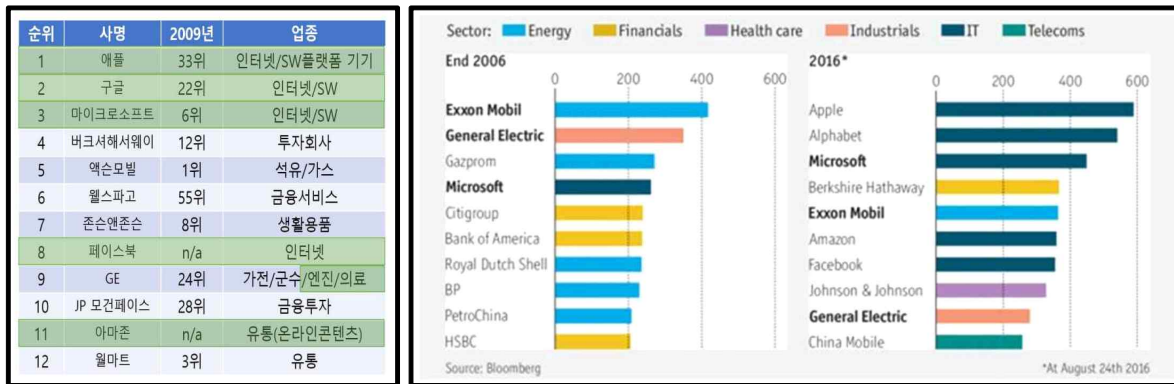


Figure 1 `06년과 `16년의 글로벌 기업 시가 총액 비교

뿐만 아니라 각 산업에서도 SW의 비중이 점차 증가하고 있다.



Figure 2 한국의 산업별 SW 활용도 및 선진국 대비 수준

그렇기에 급변하는 직업 환경에 대비하고 변화 주도를 위한 새로운 미래 역량을 필요로 한다. 이는 단순 지식습득과 루틴 업무 수행 능력보다는, 디지털 기반의 컴퓨팅사고력과 고차원의 문제 발견·해결 능력, 그리고 융합·창의력이 강조된다.

이를 위해서 교육 과정에서는 학문 간의 융합 현상이 더욱 더 강화할 필요가 있으며, 미래 사회에서 살아남는 인재를 양성하기 위해서는 융합 역량들을 배양할 수 있는 교양교육과정을 제공하는 것이 바람직하다. 지금까지는 대학 교양 IT교육 교과에서 이미 만들어진 소프트웨어를 활용하는 방법을 강

의하고 실습하도록 구성하여 제공하였다. 하지만 미래 사회를 이끌어 갈 창의적인 융합 인재들에게 필요한 것은 컴퓨터과학을 이해하고 자신들이 필요한 소프트웨어를 만들어 내는 능력을 훈련하는 교과과정과 다양한 연령의 학생들에게 범용적으로 사용가능한 교육 서비스를 필요로 한다.

그러나 현재의 교육 시스템은 입시 위주의 주입식 교육과 수능 중심의 교과 과목만을 위주로 가르치기에 학생 개인과 교육기관 자체에서 SW관련 교육에 대한 부담과 두려움을 안고 있는 실정이다. 그렇기에 본 서비스는 다수의 학생들에게 동일한 교육의 기회를 제공하고 실습을 통한 구현을 통해 프로그래밍 교육을 효율적으로 진행하는데 도움이 되고자 한다.

2.3 Services

Python을 이용하여 컴퓨팅 사고와 기본 프로그래밍 문법, 알고리즘, 자료구조들을 배울 수 있는 SW 교육 서비스를 제공한다. 본 서비스에 자체적으로 제작한 문제만이 아니라, 특정 기능을 구현하는 실습을 통해 학습한 내용을 복습하고 실제로 사용하는 방법을 익혀 프로그래밍 언어를 더 쉽게 이해할 수 있게 한다.

시스템은 크게 실습 기능, 게시판 기능을 가지고 있다.

실습 기능은 사용자에게 python을 교육시키는 기능이다. Python에서의 어떤 개념에 대한 설명과 하나의 파트를 잘 이해했는지 평가받는 테스트로 이루어져 있다. 먼저 어떤 개념에 대한 설명이 단계별로 주어진다. 설명 부분에는 개념 설명도 있고, 해당 개념을 활용하는 코드를 예시로 들면서 설명하는 경우도 있다. 단계별로 설명이 계속 이어지다가 하나의 파트를 끝내고 나면 지금까지 공부한 모든 단계의 내용에 대하여 테스트를 보게 되고, 기준 이상의 점수를 받는다면 다음 파트로 넘어간다.

게시판 기능은 자유게시판과 공지사항 게시판으로 종류가 나뉜다. 공지사항 게시판은 관리자가 모든 사용자들에게 전달하는 공지사항이 올라가는 게시판이다. 시스템을 사용하는 모든 사용자가 볼 수 있다. 자유게시판은 모든 사용자가 글을 자유롭게 작성할 수 있는 게시판이다. 어떤 개념에 대하여 이해가 되지 않을 때나 문제가 잘 풀리지 않을 때 등의 경우에 다른 사용자들에게 질문하는 글을 작성할 수 있다. 설명을 해줄 수 있는 다른 사용자가 답글을 달아 질문에 답을 해줄 수 있다.

2.4 Expectations

2.4.1 학습자

학습자는 가장 기초적인 개념에서부터 시작해서 최종적으로는 실시간 채팅 기능을 구현할 수 있을 때까지 python을 공부하는 것에 본 시스템을 사용할 수 있다. 기본적으로 사용하는 문법을 공부하기 때문에 python 기반 프로그래밍 실력 향상을 기대할 수 있다. 또한 융합인재가 갖추어야 할 사고력, 문제해결능력, 창의력, 의사표현능력, 자기 주도적 학습능력의 향상 효과를 기대한다.

2.4.2 교수자

수업에 추가로 활용할 수 있는 보충자료 같은 도구로 사용할 수 있다. 관리자로 등록되면 모든 사용자가 학습하고 있는 단계를 볼 수 있어 학생들이 어느 부분에서 잘 이해가 안 되고 있는지 파악하거나,

수업 진도 또는 수업 속도를 조절하는데 있어서 도움이 된다. 또한 많은 학생들에게 동등한 교육의 기회와 질을 제공할 수 있다.

3. Glossary

3.1 Objective

Glossary에서는 문서에 나오는 용어들의 뜻을 설명한다. 배경지식이 없는 독자들을 고려하여 문서를 이해하는 데 어려움이 없도록 가능한 모든 용어에 대해 서술한다. 단어를 명확히 정의하여 왜하거나 의미를 다르게 추정하지 않도록 한다.

3.2 Term Definition

용어	정의
Python	컴퓨터 언어의 일종으로 문법이 다른 언어에 비해 쉽고 간결하며 사람의 사고 체계와 닮아 있어 빠르고 강력하게 개발할 수 있다.
파트	대단원을 나타낸다. 자료형, 연산자, 함수 등을 예시로 들 수 있다.
단계	파트 안에 들어있는 소단원을 나타낸다. 연산자 파트를 예시로 들자면 산술연산자, 비교연산자, 논리연산자, 비트연산자 등을 들 수 있다.
실습 문제	각 단계에서 설명을 보고, 해당 내용을 직접 코딩하여 결과를 확인해보도록 해주는 간단한 개념 확인 문제이다.
연습 문제	하나 또는 연관된 몇개의 파트가 끝나면 해당 파트에 있는 모든 단계에서 배웠던 개념에 대하여 제대로 이해했는지를 평가하는 테스트이다.
공지사항	관리자가 모든 사용자에게 전달하는 공지를 나타낸다.
게시글	자유게시판에 올라가는 모든 글을 말한다.
에디터	학습자가 직접 코드를 입력할 수 있는 공간을 말한다. 결과는 포함하지 않고 오로지 코드만 포함한다.
유저	본 서비스를 이용하여 학습하는 사람을 의미한다. 로그인하고 나서 유저로 인식될 수 있다.
관리자	유저들을 관리할 수 있는 사람을 의미한다. 교수자를 포함한다. 로그인을 해야 관리자로 인식될 수 있다.
방문자	로그인하지 않은 사람을 의미한다.
사용자	본 서비스를 이용하는 모든 사람들을 말한다. 즉, 유저와 관리자를 모두 합한 범위를 의미한다.
개념 설명 페이지	Python 학습 서비스 중에서 개념을 설명하거나 코드를 주면서 한 줄씩 설명해주는 페이지를 말한다.
연습 문제	한 파트에 해당하는 모든 개념 설명을 마친 뒤, 설명을 잘 이해했는지 확인하는 테스트를 말한다.

멀티 스레드(multi thread)	하나의 프로세스 내에서 둘 이상의 스레드가 동시에 작업을 수행하는 것을 의미한다.
클라이언트(client) / 서버(server)	클라이언트/서버는 두 개의 컴퓨터 프로그램 사이에 이루어지는 역할 관계를 나타내는 것이다. 클라이언트는 다른 프로그램에게 서비스를 요청하는 프로그램이며, 서버는 그 요청에 대해 응답을 해주는 프로그램이다.
소켓	프로그램이 네트워크에서 데이터를 송수신할 수 있도록 "네트워크 환경에 연결할 수 있게 만들어진 연결부."를 의미한다.

Table 1 용어 정의

3.3 Acronyms and Abbreviations

용어	정의
UI	User Interface. 사람(사용자)과 사물 또는 시스템, 특히 기계, 컴퓨터 프로그램 등 사이에서 의사소통을 할 수 있도록 일시적 또는 영구적인 접근을 목적으로 만들어진 물리적, 가상적 매개체를 뜻한다.
DB	Data Base. 여러 사람에 의해 공유되어 사용될 목적으로 통합하여 관리되는 데이터의 집합을 말한다. 본 시스템에서는 MariaDB를 사용한다.
URL	Uniform Resource Locator. 웹 문서의 각종 서비스를 제공하는 서버들에 있는 파일의 위치를 표시하는 표준을 말한다.

Table 2 약어

4. User Requirements Definition

4.1 Objective

User requirement Definition에서는 본 시스템이 제공할 서비스에 대해 사용자의 관점에서 서술한다. 이 때 시스템에 대한 기능적 요구사항과 비기능적 요구사항을 모두 기술한다. 사용자의 이해를 돕기 위해 자연어로 기술하며, 필요에 따라 그림과 도표를 사용한다.

4.2 Functional Requirement

4.2.1 회원 가입

사용자는 서비스를 이용하기 위해 회원 정보를 입력하여 회원 가입을 해야 한다. 회원 정보는 각각의 사용자를 구분하여 식별할 수 있어야 하며, DB에 저장된다. 회원 정보로는 회원 고유 번호, 아이디, 비밀번호, 회원 종류, 이름, 전화번호, 회원이 학습하고 있는 단계, 가입일, 마지막으로 회원 정보를 수정한 수정일이 있다. 아이디는 이메일을 사용한다. DB의 Member 컬럼에 정보를 저장한다.

4.2.2 로그인

사용자가 서비스를 이용하기 위해 아이디와 비밀번호를 입력하여 회원 종류에 맞는 사용 권한을 얻

을 수 있어야 한다. 로그인을 해야지만 사용자를 유저와 관리자로 식별될 수 있다. 로그인 없이는 서비스의 이용이 제한되며 이하로 설명하는 모든 기능은 로그인을 했을 때 정상적으로 동작한다.

4.2.3 메뉴바

화면 맨 왼쪽에 메뉴바가 항상 위치해야 한다. 공지사항과 게시글의 리스트를 보여주는 페이지, python 개념을 설명하는 페이지, 연습문제 페이지, 회원 리스트를 보여주는 페이지 등이 연결되어 있어 원하는 메뉴를 클릭하면 해당 페이지로 연결된다.

4.2.4 회원 정보 보기

사용자가 회원 가입 했을 때 입력했던 정보들을 볼 수 있는 기능이다. DB에서 회원 고유 번호가 일치하는 데이터 전부를 가져와 화면에 보여준다.

4.2.5 회원 정보 수정하기

사용자의 회원 정보를 수정할 수 있는 기능이다. 회원 정보 중에서 수정가능한 항목으로는 회원의 비밀번호와 회원 종류, 이름, 전화번호가 있다. 수정일의 경우에는 따로 입력할 필요없이 해당 시각으로 변경된다. 회원 비밀번호, 이름, 전화번호는 유저나 관리자 모두 수정할 수 있지만 회원 종류 항목에 대해서는 관리자만이 수정 가능하다. 로그인한 사용자만이 이용할 수 있다.

4.2.6 회원 정보 삭제하기

사용자가 본 서비스를 탈퇴하는 기능이다. DB에 저장된 회원의 모든 정보를 삭제한다. 이후 서비스를 이용하기 위해선 다시 회원 가입을 해야한다. 사용자만이 이용할 수 있다.

4.2.7 회원 리스트 보기

관리자만이 이용할 수 있는 기능이다. DB의 Member 컬럼에 저장된 모든 사용자의 리스트를 만들어 화면에 회원 정보 중에서 아이디, 이름, 회원 종류, 등록일을 보여준다. 이름 부분을 클릭하면 선택한 회원의 회원 정보 보기 페이지로 이동한다.

4.2.8 공지사항 작성하기

관리자만이 이용할 수 있는 기능으로, 공지사항 게시판에 올라갈 글을 작성하는 기능이다. 공지사항의 제목과 내용을 입력받고, 추가로 공지사항의 고유 번호와 등록일, 수정일까지 DB의 Notice 컬럼에 저장한다. 만약 공지사항의 제목이나 공지사항의 내용 중에서 둘 중 어느 하나라도 입력된 값이 없다면 DB에 저장하지 않고 입력해달라는 메시지를 띄워준다.

4.2.9 게시글 작성하기

사용자가 이용할 수 있는 기능으로, 자유게시판에 올라갈 글을 작성하는 기능이다. 게시글의 고유 번호, 게시글을 작성한 사용자의 고유 번호, 게시글의 제목, 게시글의 내용, 방문자수, ref, subref, depth, 등록일, 수정일 정보를 DB의 Board 칼럼에 저장한다. 제목과 내용 중에서 둘 중 어느 하나라도 입력된 값이 없다면 DB에 저장하지 않고 입력해달라는 메시지를 띄운다.

4.2.10 공지사항·게시글 보기

작성된 공지사항이나 게시글의 제목과 내용 정보를 화면에 나타내주는 기능이다. 모든 사용자가 볼 수 있으며 공지사항의 경우에는 방문자도 보는 것은 가능하다.

4.2.11 공지사항·게시글 수정하기

작성된 공지사항이나 게시글 정보를 수정하는 기능이다. 제목과 내용을 수정할 수 있으며, 수정일은 입력할 필요없이 해당 시각으로 변경된다. 공지사항의 경우에는 관리자만이 이용가능하고, 게시글의 경우에는 사용자의 고유 번호와 게시글 작성자의 고유 번호가 같을 때만 이용가능하다.

4.2.12 공지사항·게시글 삭제하기

DB에 저장되어 있는 공지사항이나 게시글의 정보를 삭제하는 기능이다. 해당 글의 고유번호가 일치하는 정보를 DB에서 삭제한다. 수정할 때와 마찬가지로 공지사항은 관리자만, 게시글은 사용자의 고유 번호와 게시글 작성자의 고유 번호가 같을 때만 이용가능하다.

4.2.13 게시글 답글 달기

자유게시판에 올려진 게시글의 답글을 달수 있는 기능이다. 게시글 작성하는 페이지를 그대로 이용하지만 작성한 답글 내용 밑에 게시글의 내용이 더해져서 답글의 내용으로 저장된다. 답글의 ref 항목에는 원 게시글의 ref 값이 저장되며, depth 항목에는 원 게시글의 depth값에 1이 더해진 값이 저장된다.

4.2.14 공지사항·게시글 리스트 보기

DB의 Notice와 Board 칼럼에 저장된 글을 각각 리스트로 만들어서 화면에 보여준다. 게시글의 경우에는 depth값에 따라 제목 부분이 들여쓰기한 것처럼 나타난다.

4.2.15 이론 학습하기

학습자가 이론 학습을 할 수 있는 기능이다. 하나의 대단원은 1-1, 1-2와 같이 여러 개의 소단원으로 구성되어 있는데, 한 페이지에 소단원 하나의 내용이 들어가며 다음 버튼을 누르면 다음 소단원 페이지로, 이전 버튼을 누르면 이전 소단원 페이지로 넘어간다.

4.2.16 연습 문제 풀기

파트가 끝났을 때 유저는 연습문제를 풀게 된다. 객관식이나 단답형 문제인 경우에는 보기의 번호 또는 답만 입력받아 정답과 비교하여 일치하면 점수를 얻는다. 주관식의 경우에는 입력받은 답안에 핵심 키워드가 있는지 검사하여 있다면 점수를 얻게된다. 코드가 주어지는 문제도 있는데, 이 경우에는 문제에 따라 결과값을 답으로 쓰는 경우면 정답과 비교하여, 코드를 답으로 쓰는 문제의 경우 전체 코드를 실행시켰을 때의 결과를 정답과 비교하여 일치하면 점수를 얻는다. 기준 점수 이상이 되면 다음 파트로 넘어가는 버튼이 활성화되면서 사용자의 학습 단계에 다음 파트가 저장된다.

4.3 Non-functional Requirement

4.3.1 Product Requirement

4.3.1.1 Usability Requirement (UI)

직관적이고 이해하기 쉬운 UI를 갖추어야 한다. 실습을 위한 교육 자료가 입력을 위해 제공되는 코딩 창과 붙어있어, 자료를 보며 입력할 수 있게 한다.

4.3.1.2 Efficiency Requirement

학습을 마무리한 후 학습에 대한 평가가 즉각적으로 이루어져야 한다. 학습 중 진행한 문제 풀이와 실습을 통해 작성한 코드의 평가가 종합적으로 점수에 반영되어야 하며, 이를 제외한 모든 페이지의 응답 속도 또한 느리지 않아야 한다.

4.3.1.3 Dependability Requirement

회원 가입 시 입력한 사용자의 데이터가 제대로 저장되어야 한다. 올바른 사용자 정보를 입력했을 경우 로그인을 성공할 수 있어야 한다. 문제 풀이 후 제출된 코드는 손상되지 않아야 하며, 정답 여부를 체크한다. 정답일 경우, 푼 문제로 기록되고, 이는 점수를 측정할 때 제공한다.

4.3.1.4 Security Requirement

사용자의 개인정보는 암호화되어 보관해야 한다. 사용자의 비밀번호는 로그인 시에 사용자를 식별하는 데만 사용된다. 외부 접근으로 인해 사용자의 정보가 유출되는 일이 없어야 한다.

4.3.2 Organization Requirement

4.3.2.1 Environmental Requirement

본 서비스는 온라인 웹 서비스이다. PC와 모바일 모두 이용 가능하긴 하나, 주 사용 환경은 PC를 전제로 한다.

4.3.2.2 Operational Requirement

Spring boot, Eclipse, Tomcat server, MariaDB를 이용하여 개발한다. 개발 언어로는 Java, JavaScript, Ajax를 사용하고, Bootstrap과 Ace Editor, Anaconda를 개발에 이용한다

4.3.2.3 Development Requirement

사용자의 정보, 공지사항 정보, 게시글 정보가 데이터베이스에 저장된다. 사용자의 고유 번호를 세션에 계속 띄워둬서 여러 페이지를 이동해도 해당 정보를 항상 가지고 있도록 한다. 데이터베이스에서 정보를 가져올 때는 해당 칼럼의 고유 번호를 사용하여 가져오도록 한다

4.3.3 External Requirement

4.3.3.1 Regulatory Requirement / Legislative Requirement

사용자의 정보의 공개 범위는 본인과 관리자까지이다. 사용자의 회원 정보 중에서 비밀번호는 본인만 확인할 수 있다. 회원 종류는 관리자만이 수정할 수 있다.

공지사항 게시판에는 관리자만이 작성, 수정, 삭제가 가능하다. 유저의 경우에는 오로지 공지사항 내용을 보는 것만이 가능하다. 자유게시판에는 모든 사용자가 게시글 작성과 보는 것이 가능하다. 수정 및 삭제는 본인이나 관리자만이 가능하다.

4.3.3.2 Ethical Requirement

본 서비스는 연령 제한이 없다.

4.3.3.3 Safety / Security Requirement

사용자의 회원 정보 중에서 비밀번호는 데이터베이스에 저장될 때는 암호화해서, 회원 정보 보기 페이지에 나타낼 때는 복호화해서 나타낸다.

5. System Architecture

5.1 Objective

System Architecture에서는 본 시스템의 구조에 대해 높은 수준의 추상도로 설명한다. 시스템을 구성하는 컴포넌트와 이들의 상호작용을 기술한다. 서술의 이해도를 높이기 위해 시각화를 이용한다. 개발자가 시스템의 전체적인 구조를 이해하는 데 도움을 준다.

5.2 System Architecture

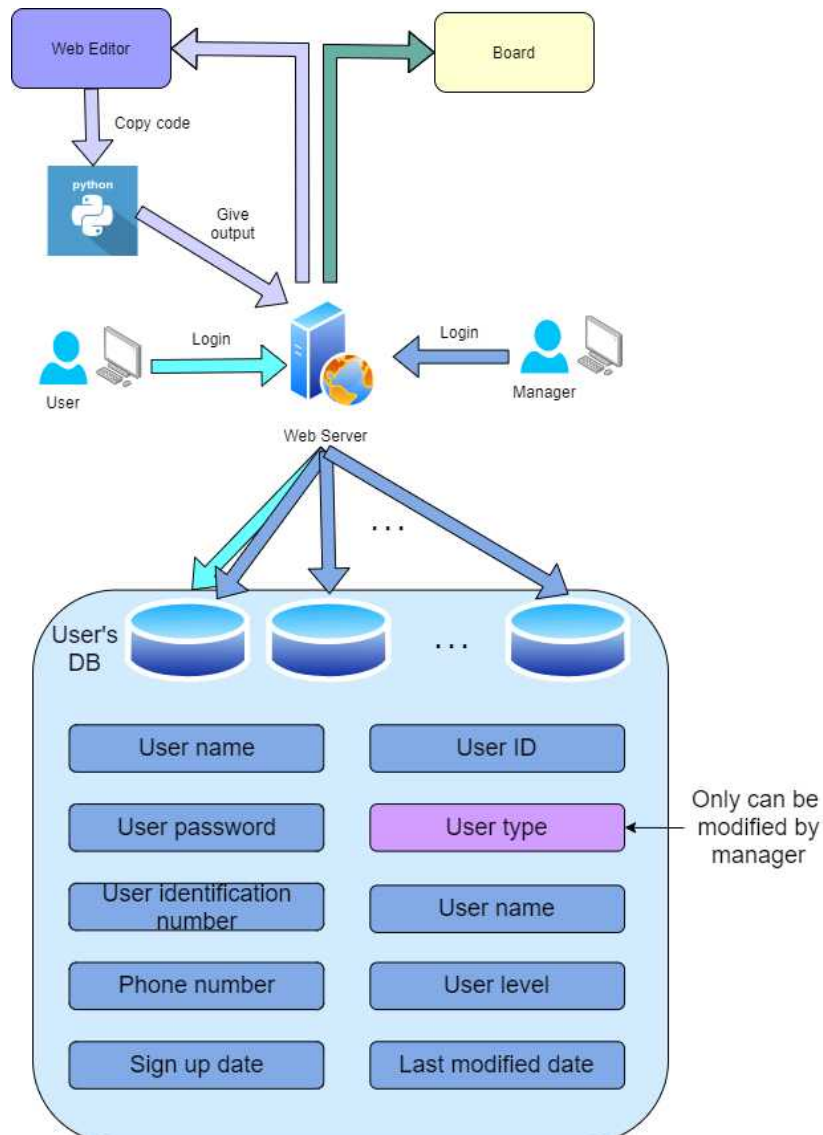


Diagram 1 System Architecture: 전체

전체 시스템 구조는 크게 Python Editor(Web Editor, Python), Board, Web Server, DB(User's DB, User name, ... , Last modified data)로 구성된다.

5.3 Sub system

5.3.1 사용자 시스템

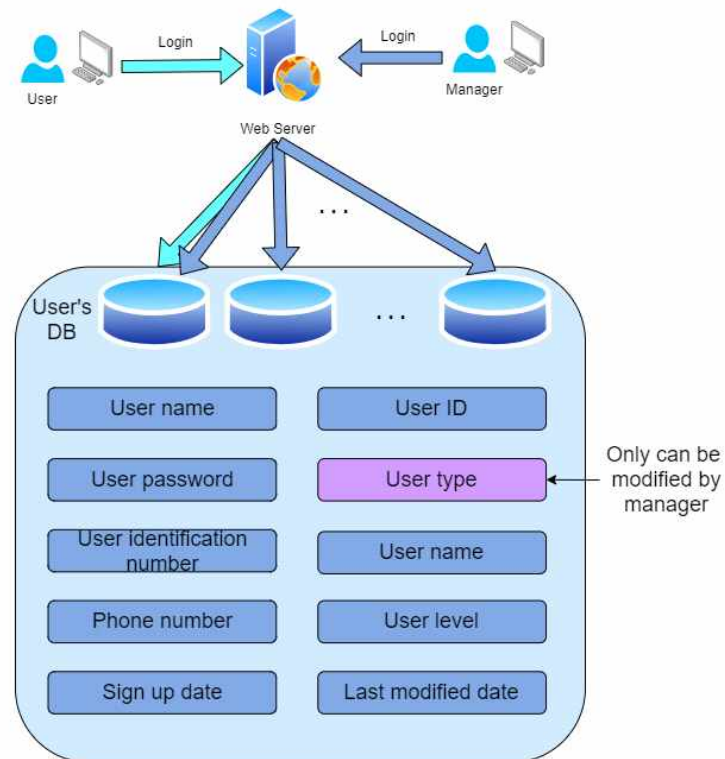


Diagram 2 System Architecture: 사용자 시스템

사용자 시스템은 사용자가 회원 가입부터 로그인, 로그아웃 등 전반적인 사용자 관련 요구사항을 담당하는 시스템이다.

5.3.2 Python & Web Editor

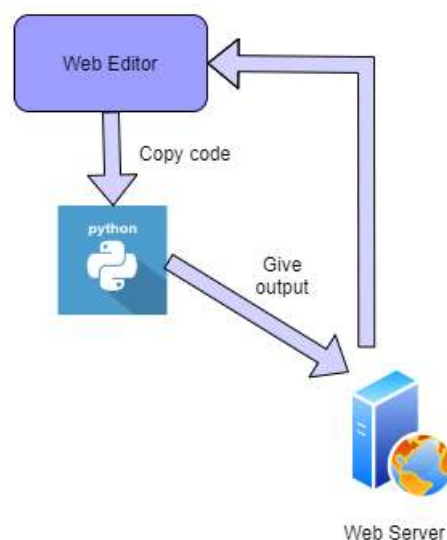


Diagram 3 System Architecture: Python & Web Editor

Python & Web Editor는 사용자가 연습 문제를 풀기 위한 파이썬 코드 작성과 관련된 시스템이다.

5.3.3 게시판 시스템

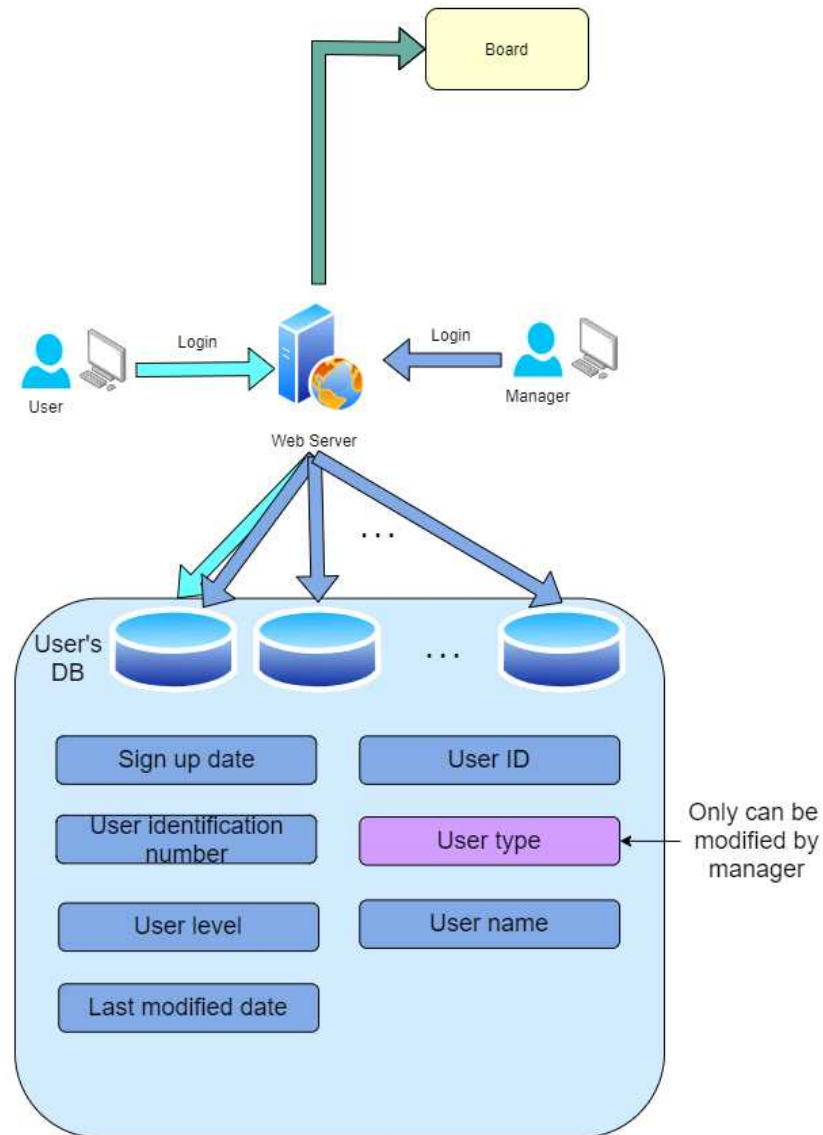


Diagram 4 System Architecture: 게시판 시스템

게시판 시스템은 학습자와 관리자(교수자)가 사용하는 게시판과 관련된 모든 요구사항을 담당하는 시스템이다.

6. System Requirements Specification

6.1 Objective

System Requirement Specification에서는 시스템을 구성하는 각 기능들의 기능적 요구사항과 비기능적 요구사항을 기술적인 관점에서 상세히 서술한다. 구조화된 문서 양식을 통해 시스템 내 기능적 부분을 명확하게 설명한다.

6.2 Functional Requirement

6.2.1 회원 가입

항목	설명
기능	회원 가입
설명	회원 가입을 위해 사용자의 정보를 입력받고, 데이터베이스에 입력받은 정보들을 저장한다.
입력	아이디, 비밀번호, 이름, 전화번호
출력	하나라도 값이 입력되지 않은 항목이 있다면 해당 항목을 입력하라는 메시지를 출력한다. 정상적으로 동작했다면 홈으로 이동한다. 데이터베이스에 저장하는 과정에서 오류가 발생했다면 해당 오류메세지를 출력한다.
처리	입력받은 정보 외에도 회원 종류를 "C"로, 회원 학습 단계를 0으로 설정하여 데이터베이스에 회원 정보 저장한다. 회원 고유 번호는 데이터베이스에 정보가 저장될 때 자동으로 잡아준다.
조건	아이디는 이메일로 받는다. 비밀번호는 암호화하여 데이터베이스에 저장한다.

Table 3 회원 가입

6.2.2 로그인

항목	설명
기능	로그인
설명	로그인을 위해서 사용자의 정보를 입력 받고, 데이터 베이스와 정보가 일치하는 지 확인하여 로그인 한다.
입력	아이디, 비밀번호
출력	아이디나 비밀번호가 틀렸을 경우 아이디나 비밀번호가 틀렸다는 메시지를 출력한다. 정상적으로 로그인에 성공했다면 홈으로 이동한다.
처리	로그인에 성공하면 학습단계를 불러와서 그 부분부터 학습할 수 있게 한다.
조건	아이디는 이메일로 받는다.

Table 4 로그인

6.2.3 이론 학습하기

항목	설명
기능	이론 학습하기
설명	학습자의 학습 단계를 데이터베이스에서 불러와 해당 챕터의 이론 내용을 학습할 수 있게 하는 기능이다.
입력	학습자의 학습 단계
출력	학습 단계에 맞는 이론 내용이 화면에 나타난다.
처리	다음 버튼을 누르면 다음 소단원 내용이, 이전 버튼을 누르면 이전 소단원 내용이 나온다. 마지막 소단원

	페이지에는 다음 버튼 대신 연습문제 버튼이 나타난다.
조건	한 페이지에 소단원 하나의 내용이 들어간다. 이전 챕터를 수강한 후에 다음 챕터를 학습할 수 있다.

Table 5 이론 학습하기

6.2.4 연습 문제 풀기

항목	설명
기능	연습 문제 풀기
설명	학습자의 학습 단계에 맞는 연습 문제를 데이터베이스에서 불러와 화면에 띄운다. 답안 입력을 위해 객관식 문제에는 체크 박스를, 단답형이나 주관식 문제에는 텍스트 상자를 사용한다. 코드가 주어지는 문제의 경우엔 화면에 editor가 나타나며, 코드를 작성해야 하는 문제라면 뼈대 코드를 학습자에게 제시한다.
입력	학습자의 학습 단계, 학습 단계에 맞는 연습 문제
출력	학습자의 답안을 채점 기능으로 전달한다.
처리	한 화면에 하나의 문제가 제시되며, 다음 버튼을 누르면 다음 문제로 넘어가고 이전 버튼을 누르면 이전 문제로 돌아간다. 마지막 문제 페이지에는 다음 버튼 대신 제출 버튼이 나타난다. 학습자가 다음 버튼을 누르면 문제의 번호와 답안을 저장해 두었다가, 제출 버튼을 누르면 채점 기능으로 넘긴다.
조건	이론 학습을 마친 후에 연습문제를 풀 수 있다. 사용자가 답안을 입력하지 않은 문제가 있어도 제출은 가능하나, 입력하지 않은 문항이 있음을 알리는 안내창을 띄운다.

Table 6 연습 문제 풀기

6.2.5 채점 결과 확인하기

항목	설명
기능	채점 결과 확인하기
설명	학습자의 답안을 받아 채점하여 학습자에게 결과를 보여주는 기능이다. 한 화면에 각 문제의 번호와 정답 여부가 나타나며, 기준 점수 이상이 되면 다음 파트로 넘어갈 수 있다.
입력	학습자가 입력한 연습문제 답안
출력	문제의 번호와 정답 여부, 점수
처리	객관식이나 단답형 문제의 경우, 학습자의 입력과 정답이 일치하면 점수를 얻는다. 주관식의 경우 입력 받은 답안에 핵심 키워드가 있다면 점수를 얻는다. 코드를 작성하는 문제에선 학습자의 코드를 실행했을

	때의 결과가 정답과 일치하면 점수를 얻는다. 기준 점수 이상이면 학습자의 학습 단계에 다음 파트가 저장된다.
조건	기준 점수 이상이 되어야 다음 파트로 넘어가는 버튼이 활성화된다.

Table 7 채점 결과 확인하기

6.3 Non-functional Requirement

6.3.1 Product Requirement

6.3.1.1 Usability Requirement (UI)

직관적이고 이해하기 쉬운 UI를 갖추어야 한다. 문제의 경우 교육 자료의 다음 페이지에서 따로 제공하며 입력 문제일 경우, 한 문제씩 제공하여 집중할 수 있도록 한다. 정답을 입력하는 UI는 가독성을 위해 배경색은 흰색, 코드는 검은색을 기본으로 하며 syntax highlighting을 지원해야 한다. 사용자의 점수를 보여줄 때는 학습한 챕터를 테이블로 보여주며 각각의 점수를 같이 제공한다.

6.3.1.2 Efficiency Requirement

학습을 마무리한 후 학습에 대한 평가가 즉각적으로 이루어져야 하며 보여지는데 걸리는 시간이 4초가 넘지 않아야 한다. 이때 학습 중 진행한 문제 풀이와 실습을 통해 작성한 코드의 평가가 종합적으로 점수에 반영되어야 하며, 이를 제외한 모든 페이지의 응답 속도 또한 3sec를 넘기지 않아야 한다.

6.3.1.3 Dependability Requirement

회원 가입시 입력한 사용자의 데이터가 저장되는 과정에서 변질되지 않아야 한다. 올바른 사용자 정보를 입력했을 경우 로그인을 성공할 수 있어야 한다. 문제 풀이 후 제출된 코드는 손상되지 않아야 하며, 정해진 test case를 통과하는지 확인 가능해야 한다. 모든 test case를 통과할 경우, 푼 문제로 기록되고 채점 데이터로 사용이 되어야 한다.

6.3.1.4 Security Requirement

사용자의 개인정보는 암호화되어 보관해야 한다. 사용자의 비밀번호는 관리자 측에서도 정확하게 알 수 없으며, 로그인 시에 사용자를 식별하는 데만 사용된다. 사용자의 개인정보는 다른 사용자에게 노출되지 않아야 한다. 외부 접근으로 인해 사용자의 정보가 유출되는 일이 없어야 한다.

6.3.2 Organization Requirement

6.3.2.1 Environmental Requirement

본 서비스는 온라인 웹 서비스이며, PC와 모바일 모두 이용할 수 있다. 사용자는 인터넷이 연결된 상태에서 사용하여야 한다. 크롬, 익스플로러에서 사용할 수 있다. 사용자의 사용 환경에 관계없이 이용에 불편이 없도록 하는 것이 목표이나, 주 사용 환경은 PC를 전제로 한다. 개발 시 우선순위로 PC 사용자를 우선순위로 둔다.

6.3.2.2 Operational Requirement

Spring boot, Eclipse, Tomcat server, MariaDB를 이용하여 개발한다. 개발 언어로는 Java, JavaScript, Ajax를 사용하고, Bootstrap과 Ace Editor, Anaconda를 개발에 이용한다

6.3.2.3 Development Requirement

사용자의 문제 풀이 데이터와 사용자 정보는 데이터베이스에 저장된다. 여러 페이지를 이동하는 동안 정보가 기억될 수 있다. 서비스 특성상 PC 사용자를 우선순위를 두고 개발하나, 모바일 환경에서도 이용할 수 있도록 한다.

6.3.3 External Requirement

6.3.3.1 Regulatory Requirement / Legislative Requirement

사용자 정보의 공개 범위는 사용자 본인과 관리자까지이며, 비밀번호의 경우 사용자 본인만 알 수 있다. 사용자 정보를 외부에 제공할 경우, 사용자 본인에게 정보 제공 동의를 받아야 한다. 사용자가 자신의 정보를 삭제하기를 원하는 경우, 관리자에게 문의하여 삭제될 수 있으나 향후 서비스 이용이 어려울 수 있다.

6.3.3.2 Ethical Requirement

본 서비스는 이용 연령 제한이 없다.

6.3.3.3 Safety / Security Requirement

개인정보보호법에 따라 고객 개인정보를 안전하게 보관하고 보호하여야 한다.

6.4 Scenario

6.4.1 회원 가입

6.4.1.1 로그인

사용자는 회원 가입을 한 계정의 아이디와 비밀번호를 기입한다. 만약 아이디와 비밀번호가 데이터 베이스에 저장된 정보와 일치할 경우 본인의 계정에 로그인 되어 학습 진행정보가 불러와지고 홈화 면으로 넘어간다. 아이디와 비밀번호가 데이터베이스와 일치하지 않을 경우 오류메시지가 나오며 로그인 되지 않는다.

6.4.2 이론 학습하기

6.4.2.1 Initial Assumption

학습자는 로그인에 성공했으며, 데이터베이스로부터 학습자의 학습 단계를 불러온 상태이다.

6.4.2.2 Normal Flow of Events

학습자의 학습 단계에 맞는 이론 내용이 화면에 나타난다. 한 화면에 소단원 하나의 내용이 나타난다. 학습자는 소단원 하나의 내용을 학습한 후 다음 버튼을 눌러 다음 소단원 내용을 학습한다. 이전

버튼을 누르면 이전 소단원 페이지로 돌아간다.

6.4.2.3 What can go wrong

이전 챕터를 학습하지 않았다면 다음 챕터를 학습할 수 없도록 한다.

6.4.2.4 System State on Completion

학습자가 이론 학습을 마치면 연습문제 버튼을 눌러 연습문제를 푼다.

6.4.3 연습문제 풀기

6.4.3.1 Initial Assumption

학습자는 이론 학습을 마친 상태이다.

6.4.3.2 Normal Flow of Events

학습자의 학습 단계를 파악하여, 학습 단계에 맞는 연습 문제를 제시한다.

연습 문제의 종류에는 객관식, 단답형, 주관식이 있으며 코드가 주어지는 문제도 있다. 객관식 문제의 경우엔 학습자가 정답이라고 생각하는 선지를 선택한다. 단답형이나 주관식 문제는 텍스트 상자의 답안을 입력한다. 주어진 코드를 보고 결과값을 작성하는 문제의 경우 텍스트 상자에 답안을 입력하며, 코드를 작성하는 문제의 경우 editor에 뼈대 코드가 제시된다.

한 화면에 하나의 문제가 제시되며 다음 버튼을 누르면 다음 문제로 넘어간다. 마지막 문제 화면에는 제출 버튼이 있다.

6.4.3.3 What can go wrong

이론 학습을 마친 후에야 연습문제를 풀 수 있도록 한다.

답안 작성이 안 된 문제가 있어도 제출을 할 수는 있으나, 답안이 입력되지 않은 문제가 있음을 학습자에게 알린다.

6.4.3.4 System State on Completion

학습자가 제출 버튼을 누르면 학습자가 작성한 답안이 채점 기능으로 전달된다.

6.4.4 채점 결과 확인하기

6.4.4.1 Initial Assumption

학습자가 연습문제의 답안을 제출한 상태이다.

6.4.4.2 Normal Flow of Events

학습자가 제출한 답안을 채점한다. 객관식이나 단답형 문제는 학습자의 답안과 정답이 일치하면 점수를 얻으며, 주관식 문제는 학습자의 답안에 핵심 키워드가 있을 경우 점수를 얻는다. 코드를 작성

하는 문제에선 학습자의 코드를 실행했을 때의 결과가 정답과 일치하면 점수를 얻는다.
한 화면에 각 문제의 번호와 정답 여부가 나타난다.

6.4.4.3 What can go wrong

채점 결과 기준 점수를 넘지 못했을 경우, 다음 단원으로 넘어갈 수 없으며 연습 문제 풀이를 재시도해야 한다.

6.4.4.4 System State on Completion

학습자가 기준 점수 이상을 획득하면 다음 파트로 넘어갈 수 있으며, 다음 학습 단계가 학습자의 학습 단계에 저장된다.

7. System Models

7.1 Objective

System Requirement Model에서는 Context, Interaction Models, Sequence Diagram, Structural Models, Behavioral Models과 같이 다양한 모델을 활용하여 시스템의 기능과 각 컴포넌트 간의 관계와 상호작용에 대해 설명한다. System Requirement Model을 통해서 시스템에 대한 이해도를 높일 수 있다.

7.2 Context Models

7.2.1 Context Model

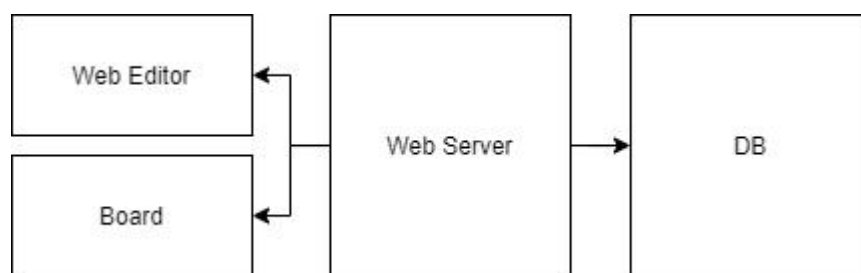


Diagram 5 Context Model

7.2.2 Process Model

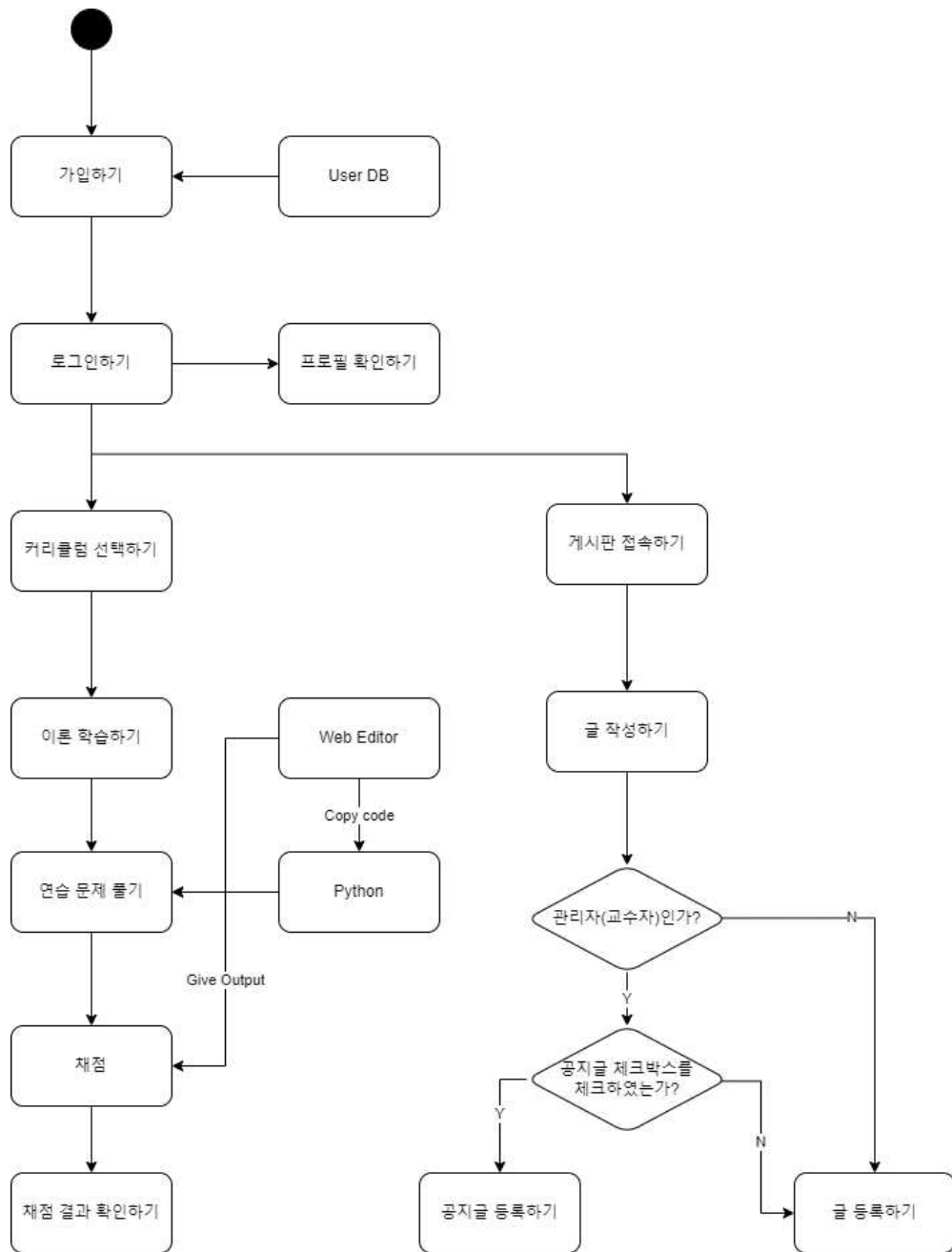


Diagram 6 Process Model

7.3 Interaction Models

7.3.1 Use Case Model

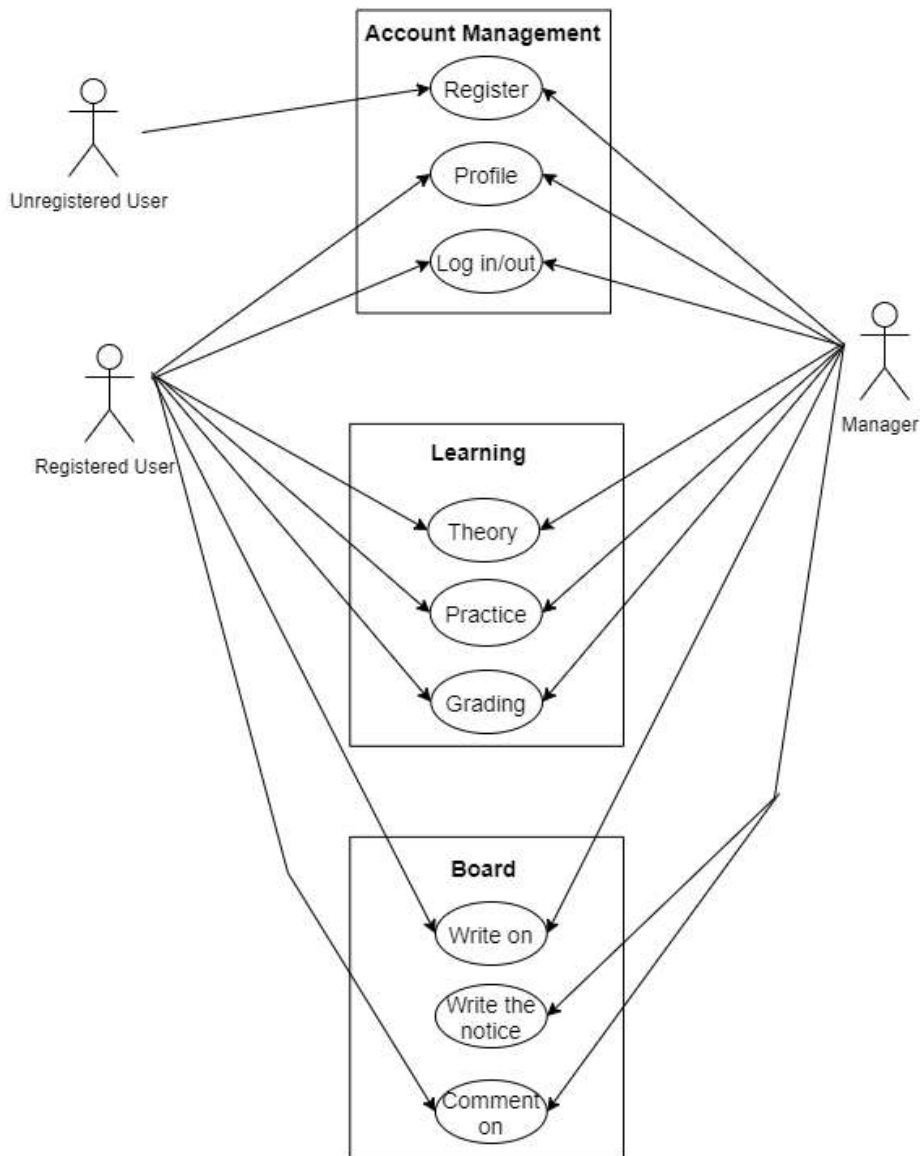


Diagram 7 Use Case Model

7.3.2 Tabular Description

7.3.2.1 회원 가입

USE CASE	Register(회원 가입)
ACTOR	학습자(미가입 사용자), 관리자(교수자)
DESCRIPTION	학습자와 관리자가 시스템에 자신의 개인정보를 입력하여 회원 가입한다. 학습자와 관리자가 입력한 사용자 정보를 데이터베이스에 저장한다.
STIMULUS	학습자와 관리자가 사용자 정보를 입력하고 회원 가입 버튼을 누른다.
RESPONSE	사용자의 회원 가입 정보를 검증한 뒤 데이터베이스에 사용자 정보를 등록한다.

Table 8 Tabular Description: Register

7.3.2.2 프로필 확인

USE CASE	Profile(프로필 확인)
ACTOR	학습자(가입한 사용자), 관리자(교수자)
DESCRIPTION	로그인 후 프로필 버튼을 눌러 사용자의 정보를 확인하는 프로필로 드려간다. 자신의 커리큘럼 진행도와 게시판에 작성한 글들을 볼 수 있다.
STIMULUS	사용자가 로그인한 후 프로필 버튼을 눌러 들어간다.
RESPONSE	프로필 화면이 보인다.

Table 9 Tabular Description: Profile

7.3.2.3 로그인/로그아웃

USE CASE	Log in/out(로그인/로그아웃)
ACTOR	학습자, 관리자(교수자)
DESCRIPTION	사용자가 서비스 이용을 위해 이메일과 비밀번호를 입력하고 사용자 권한을 얻는다. 이후 사용자가 로그아웃을 원할 때 로그아웃 버튼을 눌러 로그아웃한다.
STIMULUS	사용자가 이메일과 비밀번호를 입력하고 로그인 버튼을 누른다. 사용자가 로그아웃 버튼을 누른다.
RESPONSE	사용자의 로그인 정보와 데이터베이스에 저장된 사용자 정보를 대조하여 로그인 결과와 로그아웃을 반환한다.

Table 10 Tabular Description: Log in/out

7.3.2.4 이론 학습하기

USE CASE	Theory(이론 학습하기)
ACTOR	학습자(가입한 사용자)
DESCRIPTION	학습자의 학습 단계에 맞는 이론 내용을 불러와 학습자가 학습할 수 있도록 한다. 한 단원의 이론 학습을 마쳐야 연습 문제를 풀 수 있다.
STIMULUS	사용자는 각 소단원 내용을 학습하고 다음 버튼을 누른다.
RESPONSE	한 단원의 이론 학습을 마치면 연습 문제 버튼을 눌러 연습 문제 풀이로 넘어간다.

Table 11 Tabular Description: Theory

7.3.2.5 연습 문제 풀기

USE CASE	Practice(연습 문제 풀기)
ACTOR	학습자(가입한 사용자)
DESCRIPTION	학습자의 학습 단계에 맞는 연습 문제를 한 화면에 하나씩 제시한다. 객관식, 단답형, 주관식, 코드가 주어지는 문제가 있다. 답안 입력을 위해 객관식 문제는 체크 박스가, 단답형 및 주관식 문제에는 텍스트 상자가 주어진다. 코드를 작성해야 하는 문제의 경우 뼈대 코드가 미리 주어진다. 답안을 작성하지 않아도 제출할 수 있으나, 작성되지 않은 답안이 있음을 학습자에게 알린다.
STIMULUS	학습자는 문제 유형에 맞게 답안을 작성하고, 다음 버튼을 눌러 다음 문제로 넘어간다.
RESPONSE	학습자가 답안을 작성하고 제출 버튼을 누르면, 작성한 답안이 채점 결과 기능으로 전달된다.

Table 12 Tabular Description: Practice

7.3.2.6 채점하기

USE CASE	Grading(채점 결과 확인하기)
ACTOR	학습자(가입한 사용자), 관리자(교수자)
DESCRIPTION	학습자가 제출한 답안을 채점하여 제시한다. 한 화면에 각 문제의 번호와 정답 여부가 나타난다. 객관식 및 단답형 문제는 학습자의 답안과 정답이 일치할 경우에, 주관식 문제는 핵심 키워드가 답안에 포함되어 있을 경우에, 코드를 짜는 문제는 해당 코드를 실행했을 때의 결과값이 정답과 일치할 경우에 정답으로 인정된다.
STIMULUS	학습자가 연습문제를 제출한 후 채점 결과를 확인한다.
RESPONSE	학습자와 관리자는 문제 별 정답 여부와 점수를 확인할 수 있다. 기준 점수 이상이 되면 다음 파트로 넘어가는 버튼이 활성화된다.

Table 13 Tabular Description: Grading

7.3.2.7 글 작성하기

USE CASE	Write on(글 작성하기)
ACTOR	학습자(가입한 사용자), 관리자(교수자)
DESCRIPTION	사용자가 게시판에 접속하여 자신의 글을 작성한다. 게시판의 작성자는 사용자를 특정할 수 있는 사용자 이름(아이디) 또는 이메일 등으로 표시되며, 사용자가 글쓰기 창에 글을 작성하고 등록하면 이후 권한이 있는 사용자 모두가 글을 읽을 수 있다.
STIMULUS	사용자가 게시판에 접속하여 글쓰기 버튼을 클릭한다. 글을 작성하면 등록하기 버튼을 클릭한다.
RESPONSE	글쓰기 버튼을 클릭하면 게시판에 글을 작성할 수 있는 창이 나타나고, 글을 작성한 후 등록하기 버튼을 클릭하면 권한이 있는 사용자 모두가 읽을 수 있는 형태로 게시판에 글이 등록된다.

Table 14 Tabular Description: Write on

7.3.2.8 공지 작성하기

USE CASE	Write the notice(공지 작성하기)
ACTOR	학습자(가입한 사용자), 관리자(교수자)
DESCRIPTION	관리자(교수자)가 게시판에 접속하여 공지사항을 작성한다. 게시판의 작성자는 관리자(교수자)의 사용자 이름(아이디) 또는 이메일 등으로 표시되며, 관리자(교수자)가 글쓰기 창에 공지사항을 작성하고 등록하면 이후 권한이 있는 사용자 모두가 공지사항을 읽을 수 있다.
STIMULUS	관리자(교수자)가 게시판에 접속하여 글쓰기 버튼을 클릭한다. 공지사항을 작성한 후 공지사항로 등록하기 체크박스를 체크한다. 체크한 후 등록하기 버튼을 클릭한다.
RESPONSE	글쓰기 버튼을 클릭하면 게시판에 글을 작성할 수 있는 창이 나타나고, 글을 작성한 후 공지사항로 등록하기 체크박스를 체크하여 등록하기 버튼을 클릭하면 권한이 0있는 사용자 모두가 읽을 수 있는 형태로 게시판에 공지사항이 등록된다.

Table 15 Tabular Description: Write the notice

7.3.2.9 답글 작성하기

USE CASE	Comment on(답글 작성하기)
ACTOR	학습자(가입한 사용자), 관리자(교수자)
DESCRIPTION	사용자가 게시판에 접속하고 게시판 글에 접속하면 자신의 답글을 작성한다. 답글의 작성자는 사용자를 특정할 수 있는 사용자 이름(아이디) 또는 이메일 등으로 표시되며, 사용자가 답글 창에 답글을 작성하고 등록하면 이후 권한이 있는 사용자 모두가 답글을 읽을 수 있다.
STIMULUS	사용자가 게시판에 접속하여 글에 들어가면 답글 창에 답글을 입력하고 등록하기 버튼을 클릭한다.
RESPONSE	게시판의 글을 클릭하면 글을 읽을 수 있으며, 글 아래 답글 창에 글을 작성하고 등록하기 버튼을 클릭하면 권한이 있는 사용자 모두가 읽을 수 있는 형태로 답글이 등록된다.

Table 16 Tabular Description: Comment on

7.4 Behavioral Models

7.4.1 Event-driven Diagram

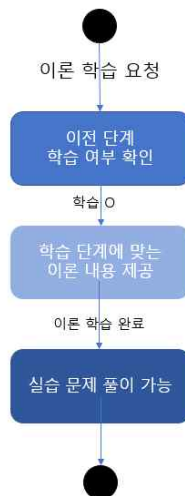


Diagram 8 Event-driven Diagram: 이론 학습

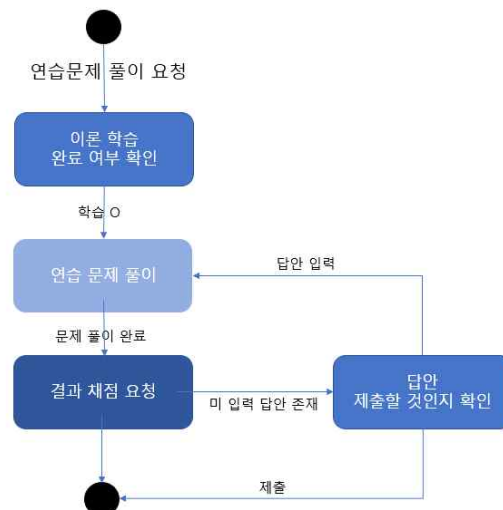


Diagram 9 Event-driven Diagram: 연습문제 풀기

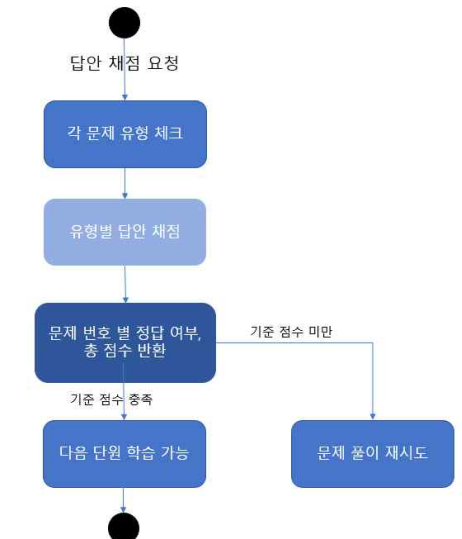


Diagram 10 Event-driven Diagram: 채점결과 확인

8. System Evolution

8.1 Objective

해당 장에서는 Qurious의 assumption과 이에 따라 발생하는 한계점을 설명한다. 또한 비즈니스 환경의 변화, 플랫폼의 변화, 사용자의 요구사항에 변화에 따라 어떻게 evolution하고 한계점을 보완할 수 있을지 서술한다.

8.2 Limitation

8.2.1 Output이 확인 불가능한 경우

완성된 실시간 채팅 프로그램이나 멀티스레드, 소켓통신 럼 단계를 순차적으로 실행했을 때 특정 Output이 없는 경우는 그것을 데이터베이스와 비교해서 정답처리를 할 방법이 없다. 따라서 예시화면과 같으면 다음 교육으로 넘어가는 수밖에 없다.

8.2.2 국소적인 내용

본 교육 프로그램은 오로지 실시간 채팅을 구현하는 것을 목적으로 만들어졌기 때문에 파이썬을 기초부터 차근차근 학습하고자 하는 학생에게는 맞지 않을 수 있다.

8.2.3 아이디나 비밀번호 찾기 불가능

회원정보 수정이 로그인을 했을 경우에만 가능하기 때문에 아이디나 비밀번호를 잊어버리면 로그인을 할 수가 없다.

8.3 System Evolution

8.3.1 Evolution for Practice checking

완성된 실시간 채팅 프로그램이나 멀티스레드, 소켓통신 럼 단계를 순차적으로 실행했을 때 특정 Output을 저장한 것을 데이터베이스와 비교하여 정답처리를 하거나 피드백을 할 수 있도록 해야 한

다. 이는 학생 스스로가 예시화면을 통한 판단 없이도 프로그램 내에서 자체적으로 제공하는 피드백으로 학습 과정을 판단할 수 있도록 해야 한다.

8.3.2 Evolution for curriculum

본 교육 프로그램이 현재 제공하고 있는 실시간 채팅 구현 커리큘럼 외에도 파이썬 기초 커리큘럼, 파이썬 활용 게임 개발, 그리고 자바, C++ 등의 커리큘럼들을 포함하여 기초와 활용까지 학습할 수 있도록 해야 한다.

8.3.3 Evolution for User Database

회원 정보 수정의 용이를 위한 아이디/비밀번호 찾기, 사용자 프로필 및 개인정보 보안 향상 등을 유도해야 한다.

8.3.4 Evolution for Board system

사용자가 게시판을 사용할 때 특정 규칙을 적용하여 프로그램의 면학 분위기를 유지해야 한다.

9. Appendices

9.1 Objective

Appendices에서는 개발되는 시스템에 사용할 구체적이고 자세한 정보를 제공한다.

9.2 Database Requirements

본 시스템에서 사용하는 데이터베이스의 요구사항을 서술한다. 각 요구사항에 *Attribute*는 *이탤릭체*, **Entity type**은 **굵은 글씨체**, Constraint는 밑줄로 표시한다.

본 시스템에서 사용하는 데이터베이스는 **사용자 데이터베이스**, **연습 문제 데이터베이스**, **게시판 데이터베이스**이다. 다음에서 설명하는 내용들은 해당되는 요구사항에 맞는 형태로 데이터베이스 테이블에 저장되어야 한다.

사용자 데이터베이스는 username을 필수적으로 가져야 하고, *email* 등을 추가적으로 가질 수 있다.

연습 문제 데이터베이스는 id를 필수적으로 가져야 하고, 추가적으로 *tittle*, *description*, *input*, *output*, *test_case_id*, *difficulty*를 attribute로 가질 수 있다.

게시판 데이터베이스는 id를 필수적으로 가져야 하고, 추가적으로 *tittle*, *contents*, *importance*를 attribute로 가질 수 있다.

9.3 Database Detail

사용자 데이터베이스는 해당 서비스에 사용자가 가입할 때 입력되며, **연습 문제 데이터베이스**, **게시판 데이터베이스**에 참조된다. **연습 문제 데이터베이스**는 사용자가 연습 문제를 제출하면 제출 결과를 저장한다. 저장된 제출 결과에 따라 사용자 데이터베이스 *level* 속성이 변경될 수 있다. **게시판 데**

이터베이스는 사용자 데이터베이스를 참조하여 사용자가 작성한 게시글의 데이터를 저장한다.

9.4 Login Details

로그인에는 아이디와 비밀번호가 필요하다.

아이디는 다음과 같은 정규표현식을 만족해야 한다.

"^[0-9a-zA-Z]([-_]?[0-9a-zA-Z])*[0-9a-zA-Z]([-_]?[0-9a-zA-Z])*[a-zA-Z]{2,3}\$"

10. Index

10.1 Objective

Index에서는 본 문서에 사용된 용어와 다이어그램, 기능에 대한 인덱스를 나타낸다.

10.2 Table Index

Table 1 용어 정의.....	08
Table 2 약어.....	08
Table 3 회원 가입.....	00
Table 4 로그인.....	00
Table 5 이론 학습하기.....	00
Table 6 연습 문제 풀기.....	00
Table 7 채점 결과 확인하기.....	00
Table 8 Tabular Description: Register.....	00
Table 9 Tabular Description: Profile.....	00
Table 10 Tabular Description: Log in/out.....	00
Table 11 Tabular Description: Theory.....	00
Table 12 Tabular Description: Practice.....	00
Table 13 Tabular Description: Grading.....	00
Table 14 Tabular Description: Write on.....	00
Table 15 Tabular Description: Write the notice.....	00
Table 16 Tabular Description: Comment on.....	00

10.3 Figure Index

Figure 1 '06년과 '16년의 글로벌 기업 시가 총액 비교.....	05
Figure 2 한국의 산업별 SW 활용도 및 선진국 대비 수준.....	05

10.4 Diagram Index

Diagram 1 System Architecture: 전체.....	12
Diagram 2 System Architecture: 사용자 시스템.....	13

Diagram 3 System Architecture: Python & Web Editor.....	13
Diagram 4 System Architecture: 게시판 시스템.....	14
Diagram 5 Context Model.....	20
Diagram 6 Process Model.....	21
Diagram 7 Use Case Model.....	22
Diagram 8 Event-driven Diagram: 이론 학습.....	25
Diagram 9 Event-driven Diagram: 연습문제 풀기.....	25
Diagram 10 Event-driven Diagram: 채점결과 확인.....	26

11. Reference

- "SW교육 도입의 배경과 현황: 디지털 세대를 위한 미래교육 - 김현철 고려대학교 컴퓨터학과 및 컴퓨터교육과 교수 한국 컴퓨터교육학회 회장