

# PACEMAKER

## SOFTWARE REQUIREMENTS SPECIFICATION

곽신우 (201411260)

문성찬 (201511260)

배윤희 (201511266)

주재빈 (201511298)

## 목차

---

1.	31.1.	31.2.
	31.3.	32.
	32.1.	32.2.
	52.3.	62.4.
	62.5.	63.
	73.1.	73.2.
	73.3.	83.4.
	83.5.	93.6.
	93.7.	103.8.
	103.9.	113.10.
	113.11.	123.12.
	123.13.	123.14.
	133.15.	133.16.
	133.17.	143.18.
	143.19.	153.20.
	153.21.	163.22.
	163.23.	173.24.
	183.25.	183.26.
	193.27.	193.28.
	203.29.	203.30.
	213.31.	213.32.
	214.	224.1.
	224.2.	224.3.
	234.4.	235.
	245.1.	245.2.
	246.	256.1.
	256.2.	256.3.
	25 부록 A.	27 부록 B.
	28	

## 1. 소개

---

### 1.1. 목적

2020 년은 인류의 현대사에 있어서 COVID-19(이하 코로나)의 시대로 기억될 것이다. 코로나 시대는 사람과 사람의 만남의 최소화를 요구했고, 이러한 압박은 교육계에도 동일하게 적용되었다. 강사와 학생이 한 공간에 모여서 수업을 듣는 기존의 방식은 코로나의 확산에 최적의 조건이었기 때문에, 강사와 학생이 접촉하지 않는 교육 방식을 사용할 필요가 있었다. 그렇게 대두된 것이 강사가 강의를 녹화하여 학생들에게 전달하고 학생들이 녹화된 강의를 보면서 공부를 하는 인터넷 강의이다.

인터넷 강의는 코로나 감염에 대한 위험을 극적으로 낮추었지만 정작 학습 효율은 충분한 수준에 이르지 못하였다. 이것은 학습에 대한 적절한 피드백이 없기 때문이다. 이에 'PACEMAKER'는 인터넷 강의와 연동하여 인터넷 강의에 대한 문서를 입력으로 적절한 문제를 생성해주는 서비스를 제공한다. 이를 통해 학생은 적절한 학습 피드백을 받을 수 있을 것으로 기대된다.

'PACEMAKER'는 사용자에게 문제를 전달할 때마다 QG 알고리즘을 통해서 새로운 문제를 생성한다. 이를 통해서 강사는 특별한 노력 없이 학생들에게 다양한 문제를 제공할 수 있다. 또한 이를 개인적인 공부에 사용하는 사람은 하나의 문서를 가지고 다양한 문제를 풀어볼 수도 있다.

결과적으로 'PACEMAKER'의 핵심 기능은 강사와 학생으로 사용자를 구분하여 강사가 만든 학습 그룹(이하 채널)에 학생을 포함시키고, 강사가 작성한 문서를 이용하여 생성된 문제를 학생들에게 전달하는 것이라고 할 수 있다.

### 1.2. 프로젝트 범위

'PACEMAKER'는 자연어 문장에서 질문과 답을 생성하는 QG 알고리즘을 핵심 기술로 하며, 이에 필요한 문서의 저장과 삭제, 편집을 포함한다. 또한 QG 알고리즘을 일반 사용자가 사용할 수 있는 인터페이스, 사용자 계정관리, 문서 관리 등 다양한 부가 기능을 모두 포함한다.

이러한 기능을 구현하기 위해 이 프로젝트는 QG 를 담당하는 부분과 사용자 정보를 저장하기 위한 DB 부분, 이들 기능을 연결하는 서버와 사용자에게 보여지는 인터페이스를 그 범위로 한다.

### 1.3. 참조

프로젝트 제안서: <http://dslab.konkuk.ac.kr/Class/2020/20GP2/Projects/proposal/IT14proposal.pdf>

요구사항 분석서: <http://dslab.konkuk.ac.kr/Class/2020/20GP2/Projects/SRA/IT14SRA.pdf>

## 2. 전체 설명

---

### 2.1. 제품 조망

'PACEMAKER'는 Accelerated Reader(이하 AR)에서 아이디어를 얻은 신규 서비스이다. AR 은 미국 정부에서 제공하는 서비스로 학생들에게 독서를 장려하는 것을 목적으로 한다. AR 은 강사가 학생의 독립적인 독서 연습을 관리하고 모니터링하는 기능을 제공한다. AR 에서는 학생 스스로가 자신의 수준에 맞는 책을 선정하고 자신의 속도에 맞게 그 책을 읽어 나간다. 학생은 자신이 고른 책을 모두 읽으면 AR 을 통해 간단한 퀴즈를 수행하고, 그 결과는

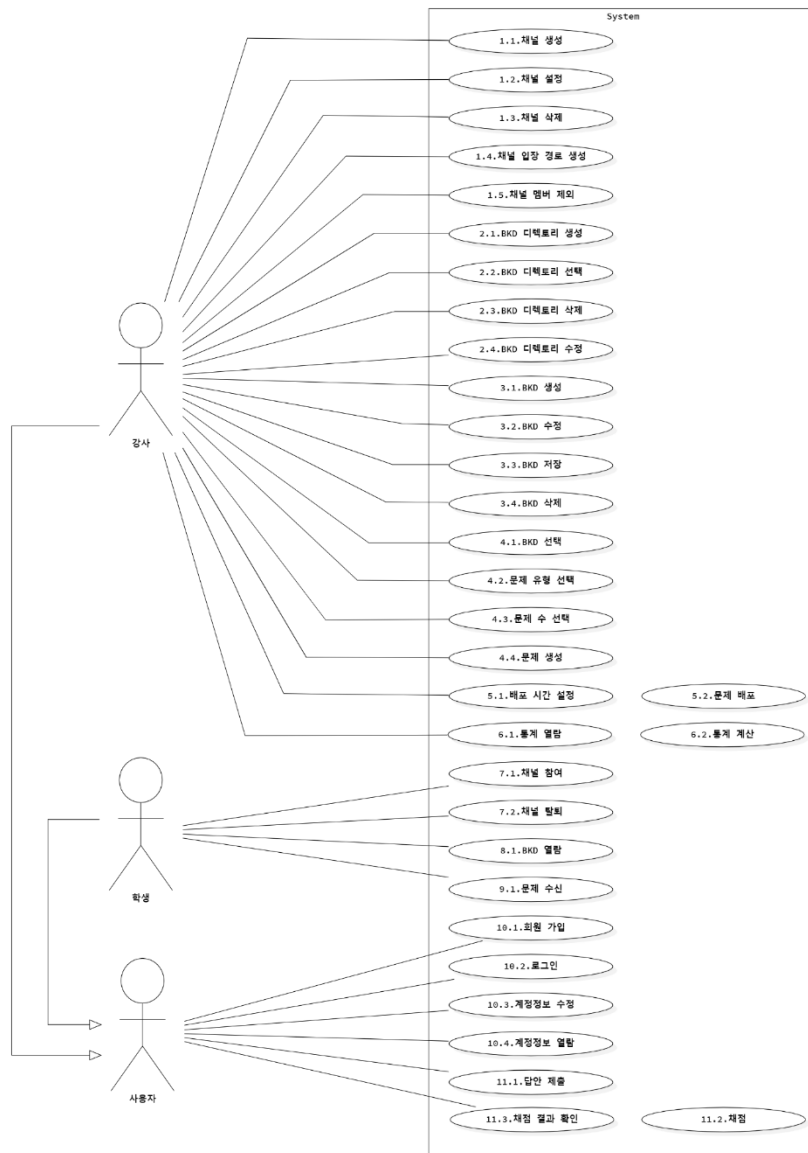
강사에게 피드백으로 제공된다. 'PACEMAKER'는 전반적인 아이디어가 AR 과 유사하지만 퀴즈를 구성하는 문제를 책이 아닌 강사가 작성한 문서에서 생성한다는 것에 차이가 있다.

이 프로젝트에서 가장 핵심적인 기능을 담당하는 것은 QG 인공신경망 모델이다. QG 는 Question Generation 의 약자로서, 자연어 단락과 함께 단락 내부의 단어 스판을 지정하여 모델에 입력하면, 해당 부분을 정답으로 하는 질문이 생성되는 것을 말한다. QG 태스크는 자연어 처리의 대표적인 태스크 중 하나이다. 단락과 질문은 시퀀스 자료이며 QG 는 한 시퀀스를 다른 시퀀스로 전사하는 행위이므로 seq2seq 인공신경망 모델을 사용해 해당 태스크를 학습시킬 수 있다.

seq2seq 는 출발어를 압축된 벡터로 표현하는 인코더와, 이 인코더가 내놓은 벡터를 도착어로 변환하는 디코더로 구성되어 Encoder-Decoder Framework 를 따른다. 인코더의 구현에는 LSTM 셀에 시퀀스 순서대로 자료를 입력하여 시퀀스의 맥락을 파악하는 방식과, 어텐션 메커니즘을 적용하여 한 시점의 자료와 다른 모든 시점 자료 간의 연관성을 파악하는 방식을 쓰게 되는데, 최신 트렌드에 따르면 BERT 처럼 어텐션 메커니즘으로만 구성된 인코더를 사용할 경우 자연어 처리 성능이 증가하는 모습을 볼 수 있다. 이 [사실은 SQuAD1.1 데이터셋 QG 성능 순위](#) 페이지에서 높은 순위를 차지한 모델들을 확인하여 알 수 있다. 우리 프로젝트 역시 상위 모델들을 참고하여 BERT 기반의 인코더를 채용하는 QG 모델을 마련한다.

QG 태스크를 수행하는 인공신경망을 활용하면 특정 지식을 입력 받아 그에 관련된 퀴즈를 자동적으로 생성할 수 있다. 그러므로 강사는 적은 노력을 들여 퀴즈를 빠르게 생성하고 검토한 다음, 이를 학생들에게 배포하여 풀도록 한다. 이후 제출된 답안을 토대로 해당 학생의 학습 상태를 강사와 학생이 점검할 수 있게 된다.

## 2.2. 제품 기능



'PACEMAKER'의 기능은 크게 두가지 영역으로 분리된다. 하나는 문제를 생성하기 위한 문서를 작성하고, 문제를 생성하고, 생성된 문제를 다른 사용자에게 배포하는 '강사'를 위한 기능 영역이다. 이 영역은 다시 채널(학생 그룹)을 만들고 관리하는 기능, 문서를 체계적으로 관리하기 위한 디렉토리 기능, 각각의 문서를 생성하거나 삭제·수정하는 기능, 문제를 생성하기 위한 파라미터를 설정하는 기능, 그리고 채널에 소속된 학생들의 통계를 보는 기능으로 구분된다.

학생의 영역에서는 채널에 참여하거나 탈퇴하는 기능, 강사가 공개한 문서를 열람하는 기능, 배포된 문제를 수신하는 기능으로 구분된다.

여기에 추가적으로 강사와 학생 모두가 공통적으로 포함하는 기능이 있는데, 그것은 회원가입과 로그인 같은 계정관리 기능, 그리고 답안을 제출하고 채점결과를 보는 기능이다. 강사 또한 자신이 생성한 문제를 풀어보거나 채점할 수 있어야 한다고 판단하여 이 기능을 공통 사용자 영역에 포함시켰다.

### 2.3. 대상 사용자와 특징

'PACEMAKER'를 사용하는 사용자는 크게 두 분류로 나눌 수 있다. 하나는 문제를 생성하고자 하는 강사와 또 하나는 문제를 풀어야 하는 학생이다. 이들 집단은 기본적으로 온라인 강의를 사용하는 강사와 그 강의를 시청하는 학생을 전제하고 있지만, 이것이 반드시 온라인 강의에 대해서만 사용될 필요는 없다.

이외에는 자신이 만든 문서에서 문제를 생성하고 이를 스스로 소비하는 사용자도 존재한다. 이러한 경우는 별도의 사용자군을 구성하기보다 상기된 두 집단의 특징을 모두 가지고 있다고 봐도 무방하다고 판단하여 따로 구분하지 않았다.

### 2.4. 운영 환경

서비스의 운영환경 크게 4 가지 부분으로 구분된다.

- 1 QG 서버는 QG 태스크를 수행하는 인공신경망 모델이 실행되는 서버이다. DMS Lab 서버실의 한 인스턴스를 사용할 것이며, Ubuntu 16.04 위에 Python 3.7 과 PyTorch 라이브러리가 설치되고 그 위에 인공신경망 모델이 동작한다. 한편, 모델 성능을 향상시키기 위하여 12GB VRAM 을 가진 NVIDIA Ti GPU 4 대가 지원된다.
- 2 사용자의 정보와 기반 문서를 저장할 DB 서버는 AWS 에서 MySQL(8.0)을 사용하여 구축된다.
- 3 이들 서비스를 연결하고 사용자가 접속하게 될 서버는 AWS 에서 동작된다. 리눅스 환경(Ubuntu 18.04)에서 파이썬(3.6)과 Django(2.0 - 안정화된 버전)를 이용하여 개발 후 AWS 클라우드 서비스를 이용하여 동작할 것이다.
- 4 또한 실제로 사용자가 서비스를 사용하기 위한 UI 역할을 수행하는 부분이 있다. 이 서비스는 웹기반으로 동작하기 때문에, 사용자 UI 는 그 사용자의 PC 에 설치된 브라우저가 될 것이다. 이때 요구되는 환경은 ECAM2015 이상을 구동할 수 있는 크롬 브라우저를 기본으로 한다.

### 2.5. 설계 및 운영 제약사항

해당 프로젝트를 설계함에 있어서 몇 가지 달성해야 하는 목표가 있다.

먼저 QG 모델의 BERT 기반 인코더를 구현함에 있어서 [huggingface/transformers](https://huggingface.co/transformers) 라이브러리를 활용할 것이므로, 구현 언어 및 플랫폼을 Python3.X 와 PyTorch 로 제한한다.

또 신경망 학습의 비용 측면을 고려하여, DMS LAB 의 GPU 서버 중 하나를 제공받아 인공지능 학습 및 서비스화한다. BERT 인코더의 가중치는 Pre-Trained 된 모델의 가중치를 사용하도록 한다.

QG 알고리즘이 변경되거나 QG 서버의 교체가 발생하는 경우에도 유동적으로 대처할 수 있도록 App Server 는 QG 서버와 일정한 인터페이스를 만족하는 통신 이외에, 서로에 대해 의존성을 가지지 않아야 한다. 이는 DB 와 App Server 사이의 관계에서도 마찬가지로 적용될 필요가 있다.

마지막으로 이 프로젝트에서는 Web Client 를 유일한 UI 로 가지고 있지만, 이후에는 안드로이드 어플리케이션이나 Chrome Extension 등과 같은 인터페이스에서도 서비스를 제공할 수 있어야 한다. 이를 위해서 Web Client 와 App Server 의 유착을 최소화하고 가능하면 쉽게 분리할 수 있는 구조를 가져야 한다.

### 3. 유스케이스

---

#### 3.1. 1.1.채널 생성

##### 3.1.1. Type: Evident

##### 3.1.2. 이벤트 흐름

###### 1) 기본 흐름

(A): Actor, (S): System

1. (A): 채널 생성을 요청한다.
2. (S): 채널 생성 인터페이스를 보여준다.
3. (A): 생성할 채널의 정보를 입력한다.
4. (S): 채널 생성이 완료되었음을 통지한다.

###### 2) 대체 흐름

Line4 : 세부 정보가 입력되지 않으면 세부 정보는 기본 값으로 지정된다.

###### 3) 예외 흐름

Line4 : 필수 정보를 입력하지 않으면 채널이 생성되지 않는다.

##### 3.1.3. 사전조건

로그인이 된 상태여야 한다.

##### 3.1.4. 사후조건

새로운 채널이 생성된다.

#### 3.2. 1.2.채널 설정

##### 3.2.1. Type: Evident

##### 3.2.2. 이벤트 흐름

###### 1) 기본 흐름

(A): Actor, (S): System

1. (A): 설정을 요청한다.
2. (S): 채널 설정 화면을 보여준다.
3. (A): 채널의 정보를 유지하거나, 변경한다.
4. (A): 설정된 수정사항을 반영하도록 요청한다.
5. (S): 수정사항에 맞게 채널을 설정한다.

2) 대체 흐름

Line3: 세부 정보가 입력되지 않으면 수정 전 정보를 유지한다.

3) 예외 흐름

3.2.3. 사전조건

채널 목록이 표시된 상태이다.

채널 목록 중 하나를 선택한 상태이다.

3.2.4. 사후조건

채널 정보가 수정된다.

**3.3. 1.3.채널 삭제**

3.3.1. Type: Evident

3.3.2. 이벤트 흐름

1) 기본 흐름

(A): Actor, (S): System

1. (A): 삭제를 요청한다

2. (S): 선택된 채널을 채널 목록에서 삭제한다.

3. (S): 해당 채널의 멤버들을 채널에서 탈퇴 시킨다.

2) 대체 흐름

3) 예외 흐름

3.3.3. 사전조건

채널 목록이 표시된 상태이다.

채널 목록 중 하나를 선택한 상태이다.

3.3.4. 사후조건

채널에 포함된 모든 멤버의 채널 목록에서 채널이 삭제된다.

**3.4. 1.4.채널 입장 경로 생성**

3.4.1. Type: Evident

3.4.2. 이벤트 흐름

1) 기본 흐름

(A): Actor, (S): System

1. (A): 입장 경로 생성을 요청한다.

2. (S): 채널 입장 경로를 생성한다.

2) 대체 흐름

Line2: 채널 입장 경로가 이미 생성된 경우 기존의 경로를 다시 보여준다.

3) 예외 흐름



### 3.4.3. 사전조건

채널 목록이 표시된 상태이다.

채널 목록 중 하나를 선택한 상태이다.

### 3.4.4. 사후조건

채널 입장 경로가 생성된다.

## 3.5. 1.5.채널 멤버 제외

### 3.5.1. Type: Evident

#### 3.5.2. 이벤트 흐름

##### 1) 기본 흐름

(A): Actor, (S): System

1. (A): 멤버 제외를 요청한다.

2. (S): 해당 멤버를 채널 멤버 목록에서 제외한다.

3. (S): 해당 멤버를 채널에서 탈퇴 시킨다..

##### 2) 대체 흐름

##### 3) 예외 흐름

### 3.5.3. 사전조건

채널 멤버 목록이 표시된 상태이다.

채널 멤버가 1 명 선택된 상태이다.

### 3.5.4. 사후조건

멤버가 해당 채널에서 제외된다.

## 3.6. 2.1.BKD 디렉토리 생성

### 3.6.1. Type: Evident

#### 3.6.2. 이벤트 흐름

##### 1) 기본 흐름

(A): Actor, (S): System

1. (A): BKD 디렉토리 생성을 요청한다.

2. (S): BKD 디렉토리 생성 화면을 보여준다.

3. (A): 생성할 BKD 디렉토리의 이름을 입력한다.

4. (S): BKD 디렉토리를 생성하여 BKD 디렉토리 목록에 추가한다.

##### 2) 대체 흐름

Line3: BKD 디렉토리 이름을 입력하지 않으면 기본 이름으로 설정된다.

##### 3) 예외 흐름

Line4: BKD 디렉토리 이름이 중복되면 생성되지 않는다.

3.6.3. 사전조건  
BKD 목록이 표시되고 있다.

3.6.4. 사후조건  
새로운 BKD 디렉토리가 생성된다.

### 3.7. 2.2.BKD 디렉토리 선택

3.7.1. Type: Evident

3.7.2. 이벤트 흐름

1) 기본 흐름

(A): Actor, (S): System

1. (A): BKD 디렉토리 열람을 요청한다.

2. (S): 선택된 BKD 디렉토리 내의 BKD 목록을 보여준다.

2) 대체 흐름

3) 예외 흐름

3.7.3. 사전조건

BKD 디렉토리 목록이 표시된 상태이다.

BKD 디렉토리 목록 중 하나를 선택한 상태이다.

3.7.4. 사후조건

BKD 디렉토리 내부의 BKD 가 표시된다.

### 3.8. 2.3.BKD 디렉토리 삭제

3.8.1. Type: Evident

3.8.2. 이벤트 흐름

1) 기본 흐름

(A): Actor, (S): System

1. (A): BKD 디렉토리 삭제를 요청한다.

2. (S): 선택된 BKD 디렉토리를 삭제한다.

2) 대체 흐름

3) 예외 흐름

3.8.3. 사전조건

BKD 디렉토리 목록이 표시된 상태이다.

BKD 디렉토리 목록 중 하나를 선택한 상태이다.

3.8.4. 사후조건

BKD 디렉토리가 삭제된다.

내부의 BKD 는 삭제되지 않는다.

### 3.9. 2.4.BKD 디렉토리 수정

3.9.1. Type: Evident

3.9.2. 이벤트 흐름

1) 기본 흐름

(A): Actor, (S): System

1. (A): BKD 디렉토리 수정을 요청한다.
2. (S): BKD 디렉토리 수정 화면을 보여준다.
3. (A): BKD 디렉토리 이름을 유지하거나, 변경한다.
4. (A): BKD 디렉토리에 포함될 BKD 를 선택한다.
5. (A): 설정된 수정사항을 반영하도록 요청한다.
6. (S): 수정사항에 맞게 BKD 디렉토리를 수정한다.

2) 대체 흐름

3) 예외 흐름

Line6: 이름이 중복될 경우 수정하지 않는다.

3.9.3. 사전조건

BKD 디렉토리 목록이 표시된 상태이다.

BKD 디렉토리 목록 중 하나를 선택한 상태이다.

3.9.4. 사후조건

BKD 디렉토리가 수정된다.

### 3.10. 3.1.BKD 생성

3.10.1. Type: Evident

3.10.2. 이벤트 흐름

1) 기본 흐름

(A): Actor, (S): System

1. (A): BKD 수정을 요청한다.
2. (S): BKD 수정 화면을 보여준다.
3. (A): BKD 이름이나 내용을 유지하거나, 변경한다.

2) 대체 흐름

3) 예외 흐름

3.10.3. 사전조건

BKD 목록이 표시된 상태이다.

BKD 목록 중 하나를 선택한 상태이다.

3.10.4. 사후조건

BKD 수정에 대한 화면을 표시한다.

### 3.11. 3.2.BKD 수정

3.11.1. Type: Evident

3.11.2. 이벤트 흐름

1) 기본 흐름

(A): Actor, (S): System

1. (A): BKD 수정을 요청한다.

2. (S): BKD 수정 화면을 보여준다.

3. (A): BKD 이름이나 내용을 유지하거나, 변경한다.

2) 대체 흐름

3) 예외 흐름

3.11.3. 사전조건

3.11.4. 사후조건

### 3.12. 3.3.BKD 저장

3.12.1. Type: Evident

3.12.2. 이벤트 흐름

1) 기본 흐름

(A): Actor, (S): System

1. (A): BKD 저장을 요청한다.

2. (S): BKD 를 저장한다.

2) 대체 흐름

3) 예외 흐름

Line2: 이름이 중복되면 저장되지 않는다.

3.12.3. 사전조건

3.12.4. 사후조건

### 3.13. 3.4.BKD 삭제

3.13.1. Type: Evident

3.13.2. 이벤트 흐름

1) 기본 흐름

(A): Actor, (S): System

1. (A): BKD 삭제를 요청한다.

2. (S): 선택된 BKD 를 삭제한다.

2) 대체 흐름

3) 예외 흐름

3.13.3. 사전조건

3.13.4. 사후조건

### 3.14. 4.1.BKD 선택

3.14.1. Type: Evident

3.14.2. 이벤트 흐름

1) 기본 흐름

(A): Actor, (S): System

1. (A): BKD 선택을 요청한다.

2. (S): BKD 목록을 표시한다.

3. (A): 문제를 생성할 BKD 하나를 선택한다.

4. (S): 선택된 BKD 를 표시한다.

2) 대체 흐름

3) 예외 흐름

3.14.3. 사전조건

문제 설정 단계에 있다.

3.14.4. 사후조건

문제를 생성할 BKD 가 선택된 상태이다.

### 3.15. 4.2.문제 유형 선택

3.15.1. Type: Evident

3.15.2. 이벤트 흐름

1) 기본 흐름

(A): Actor, (S): System

1. (A): 생성될 문제 유형을 설정한다.

2. (S): 설정된 문제 유형을 표시한다.

2) 대체 흐름

3) 예외 흐름

3.15.3. 사전조건

문제 설정 단계에 있다.

3.15.4. 사후조건

문제 유형이 선택된 상태이다.

### 3.16. 4.3.문제 수 선택

3.16.1. Type: Evident

### 3.16.2. 이벤트 흐름

#### 1) 기본 흐름

(A): Actor, (S): System

1. (A): 생성될 문제 수를 설정한다.

2. (S): 설정된 문제 수를 표시한다.

#### 2) 대체 흐름

#### 3) 예외 흐름

### 3.16.3. 사전조건

문제 설정 단계에 있다.

### 3.16.4. 사후조건

문제 수가 선택된 상태이다.

## 3.17. 4.4.문제 생성

### 3.17.1. Type: Evident

### 3.17.2. 이벤트 흐름

#### 1) 기본 흐름

(A): Actor, (S): System

1. (A): 문제 생성을 요청한다.

2. (S): 문제 생성에 필요한 설정 화면을 표시한다.

3. (A&S): Usecase 4.1 을 수행한다.

4. (A&S): Usecase 4.2 를 수행한다.

5. (A&S): Usecase 4.3 을 수행한다.

6. (A): 문제 생성을 요청한다.

7. (S): 배포 시간 설정 단계로 넘어간다.

#### 2) 대체 흐름

#### 3) 예외 흐름

Line7: Line3, Line4, Line5 가 수행되지 않으면 요청이 수락되지 않는다.

### 3.17.3. 사전조건

기본 화면이 표시된 상태이다.

### 3.17.4. 사후조건

배포 시간 설정 단계로 넘어간다.

## 3.18. 5.1.배포 시간 설정

### 3.18.1. Type: Evident

### 3.18.2. 이벤트 흐름

1) 기본 흐름

(A): Actor, (S): System

1. (A): 문제를 배포할 채널을 선택한다.
2. (A): 문제를 배포할 날짜/시간/분/초를 설정한다.
3. (A): 문제 배포를 요청한다.
4. (S): 설정된 시간에 따라 문제 배포가 예약된다.

2) 대체 흐름

Line2: 시간이 선택되지 않으면 설정 완료 후 즉각 배포된다.

3) 예외 흐름

Line1: 채널이 선택되지 않으면 자기 자신으로 설정된다.

Line2: 현재 시각보다 이전 시각으로 설정할 수 없다.

3.18.3. 사전조건

문제 생성에 필요한 조건 설정이 완료된 상태이다.

3.18.4. 사후조건

문제 배포가 예약된다.

### 3.19. 5.2.문제 배포

3.19.1. Type: Hidden

3.19.2. 이벤트 흐름

1) 기본 흐름

(S): System

1. (S): 문제를 조건에 맞게 생성한다.
2. (S): 생성된 문제들을 설정된 채널 멤버에게 배포한다.

2) 대체 흐름

3) 예외 흐름

Line2: 설정된 채널이 이미 삭제된 상태라면 배포되지 않는다.

3.19.3. 사전조건

현재 배포되어야 할 문제가 존재한다.

3.19.4. 사후조건

문제가 채널 멤버에게 배포된다.

### 3.20. 6.1.통계 계산

3.20.1. Type: Hidden

3.20.2. 이벤트 흐름

1) 기본 흐름

(A): Actor, (S): System

1. (S): 통계를 계산할 채널의 답안 데이터를 읽어 들인다.
2. (S): 해당 답안 데이터에서 정답률과 응답률을 계산한다
3. (S): 계산된 통계 데이터를 DB 에 저장한다.

2) 대체 흐름

Line1: 이전에 계산한 통계와 변화가 없을 경우 line 2, line 3 를 생략한다.

3) 예외 흐름

Line1: 통계를 계산하기에 충분한 데이터가 없는 경우, 계산을 수행하지 않는다.

3.20.3. 사전조건

통계를 계산할 수 있는 채널이 지정되어 있다.

3.20.4. 사후조건

채널의 통계 데이터를 계산하여 DB 에 저장한다.

### 3.21. 6.2.통계 열람

3.21.1. Type: Evident

3.21.2. 이벤트 흐름

1) 기본 흐름

(A): Actor, (S): System

1. (A): 통계 열람을 요청한다.
2. (S): 통계를 열람할 수 있는 채널을 표시한다.
3. (A): 통계를 열람할 채널을 선택한다.
4. (S): 선택된 채널에 대해 Usecase 6.1.을 수행한다.
5. (S): 해당 채널의 통계와 채널에 포함된 BKD 목록을 표시한다.
6. (A): 통계를 열람할 BKD 를 표시된 목록 중에서 선택한다.
7. (S): 선택된 BKD 의 통계를 표시한다.

2) 대체 흐름

3) 예외 흐름

Line1: 사용자가 관리 중인 채널이 아니라면 요청할 수 없다.

Line2: 생성된 통계가 없을 경우 보여주지 않는다.

3.21.3. 사전조건

3.21.4. 사후조건

통계 결과가 표시된다.

### 3.22. 7.1.채널 참여

3.22.1. Type: Evident



### 3.22.2. 이벤트 흐름

#### 1) 기본 흐름

(A): Actor, (S): System

1. (A): 채널 입장 경로를 통해 채널 참여를 요청한다.
2. (S): 참여 코드 입력을 요구한다.
3. (A): 참여 코드를 입력한다.
4. (S): 해당 채널에 참여시킨다.
5. (S): (A)의 채널 목록에 해당 채널을 추가한다.
6. (S): 채널의 멤버 목록에 해당 유저를 추가한다.

#### 2) 대체 흐름

Line2: 참여 코드가 없는 경우에는 Line4 로 이동한다.

#### 3) 예외 흐름

Line1: 존재하지 않은 채널이면 요청할 수 없다.

Line3: 참여 코드가 잘못된 경우 참여할 수 없다.

### 3.22.3. 사전조건

채널 입장 경로가 생성되어 있다.

### 3.22.4. 사후조건

채널에 입장한다.

## 3.23. 7.2.채널 탈퇴

### 3.23.1. Type: Evident

### 3.23.2. 이벤트 흐름

#### 1) 기본 흐름

(A): Actor, (S): System

1. (A): 탈퇴를 요청한다.
2. (S): 채널 목록에서 해당 채널을 삭제한다.
3. (S): 채널의 멤버 목록에서 해당 멤버를 제외한다.
4. (S): 채널에 남아있는 탈퇴자의 정보를 삭제한다.
5. (S): 채널 관리자에게 탈퇴 사실을 고지한다.

#### 2) 대체 흐름

#### 3) 예외 흐름

### 3.23.3. 사전조건

채널 목록이 표시된 상태여야 한다.

채널 목록 중 하나가 선택된 상태여야 한다.

#### 3.23.4. 사후조건

해당 채널에서 탈퇴한다.

### 3.24. 8.1.BKD 열람

#### 3.24.1. Type: Evident

#### 3.24.2. 이벤트 흐름

##### 1) 기본 흐름

(A): Actor, (S): System

1. (A): BKD 열람을 요청한다.
2. (S): 공개된 BKD 목록을 보여준다.
3. (A): BKD 목록 중 하나를 선택한다.
4. (S): 해당 BKD 내용을 보여준다.

##### 2) 대체 흐름

##### 3) 예외 흐름

#### 3.24.3. 사전조건

채널 목록이 표시된 상태이다.

채널 목록 중 하나를 선택한 상태이다.

#### 3.24.4. 사후조건

BKD 내용이 표시된다.

### 3.25. 9.1.문제 수신

#### 3.25.1. Type: Evident

#### 3.25.2. 이벤트 흐름

##### 1) 기본 흐름

(A): Actor, (S): System

1. (S): 배포된 문제를 수신한다.
2. (A): 수신된 문제의 열람을 요청한다.
3. (S): 문제 내용을 보여준다.

##### 2) 대체 흐름

##### 3) 예외 흐름

#### 3.25.3. 사전조건

해당 사용자가 속한 채널에 문제가 배포된 상태이다.

#### 3.25.4. 사후조건

문제가 표시된다.

### 3.26. 10.1.회원 가입

#### 3.26.1. Type: Evident

#### 3.26.2. 이벤트 흐름

##### 1) 기본 흐름

(A): Actor, (S): System

1. (A): 회원가입을 요청한다
2. (S): 개인정보 입력 창을 띄워준다
3. (A): 개인정보를 입력한다. (ID/PW 포함)
4. (A): ID 의 중복 여부 확인을 요청한다.
5. (S): ID 의 중복 여부를 알려준다.
6. (A): 회원가입에 성공한다.

##### 2) 대체 흐름

Line6: Line5 에서 ID 가 중복이면, ID 가 중복이 아닐 때까지 Line3~Line5 를 계속 반복한다.

##### 3) 예외 흐름

Line3: 필수 개인정보 중 하나라도 입력되지 않았을 시 Line4 로 넘어가지 않는다.

#### 3.26.3. 사전조건

로그인 전 화면이 띄워진 상태이다.

#### 3.26.4. 사후조건

회원가입이 완료된다.

### 3.27. 10.2.로그인

#### 3.27.1. Type: Evident

#### 3.27.2. 이벤트 흐름

##### 1) 기본 흐름

(A): Actor, (S): System

1. (A): ID 와 PW 를 입력하여 로그인을 요청한다
2. (S): 로그인을 실행한다.

##### 2) 대체 흐름

##### 3) 예외 흐름

Line1: ID 나 PW 중 하나를 입력하지 않을 경우 로그인에 실패한다.

Line1: ID 와 PW 의 짝이 맞지 않는다면 로그인에 실패한다.

#### 3.27.3. 사전조건

로그인 전 화면이 띄워진 상태이다.

#### 3.27.4. 사후조건

로그인에 성공한다.

### 3.28. 10.3.계정정보 수정

#### 3.28.1. Type: Evident

#### 3.28.2. 이벤트 흐름

##### 1) 기본 흐름

(A): Actor, (S): System

1. (A): 계정정보 수정을 요청한다.
2. (S): 수정 전 본인확인을 위한 PW 입력을 요구한다.
3. (A): PW 를 입력한다.
4. (S): 개인정보 수정화면을 띄워준다.
5. (A): 개인정보를 유지하거나 변경한다.
6. (A): 설정된 수정사항을 반영하도록 요청한다.
7. (S): 수정사항에 맞게 계정정보를 수정한다.

##### 2) 대체 흐름

##### 3) 예외 흐름

Line3: PW 가 해당 계정의 PW 와 일치하지 않으면 Line4 로 넘어가지 않는다.

Line5: 필수 입력정보를 입력하지 않으면 Line6 으로 넘어가지 않는다.

#### 3.28.3. 사전조건

로그인이 완료 된 상태여야 한다.

#### 3.28.4. 사후조건

계정정보가 수정된다.

### 3.29. 10.4.계정정보 열람

#### 3.29.1. Type: Evident

#### 3.29.2. 이벤트 흐름

##### 1) 기본 흐름

(A): Actor, (S): System

1. (A): 정보 열람을 요청한다.
2. (S): 선택된 멤버의 공개 가능한 계정정보를 표시한다.

##### 2) 대체 흐름

##### 3) 예외 흐름

#### 3.29.3. 사전조건

관리 중인 채널의 멤버 목록이 표시된 상태이다.

멤버 목록 중 하나를 선택한 상태이다.

#### 3.29.4. 사후조건

계정정보를 표시한다.

### 3.30. 11.1.답안 제출

#### 3.30.1. Type: Evident

#### 3.30.2. 이벤트 흐름

##### 1) 기본 흐름

(A): Actor, (S): System

1. (A): 열람한 문제들의 답안을 작성한다.

2. (A): 답안 제출을 요청한다.

3. (S): 답안 정보를 저장한다.

##### 2) 대체 흐름

##### 3) 예외 흐름

#### 3.30.3. 사전조건

문제를 수신하여 이를 열람한 상태여야 한다.

#### 3.30.4. 사후조건

문제들의 답안을 제출한다.

### 3.31. 11.2.채점

#### 3.31.1. Type: Hidden

#### 3.31.2. 이벤트 흐름

##### 1) 기본 흐름

(S): System

1. (S): 제출된 답안과 정답을 비교하여 채점한다.

2. (S): 채점 결과를 저장한다.

##### 2) 대체 흐름

##### 3) 예외 흐름

#### 3.31.3. 사전조건

답안이 제출된 상태이다.

#### 3.31.4. 사후조건

채점 결과를 저장한다.

### 3.32. 11.3.채점 결과 확인

#### 3.32.1. Type: Evident

#### 3.32.2. 이벤트 흐름

- 1) 기본 흐름  
(A): Actor, (S): System
  1. (A): 채점 결과 확인을 요청한다.
  2. (S): 채점 결과를 보여준다.
- 2) 대체 흐름
- 3) 예외 흐름

3.32.3. 사전조건  
채점이 완료된 상태이다.

3.32.4. 사후조건  
채점 결과를 보여준다.

## 4. 시스템 특징

---

### 4.1. QG Server

#### 4.1.1. 설명과 우선순위

인공신경망 모델을 사용하여 사용자가 입력한 BKD 를 토대로 문제를 만들어낸다. 프로젝트의 핵심 주제인 "자동 문제 생성" 기능을 담당하므로 가장 높은 우선순위로 구현되어야 한다.

#### 4.1.2. 입력/출력 순서

App Server 에 의해 호출되어 문제를 생성할 단락을 입력 받는다. 해당 단락에서 정답이 될 만한 정답 스펠을 찾은 다음, 각 정답 스펠에 대해 문제를 생성한다. 생성된 문제와 정답 쌍을 App Server 에 반환한다.

#### 4.1.3. 기능 요구사항

App Server 로부터 BKD 를 입력 받아 정답 스펠을 파악할 수 있어야 한다.

BKD 와 정답 스펠을 입력 받은 인공신경망이 문제를 생성할 수 있어야 한다.

정답 문제 쌍을 App Server 로 반환할 수 있어야 한다.

### 4.2. App Server

#### 4.2.1. 설명과 우선순위

사용자의 요청에 따라 수행할 작업을 컨트롤 한다. 문제 생성 요청 시 DB 와 QG 서버를 통해 문제를 생성하고, 로그인과 회원가입, 채널의 전반적인 관리와 사용자 계정의 전반적인 관리를 수행한다. 또한, 채점 후의 결과를 통계내는 작업도 수행한다.

사용자의 요청에 따른 작업 수행 컨트롤이 주 목적이기 때문에 프로젝트의 핵심 주제인 "자동 질문 생성" 역할을 담당하는 QG 서버의 뒤를 이어 두번째로 높은 우선순위를 갖는다.

#### 4.2.2. 입력/출력 순서

Web Client 로부터 사용자의 요청에 따라 각기 다른 입력을 받는다. 수행해야 하는 작업에 따라 DB 에 저장된 데이터가 필요하다면 DB Server 에, 질문이 생성되어야 하면 QG Server 에 필요한 출력을 한다. 그리고 DB Server 의 출력이나 QG Server 의 출력을 입력으로 받으면, 이를 Web Client 에 출력한다.

#### 4.2.3. 기능 요구사항

DB 에 필요한 데이터를 저장하도록 요청할 수 있어야 한다.

DB 에 저장되어 있는 Data 를 읽거나 쓸 수 있어야 한다.

문제 생성에 필요한 Data 는 QG 서버에 넘겨줄 수 있어야 한다.

사용자가 요청한 작업의 결과를 Web Client 에게 보낼 수 있어야 한다.

사용자가 요청한 작업 이외의 자동으로 처리되어야 하는 채점이나 통계 계산을 수행할 수 있어야 한다.

### 4.3. DB Server

#### 4.3.1. 설명과 우선순위

DB Server 는 계정, 채널, BKD 등의 데이터를 저장하여 관리하는 기능을 한다. App Server 와 소통하며 상호 간에 데이터를 전달한다. 시스템의 전체적인 구현에 있어서는 핵심 알고리즘 구현 및 App Server 의 구현보다는 낮은 우선순위를 갖는다.

#### 4.3.2. 입력/출력 순서

DB Server 는 App Server 로부터 입력을 받는다. 입력에 따라 App Server 로부터 받은 데이터를 DB 에 저장하거나, 혹은 DB 에 저장된 데이터를 App Server 에 출력할 수 있다. DB Server 는 오직 App Server 로부터의 입력에 의해서 수동적으로 동작한다.

#### 4.3.3. 기능 요구사항

데이터에 대한 총괄적 관리 및 출력이 요구된다. 입력 받은 데이터를 토대로 DB 를 갱신하는 기능이 필요하며, 요청에 따라 데이터를 조건에 맞게 뽑아내어 적합한 형태로 출력하여 App Server 에 전달하는 기능 또한 필요하다.

### 4.4. Web Client

#### 4.4.1. 설명과 우선순위

Web Client 는 사용자와 직접적으로 상호작용을 하는 UI 를 담당하며 App Server 에서 제공한 정보를 사용자에게 적절한 형태로 가공하여 보여주는 역할을 수행한다. Web Client 는 SPA(Single Page Application)로 구성될 것이고 사용자에게 직접적으로 보여지는 부분인 만큼 상당히 중요한 역할을 담당한다.

하지만, 요구하는 인터페이스를 모두 포함하고 있다면 시스템의 기능을 충족하는 것에 있어서 높은 우선순위를 부여하기 어렵다. 따라서 Web Client 는 기본적인 기능을 우선적으로 구현하여 App Server 와의 통신을 테스트하고, 나머지 컴포넌트의 기능이 충분히 구현되면 그때 UI 와 사용성을 고려하여 다시 구현할 것이다.

#### 4.4.2. 입력/출력 순서

Web Client 는 실제 사용자와 App Server 로부터 입력을 받을 수 있다. 사용자에 의한 입력이 발생하면 Web Client 는 입력에 대한 정보를 서버로 출력한다. 서버에서 해당 입력에 대한 적절한 처리를 반환하면 Web Client 는 반환된 데이터를 가공하여 사용자에게 표시한다.

App Server 가 Web Client 에 입력을 하는 경우는 Server 에 의한 Push 가 발생하는 경우이다. 이는 예를 들어 '9.1.문제 수신'과 같은 경우에 발생하며, 사용자의 입력 없이 서버에서 바로 Web Client 에게 표시나 업데이트를 지시한다.

#### 4.4.3. 기능 요구사항

Web Client 에 요구되는 기능은 사용자와의 상호작용 대부분이다. 문제의 생성, 통계 계산 등과 같이 순수하게 서버에서만 동작하는 경우를 제외한 대부분을 포함한다. 이는 로그인 등을 포함하는 계정관리 부분, 문서 디렉토리를 관리하는 부분, 문서를 작성하는 부분, 채널을 생성하고 관리하는 부분, 문제를 생성하는 부분, 문제를 배포하는 부분, 문제를 수신하는 부분, 문제를 풀고 결과를 확인하는 부분, 통계 정보를 확인하는 부분으로 구분된다. 세부적인 사항은 보다 구체화될 수 있다.

Web Client 는 사용자와 직접적으로 상호작용을 하는 만큼 사용자의 무효한 입력이나 잘못된 접근을 1 차적으로 차단한다(App Server 에서 2 차적으로 검증을 수행한다). 또한 서버와의 통신에 있어서 그 데이터를 암호화하는 등 보안에 대한 기능도 Web Client 에서 담당해야 할 것이다.

## 5. 외부 인터페이스 요구사항

---

### 5.1. 사용자 인터페이스

이 프로젝트는 그 목적이 꽤나 단순하다. 결국은 문서를 받아서 이를 이용해서 문제를 생성하면 되는 것이다. 나머지는 그 문제를 어떻게 쓸 것인가에 대한 문제라고 할 수 있다. 하지만 그 과정에서 다양한 기능이 요구되고 그 때문에 사용자는 이 서비스가 자칫 복잡하고 사용하기 어렵다는 느낌을 받을 수도 있다. 이를 막기 위해서 사용자 인터페이스는 매우 간결하고 단순하며 직관적일 것을 요구한다.

추가적으로 'PACEMAKER'의 인터페이스는 데스크톱 앱과 유사한 사용자 경험을 제공할 것을 요구한다. 이를 위해서는 SPA 형태로 웹을 구현할 필요가 있다. 단, 브라우저에서 구동되는 것은 동일하므로 단축키와 같은 기능은 제외하도록 한다.

### 5.2. 통신 인터페이스

APP Server 와 DB Server 사이의 통신은 TCP/IP Socket 통신을 사용하여 구현한다. 두 Server 간의 통신 시, 주고 받는 Data 가 손상되거나 잃어버리지 않아야 한다.

APP Server 와 QG Server 사이의 통신은 RESTful 통신을 사용하여 구현한다. 이는 QG Server 를 일종의 Web Server 와 같은 형태로 구현하여 App Server 의 요청을 처리하는 형태로 예상된다.

APP Server 와 Web Client 사이의 통신은 JSON 을 사용한 RESTful 로 구현한다. 이때 Web Client 의 Reload 가 발생하지 않도록 Update 에 필요한 데이터만 주고 받아야 한다. dd



## 6. 비 기능 요구사항

---

### 6.1. 성능 요구사항

BKD 하나 당 문제 생성에 소요되는 시간이 15 초 미만이어야 한다.

사용자가 BKD 를 입력하고 문제의 생성을 기다릴 때 너무 오랜 시간이 걸린다면 서비스 품질 저하로 이어진다. 그러므로 문제 생성을 완수하는데 걸릴 시간을 제한할 필요가 있다. 모델에 사용되는 BERT 인코더는 상당한 계산량이 요구되는 병목점으로 작용하게 되는데 이를 직접적으로 해결하기 위해 컴퓨팅 성능을 높이는 방법엔 한계가 존재한다. 그러므로 사용자 입력의 크기를 제한하는 간접적 방식으로 달성되어야 할 것이다.

### 6.2. 보안 요구사항

보안사항은 다음 3 가지 부분으로 나누어져 있다.

#### i. App Server 와 QG Server 사이의 통신 암호화

App Server 와 QG Server 는 RESTful 을 이용하여 통신한다. 이때 데이터에 해당하는 JSON 부분에 대한 적절한 암호화가 필요하다. 또한 App 서버에서 QG 서버를 사용할 때 적절한 인증을 통해서 적절치 않은 QG 서버로의 접근을 방지할 필요가 있다.

#### ii. App Server 와 Web Client 사이의 통신 암호화

App Server 와 Web Client 는 RESTful 을 이용하여 통신한다. Web Client 는 사용자로부터 개인정보나 ID, 비밀번호와 같은 민감한 정보를 입력 받아서 App Server 로 전달한다. 이 과정에서의 통신은 반드시 암호화를 거쳐야 한다. 또한 BKD 나 타 사용자의 ID, 채널에 대한 정보 등도 민감도는 비교적 낮으나 이 역시 평문으로 전송하는 것은 무리가 있다. 따라서 App Server 와 Web Client 사이의 통신은 기본적으로 암호화를 적용할 필요가 있다.

#### iii. DB 내부에 저장하는 데이터의 암호화

DB 에는 사용자의 개인정보와 ID, PW 등을 포함한 매우 민감한 정보들이 다수 포함되어 있다. 따라서 최악의 경우에 DB 가 유출된다 하더라도 그 정보를 사용하지 못하게 할 필요가 있다. 이를 위해서 DB 에 저장되는 모든 정보는 App Server 에 의해 암호화하여 전송되어야 하며, DB Server 는 복호화 하지 못하는 상태로 DB 에 저장을 해야 한다. 다만 이 경우에 App Server 에서 DB 의 데이터를 검색하기 위한 고유번호는 암호화하지 않는다.

### 6.3. 서비스 품질 요구사항

테스트 BKD 로 생성된 문제의 80% 이상을 사용할 수 있다.

인공신경망이 생성한 문제는 비문을 포함하거나 정답과는 전혀 연관 없이 엉뚱한 문제를 생성할 수도 있다. 자동 생성한 문제를 활용할 수 없다면 우리의 서비스의 존재는 타당하지 않기 때문에 모델 성능을 끌어올리는 데 많은 자원이 투입되어야 한다. 입력 가능한 모든 BKD 에 대하여 모델의 출력을 검수할 수 없으므로, 테스트 단계에서 사용할 BKD 를 마련하고, 해당 BKD 에 대하여 모델이

생성하는 문제의 사용가능성을 평가한다. 여기서 사용가능성을 평가한다는 것은 모델이 반환한 문제 정답 쌍이 논리적으로 매치되는 지 일일이 살펴본다는 것이다.

부록A. Glossary

순번	용어	설명
1	BKD	문제 생성의 기반이 되는 정보가 작성된 문서
2	채널	BKD 를 통해 생성된 문제를 배포 받을 수 있는 개인 및 집단
3	강사	BKD 를 사용하여 문제를 생성 및 배포하는 사용자
4	학생	생성된 문제를 배포 받아 풀이하는 사용자
5	채널 관리자	채널에 멤버를 초대하거나 탈퇴 시키는 등의 채널에 대한 모든 권한을 가지는 사용자. 채널을 생성한 사용자.
6	채널 멤버	채널 관리자에 의해 생성된 채널에 소속된 사용자
7	QG	BKD 를 입력 받아 인공지능망을 사용하여 자동적으로 질문과 답의 쌍을 생성하는 행위
8	문제	QG 에 의해 생성된 질문과 답의 쌍. 그 중에서 질문을 표현한 것.
9	정답 스팩	어떤 단락 내부에서 질문의 정답으로 활용할 부분이다. 예를 들어, "저는 15 살 학생입니다" 라는 단락에서 "15 살 학생" 은 "저는 누구입니까?" 라는 질문에 대한 정답 스팩이라 할 수 있다.
10	정답률	문제를 수령한 사용자 대비 정답을 선택한 사용자의 비율
11	응답률	문제를 수령한 사용자 대비 답안을 제출한 사용자의 비율

## 부록B. Analysis Model

