

# Requirement Specification

## QURIOUS

소프트웨어공학개론 4 조



|     |           |      |            |
|-----|-----------|------|------------|
| 제출일 | 19.05.06  | 그룹   | 4조 Qurious |
| 과목  | 소프트웨어공학개론 | 담당교수 | 이은석 교수님    |
| 이름  | 조상연       | 학번   | 2013313217 |
| 이름  | 김연재       | 학번   | 2014311060 |
| 이름  | 최지원       | 학번   | 2015311016 |
| 이름  | 손헌        | 학번   | 2013310546 |
| 이름  | 정광현       | 학번   | 2013313119 |

## 목차

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b><i>Preface</i></b>                     | <b>5</b>  |
| 1.1      | Objective                                 | 5         |
| 1.2      | Readership                                | 5         |
| 1.3      | Document Structure                        | 5         |
| A.       | Preface                                   | 5         |
| B.       | Introduction                              | 5         |
| C.       | Glossary                                  | 6         |
| D.       | User Requirement Definition               | 6         |
| E.       | System Architecture                       | 6         |
| F.       | System Requirement Specification          | 6         |
| G.       | System Model                              | 6         |
| H.       | System Evolution                          | 7         |
| I.       | Appendix                                  | 7         |
| J.       | Index                                     | 7         |
| <b>2</b> | <b><i>Introduction</i></b>                | <b>7</b>  |
| 2.1      | Objective                                 | 7         |
| 2.2      | Needs                                     | 7         |
| 2.3      | Curious                                   | 9         |
| 2.4      | Expectations                              | 10        |
| A.       | 학습자                                       | 10        |
| B.       | 교수자                                       | 10        |
| <b>3</b> | <b><i>Glossary</i></b>                    | <b>10</b> |
| 3.1      | Objective                                 | 10        |
| 3.2      | Term Definition                           | 10        |
| 3.3      | Acronyms and Abbreviations                | 11        |
| <b>4</b> | <b><i>User Requirement Definition</i></b> | <b>11</b> |
| 4.1      | Objective                                 | 11        |

|            |   |           |
|------------|---|-----------|
| <b>4.2</b> | <b>Functional requirement</b>           | <b>12</b> |
| A.         | 회원가입                                    | 12        |
| B.         | 로그인                                     | 12        |
| C.         | 레벨 테스트                                  | 12        |
| D.         | 문제 추천 받기                                | 12        |
| E.         | 내부 문제 풀기                                | 12        |
| F.         | 외부 문제 풀기                                | 13        |
| G.         | 마이페이지                                   | 13        |
| H.         | 랭킹 확인                                   | 13        |
| I.         | 내부 문제 관리                                | 13        |
| J.         | 외부 문제 관리                                | 14        |
| <b>4.3</b> | <b>Nonfunctional requirement</b>        | <b>14</b> |
| A.         | Product Requirement                     | 14        |
| B.         | Organization Requirement                | 15        |
| C.         | External Requirement                    | 15        |
| <b>5</b>   | <b>System Architecture</b>              | <b>16</b> |
| <b>5.1</b> | <b>Objective</b>                        | <b>16</b> |
| <b>5.2</b> | <b>Curious System Architecture</b>      | <b>17</b> |
| <b>5.3</b> | <b>Sub system</b>                       | <b>18</b> |
| A.         | 웹(메인)서버 시스템 상세                          | 18        |
| B.         | 사용자 시스템                                 | 19        |
| C.         | 랭킹시스템                                   | 20        |
| D.         | 진단고사 시스템                                | 21        |
| E.         | 문제 풀이 시스템                               | 22        |
| F.         | 문제 추천 시스템                               | 23        |
| G.         | 문제 관리 시스템                               | 24        |
| <b>6</b>   | <b>System Requirement Specification</b> | <b>24</b> |
| <b>6.1</b> | <b>Objective</b>                        | <b>24</b> |
| <b>6.2</b> | <b>Functional Requirement</b>           | <b>25</b> |
| A.         | 회원가입                                    | 25        |

|  |           |
|--|-----------|
| B. 로그인 .....                               | 25        |
| C. 진단고사 응시 .....                           | 26        |
| D. 진단고사 채점 .....                           | 27        |
| E. 문제 추천 받기 .....                          | 27        |
| F. 추천 서버에서 문제 추천하기 .....                   | 28        |
| G. 진단고사 기반 추천 .....                        | 29        |
| H. 내부 문제 풀기 .....                          | 29        |
| I. 외부 문제 풀기 .....                          | 30        |
| J. 마이페이지 .....                             | 31        |
| K. 랭킹확인 .....                              | 31        |
| L. 내부 문제 관리 .....                          | 32        |
| M. 외부 문제 관리 .....                          | 33        |
| <b>6.3 Nonfunctional Requirement .....</b> | <b>33</b> |
| A. Product Requirement .....               | 33        |
| B. Organization Requirement .....          | 34        |
| C. External Requirement .....              | 35        |
| <b>6.4 Scenario .....</b>                  | <b>35</b> |
| A. 회원가입 .....                              | 35        |
| B. 로그인 .....                               | 36        |
| C. 레벨 테스트 .....                            | 37        |
| D. 문제 추천 받기 .....                          | 38        |
| E. 내부 문제 풀기 .....                          | 39        |
| F. 외부 문제 풀기 .....                          | 40        |
| G. 마이페이지 .....                             | 41        |
| H. 랭킹 확인 .....                             | 42        |
| I. 내부 문제 관리 .....                          | 43        |
| J. 외부 문제 관리 .....                          | 44        |
| <b>7 System Model .....</b>                | <b>45</b> |
| <b>7.1 Objective .....</b>                 | <b>45</b> |
| <b>7.2 Context Models .....</b>            | <b>45</b> |
| A. Context Model .....                     | 45        |

|   |           |
|---|-----------|
| B. Process Model .....                      | 45        |
| <b>7.3 Interaction Models .....</b>         | <b>47</b> |
| A. Use Case Model .....                     | 47        |
| B. Tabular Description .....                | 48        |
| C. Sequence Diagram .....                   | 53        |
| <b>7.4 Structural Models .....</b>          | <b>58</b> |
| A. Class Diagram .....                      | 58        |
| <b>7.5 Behavioral Models .....</b>          | <b>59</b> |
| A. Event-driven Diagram .....               | 59        |
| <b>8 System Requirement Evolution .....</b> | <b>60</b> |
| <b>8.1 Objective .....</b>                  | <b>60</b> |
| <b>8.2 Limitation .....</b>                 | <b>60</b> |
| A. 외부 문제 의존성 .....                          | 60        |
| B. 추천 정확도 문제 .....                          | 61        |
| <b>8.3 System Evolution .....</b>           | <b>61</b> |
| A. Evolution for AI tutor .....             | 61        |
| B. Evolution for Mobile .....               | 61        |
| C. Evolution for User Requirement .....     | 62        |
| <b>9 Appendix .....</b>                     | <b>62</b> |
| <b>9.1 Objectives .....</b>                 | <b>62</b> |
| <b>9.2 Database Requirements .....</b>      | <b>62</b> |
| <b>9.3 Database Detail .....</b>            | <b>63</b> |
| <b>10 Index .....</b>                       | <b>63</b> |
| <b>10.1 Objectives .....</b>                | <b>63</b> |
| <b>10.2 Table Index .....</b>               | <b>63</b> |
| <b>10.3 Figure Index .....</b>              | <b>64</b> |
| <b>10.4 Diagram Index .....</b>             | <b>65</b> |
| <b>11 Reference .....</b>                   | <b>66</b> |

# 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 이용해 정의하고 표현하며 서술한다. 전문 지식이 없는 독자들을 고려하여 자연어와 도표, 그림 등을 이용해 서술한다. Client manager과 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 developer과 system architects들을 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 구성하는 컴포넌트와 이들의 상호작용을 기술한다. 서술의 이해도를 높이기 위해 시각화를 이용한다. 개발자가 시스템의 전체적인 구조를 이해하는 데 도움을 준다.

### F. System Requirement 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 Model

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. Appendix

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

2015년 7월 교육부와 과학기술정보통신부(전 미래창조과학부)는 'SW중심사회를 위한 인재양성 추진계획'을 발표했다. 계획의 기본방향은 미래사회의 창의인재 양성을 위해 SW 교육을 초·중·고에 확산하고, 현장 수요에 부합하는 인력 공급을 위해 대학 SW교육의 혁신을 추진한다는 구체적인 내용이다. 초·중·고 교육과정 개편의 핵심은 2018년부터 중학교의 정보 과목이 선택교과에서 34시간 이상의 필수교과로 바뀌었으며, 2019년부터 초등학교의 실과 내의 ICT 단원이 17시간 이상의 SW 기초교육 실시로 개편된다. 초등학교의 실과 과목 개편은 문제해결과정, 알고리즘, 프로그래밍 체험 및 정보윤리가 주요 내용이며, 중학교의 정보 과목 개편은 컴퓨팅사고 기반 문제 해결, 간단한 알고리즘, 프로그래밍 개발 등을 진행한다.





Figure 1 SW중심사회를 위한 인재양성 추진계획

| 구분               | 기존                       | 개편안                             | 주요 개편 방향                                    |
|------------------|--------------------------|---------------------------------|---|
| 초등학교<br>(2019년~) | 실과 내<br>ICT 단원<br>(12시간) | 실과 내<br>SW 기초교육 실시<br>(17시간 이상) | · 문제해결과정, 알고리즘, 프로그래밍 체험<br>· 정보윤리의식 함양     |
| 중학교<br>(2018년~)  | 정보 과목<br>[선택교과]          | 정보 과목<br>[필수교과]<br>(34시간 이상)    | · 컴퓨팅사고 기반 문제 해결 실시<br>· 간단한 알고리즘, 프로그래밍 개발 |
| 고등학교<br>(2018년~) | 정보 과목<br>[심화선택 과목]       | 정보 과목<br>[일반선택 과목]              | · 다양한 분야와 융합하여 알고리즘,<br>프로그래밍 설계            |

자료: 'SW 중심사회를 위한 인재양성 추진계획', 교육부, 과학기술정보통신부, BNK 투자증권

Figure 2 주요 개편 방향

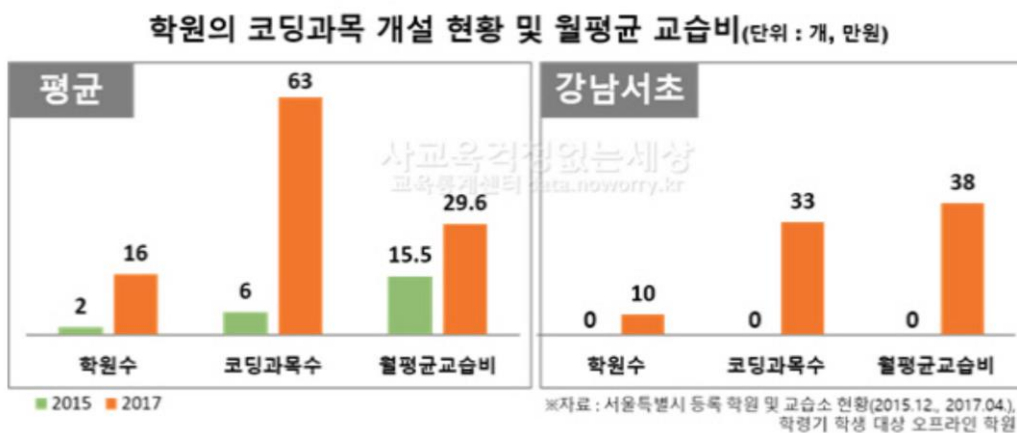


Figure 3 사교육 현황

정부에서 SW중심사회를 대비하는 정책을 펼치고 있으며, 이에 따라 초·중고 교육과정도 개편되고 있다. 소프트웨어 교육의 필요성이 증대되고 공교육에도 적용되고 있다.

소프트웨어 교육의 필요성이 증대되고 초·중등 소프트웨어교육 의무화로 소프트웨어 교육을 실시하는 학교의 비율이 2012년 21.9%에서 2015년 40.4%로 증가했다. 국가과학기술자문회의에 따르면, 소프트웨어 융합 교육 인식 조사 결과 초·중등 학부모 88%가 미래 사회를 살아가는 필수 역량으로 소프트웨어를 응답하는 등 학부모의 소프트웨어 교육의 관심이 확대되고 있다.

그러나 중학교의 '정보'교과 담당 교원은 2016년 학교당 평균 0.4명에 불과하는 등 소프트웨어 교육 콘텐츠와 교원, 인프라가 미흡한 상태이다. 이에 따라 학부모와 학생은 사교육에 의지할 수 없는 실정이며, 부담도 증대되고 있다.

본 서비스는 사용자의 코딩 실력 수준을 진단하고 이에 따라 문제를 추천하여 학부모와 학생의 부담을 감소시키고 프로그래밍에 대한 흥미를 증가시키고자 한다.

## 2.3 Quirious

개인화된 맞춤 진단고사를 제공하고, python을 이용하여 컴퓨팅 사고와 기본 프로그래밍 문법, 알고리즘, 자료구조들을 배울 수 있는 SW 교육 서비스. 본 서비스에 자체적으로 제작한 문제뿐 아니라, 본 서비스 외 다른 외부 문제 은행의 양질의 문제들 또한 사용자 수준별로 제공한다. 맞춤 서비스를 통해 불평등을 최대한 해소한다.

시스템은 크게 레벨테스트 기능, 추천 기능, 문제 풀이 기능을 가지고 있다.

레벨테스트 기능은 사용자가 Quirious의 진단고사를 풀면 초급, 중급, 고급 별 3단계 기준으로 사용자의 실력을 측정하는 기능이다. 진단고사는 처음 사용자가 초급, 중급, 고급 난이도를 선택할 수 있으며 객관식과 코딩 테스트로 이루어져 있다. 정답률을 기준으로 사용자의 실력수준이 결정되며, 정답률에 따라 사용자가 선택한 진단고사 난이도보다 실력수준이 낮거나 높게 나올 수도 있다.

추천 기능은 사용자의 레벨테스트 결과인 실력 수준을 참고하여 고객 맞춤형으로 알고리즘 문제를 추천하는 기능이다. 이 때 사용자는 Quirious의 자체 제작 문제인 내부 문제를 추천 받을 수도 있고, 기존의 유명한 알고리즘 사이트(백준, 해커랭크)의 문제들을 추천 받을 수도 있다. 다른 사용자들과 협업필터링을 사용하기 때문에, 추천 알고리즘은 하루 단위로 갱신된다.

풀이 기능은 사용자가 추천된 문제를 풀고 채점하는 기능이다. 풀이 기능은 내부 문제만

지원하며, 외부 문제는 외부 문제 은행 사이트에서 직접 본다. 추천 문제 설명과 입력값, 출력값 예시가 주어지고 사용자는 코드 에디터에서 코드를 작성한다. 제출을 하면 해당 문제에 대한 결과화면을 통해 자신의 코드를 진단받을 수 있다. 추천된 내부 문제를 일정한 기준 이상 해결하면, 사용자의 레벨은 상승하게 된다.

## 2.4 Expectations

### A. 학습자

진단고사를 통해 학생의 실력을 측정하고, 분반을 나누는 자료로 추후 사용할 수 있다. 학기 중 학생들의 수준을 측정 후 수업 방향을 수정하거나 수업의 커리큘럼에 반영하는 등 수업의 질 향상에 도움을 받을 수 있다. 직접적으로 문제들을 수업 자료로 활용할 수 있다.

### B. 교수자

학습자는 프로그래밍에 흥미를 갖고 입문하는 것부터 해커톤 등 프로그래밍 대회를 준비하거나 또는 입사시험을 준비하는 등 다양한 수준에서 본 서비스를 사용할 수 있다. 기본적으로 python 기반 프로그래밍 실력 향상을 기대할 수 있다.

## 3 Glossary

### 3.1 Objective

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.

### 3.2 Term Definition

| 용어            | 정의   |
|---------------|--|
| <b>Python</b> | 컴퓨터 언어의 일종으로 간결하고 생산성 높은 프로그래밍 언어.   |
| <b>실력수준</b>   | 사용자의 진단고사 결과물로, 사용자의 컴퓨터 지식 및 python 지식의 수준을 나타낸다. 실력수준은 등급(초급, 중급, 고급)과 단계(1 단계, 2 단계, 3 단계)로 표시한다. |

|                 |  |
|-----------------|--|
| <b>내부 문제</b>    | Qurious 에서 자체적으로 제작하여 제공하는 문제들을 말한다.   |
| <b>외부 문제 은행</b> | Qurious 외의 알고리즘 문제를 제공하는 다른 사이트들을 말한다.<br>현재 버전에서 외부 문제 은행은 백준과 해커랭크만 해당한다.  |
| <b>외부 문제</b>    | 외부 문제 은행에서 제공하는 문제들이다.   |
| <b>Qurious</b>  | 본 서비스를 제공하는 시스템의 이름이다.   |
| <b>사용자</b>      | Qurious 를 서비스를 이용하여 학습하는 사람을 말한다.  |
| <b>진단고사</b>     | 사용자의 컴퓨터 지식 및 python 지식 수준을 측정하기 위해 Qurious 에서 자체적으로 제작하여 제공하는 온라인 시험이다. 본 서비스를 처음 사용하는 사용자에게 문제를 추천할 때 사용할 데이터를 얻기 위해 제공한다. |
| <b>문제 유형</b>    | 알고리즘, python 언어, 자료구조별로 컴퓨팅 지식을 분야별로 나누어 놓은 것을 의미한다.   |
| <b>난이도</b>      | Qurious 가 자체적으로 문제유형별, 실력수준별로 나누어 놓은 문제의 수준을 분류한 것이다.  |
| <b>크롤러</b>      | 외부 문제 은행의 정보를 수집하여 Qurious 에게 제공하는 서브시스템이다.  |

**Table 1 용어 정의**

### 3.3 Acronyms and Abbreviations

| 용어         | 정의   |
|------------|--|
| <b>URL</b> | Uniform Resource Locator. 웹 문서의 각종 서비스를 제공하는 서버들에 있는 파일의 위치를 표시하는 표준을 말한다. |
| <b>DB</b>  | Data Base. 여러 사람에 의해 공유되어 사용될 목적으로 통합하여 관리되는 데이터의 집합을 말한다.                 |

**Table 2 약어**

## 4 User Requirement 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

### A. 회원가입

사용자는 서비스를 이용하기 위해 회원 정보를 입력하고, 회원 가입을 해야 한다. 회원 정보는 각 사용자를 식별할 수 있어야 하며, 데이터베이스에 저장된다. 회원 정보로는 이메일, 비밀번호, 이름, 알고리즘 서비스 계정 정보가 있다.

### B. 로그인

사용자가 서비스를 이용하기 위해 이메일과 비밀번호를 입력하고 사용 권한을 얻을 수 있어야 한다. 로그인 후에 사용자를 위한 문제를 추천받고, 풀었던 문제를 다시 볼 수 있다. 로그인 없이는 서비스를 이용할 수 없다.

### C. 레벨 테스트

처음 본 서비스를 사용하는 사용자에게 제공하는 기능이다. 처음 이용하는 경우, 문제를 추천해 줄 사용자 정보가 전무하므로, 진단고사를 통해 초반 문제 추천을 진행한다. 사용자는 본 서비스에서 제작하여 제공하는 코딩 문제를 난이도를 자의적으로 선택하여 진단고사를 푼다. 사용자가 진단고사를 응시하고 채점한 후, 사용자의 수준을 진단하여 외부 문제를 추천한다.

### D. 문제 추천 받기

로그인을 하고, 진단고사를 치른 사용자에게 제공하는 기능이다. 사용자는 자신의 현재 수준에 맞는 문제들을 추천 받을 수 있다. 문제는 내부 문제, 외부 문제로 나누어 진다. 일정 수 이상의 문제를 다 푼 사용자는 추가적으로 문제를 추천 받을 수 있다.

### E. 내부 문제 풀기

사용자가 추천받은 문제들 중에서 Quirious 자체 문제를 골라 이를 직접 풀어보고 채점 받을 수 있는 기능이다. Quirious 자체 문제를 클릭하면 사용자는 문제 풀기 페이지로 넘

어가고 거기서 문제 설명, 입력 값, 출력 값을 보고 코드를 직접 쳐서 문제를 풀게 된다. 이때 문제를 풀 때 코드 에디터를 통해 편리하게 코드를 작성할 수 있고 제출하기 버튼을 누르면 결과화면을 통해 자신의 코드에 대한 진단을 받을 수 있다.

#### F. 외부 문제 풀기

사용자가 외부 문제은행(백준, 해커랭크)에서 사용자 맞춤 문제를 추천받고 외부 문제은행 사이트로 이동할 수 있는 기능이다. 사용자가 진단고사를 풀고 오면, 사용자의 부족한 개념과 실력을 고려하여 외부 문제를 추천한다. 사용자가 외부 문제를 풀고 다시 Qurious 사이트로 돌아오면 체크박스에 체크한다. 인증절차를 거친 후 풀었다고 확인된 문제가 3개 이상이 되면 다시 외부 문제를 추천한다. 이 때 이미 푼 문제는 다시 추천하지 않는다.

#### G. 마이페이지

로그인한 사용자를 위한 개인화 기능이다. 마이페이지 기능에서 회원정보를 확인하여 비밀번호를 수정할 수 있다. 사용자는 마이 페이지에서 자신이 푼 문제들을 확인할 수 있다. 문제 리스트에 들어가면, 자신이 어떤 문제를 풀었고, 몇 회 시도했는지 확인할 수 있다. 푼 문제는 최신순으로 정렬되어 있다.

#### H. 랭킹 확인

사용자가 개인 랭킹과 전체 랭킹을 확인할 수 있는 기능이다. 개인 랭킹은 사용자가 속한 등급에서 등수로(혹은 백분위로) 사용자의 실력을 나타낸 것이고, 전체 랭킹은 전체 사용자들의 등수, 닉네임, 푼 문제와 제출한 문제 개수, 정답 비율을 보여준다. 전체 랭킹은 Qurious의 전체 사용자의 푼 내부분제를 기준으로 등수가 매겨진다.

#### I. 내부 문제 관리

관리자는 내부 문제를 생성할 수 있다. 내부 문제를 생성할 때 문제 설명, 입출력에 대한 설명, 예시 입력, 예시 출력을 입력하고 테스트 케이스를 zip로 압축하여 업로드하여 등록한다.

## J. 외부 문제 관리

관리자는 외부 백준, Hackerrank 문제를 추가하고 이를 관리할 수 있다. 문제를 추가할 때 해당 문제의 출처, 난이도, 분류, 링크, 문제 번호, 문제 설명을 입력하고 이를 업로드한다.

## 4.3 Nonfunctional requirement

### A. Product Requirement

#### A.1 Usability Requirement (UI)

직관적이고 이해하기 쉬운 UI를 갖추어야 한다. 정답을 입력하는 UI는 가독성이 높아야 한다. 문제 리스트를 보여줄 때는 여러 문제를 한 번에 가독성 있게 볼 수 있도록 해야 한다.

#### A.2 Efficiency Requirement

문제 추천 모델은 하루에 한 번 업데이트가 이루어져야 한다. 진단고사를 본 후에 문제가 추천되어 사용자에게 보여지는데 걸리는 시간은 너무 느리지 않아야 하며, 진단고사 추천을 제외한 모든 페이지의 응답속도 또한 너무 느리지 않아야 한다.

#### A.3 Dependability Requirement

회원가입 시 입력한 사용자의 데이터가 제대로 저장되어야 한다. 올바른 사용자 정보를 입력했을 경우 로그인에 성공할 수 있어야 한다. 문제 풀이 후 제출된 코드는 손상되지 않아야 하며, 정답 유무를 체크한다. 정답일 경우, 풀 문제로 기록되고, 이는 추천할 때 사용된다. 추천되는 문제는 사용자의 실력수준에 맞춰 추천되어야 한다.

#### A.4 Security Requirement

사용자의 개인정보는 암호화되어 보관해야 한다. 사용자의 비밀번호는 curious측에서도

정확하게 알 수 없으며, 로그인 시에 사용자를 식별하는데만 사용된다. 외부 접근으로 인해 사용자의 정보가 유출되는 일이 없어야 한다.

## B. Organization Requirement

### B.1. Environmental Requirement

Qurios 서비스는 온라인 웹서비스이며, PC와 모바일 모두 이용할 수 있다. 사용자의 사용 환경에 관계 없이 이용에 불편이 없도록 하는 것이 목표이나, 주 사용 환경은 PC를 전제로 한다.

### B.2. Operational Requirement

Vue.js, Django, Docker를 이용해 개발한다. 개발 언어는 Python, JavaScript이며, Qingdao University의 Opensource를 채점과 개발에 이용한다.

### B.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.

## C. External Requirement

### C.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.



### C.2. ethical requirement

본 서비스는 이용 연령 제한이 없다.

### C.3. safety/security requirement

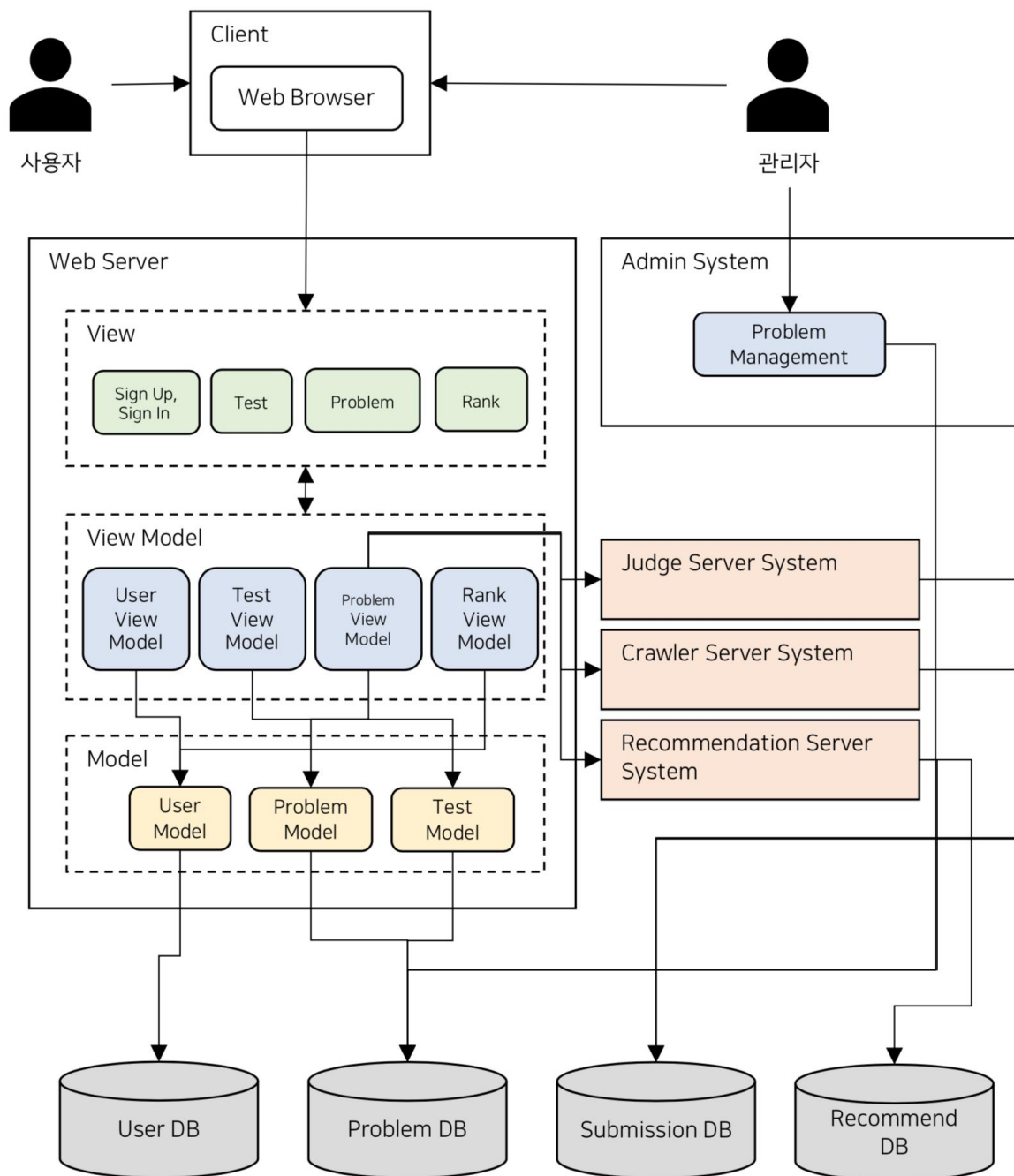
개인정보보호법에 따라 고객 개인정보를 안전하게 보관하고 보호하여야 한다.

## 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 Qurious System Architecture

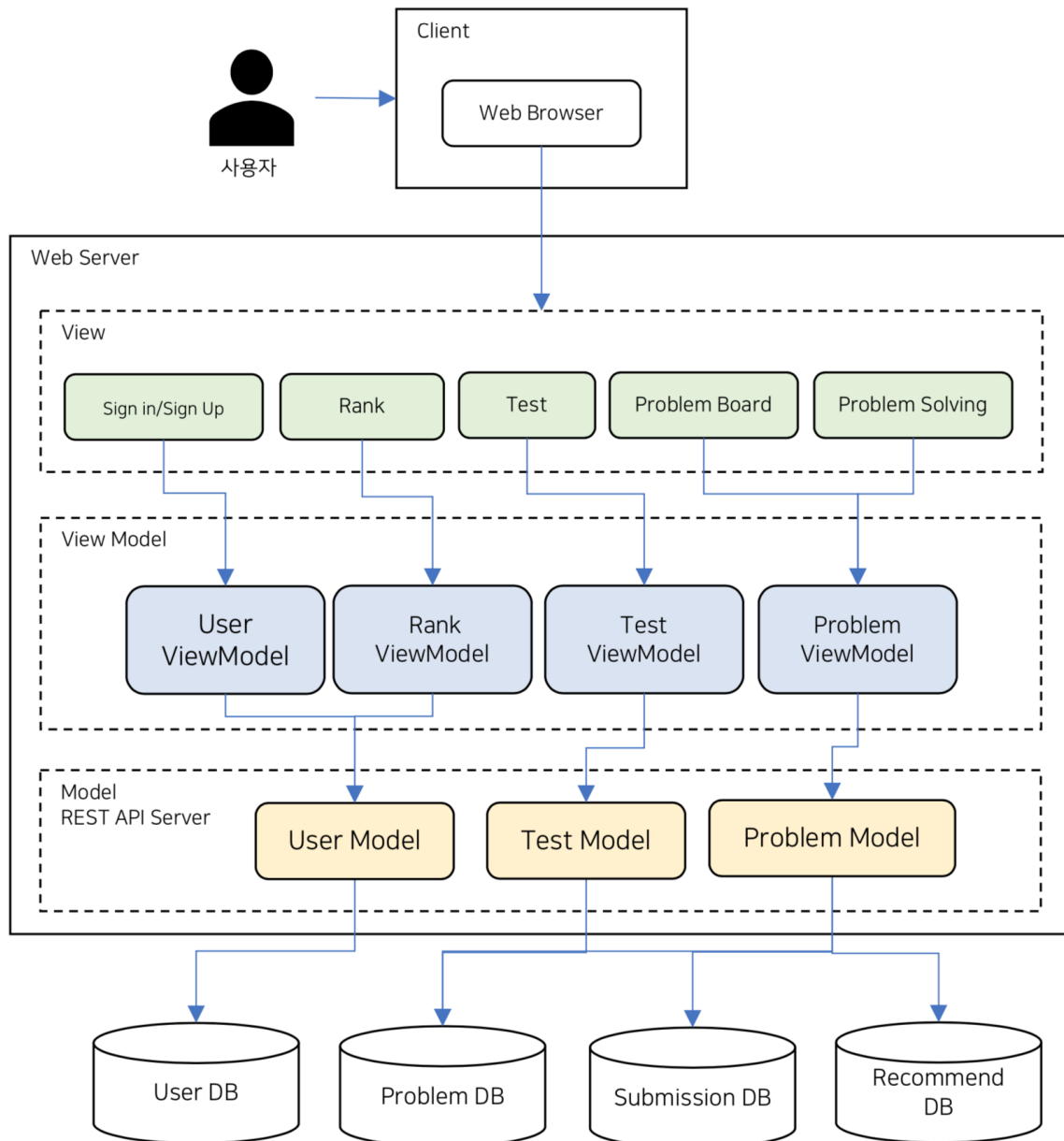


**Diagram 1 System Architecture: Qurious 전체**

Qurious의 전체 시스템 구조는 Web Application 구조로 Web Browser로 접속하며 크게 Web Server, Judge Server, Crawler Server, Recommendation Server 시스템으로 구성된다.

### 5.3 Sub system

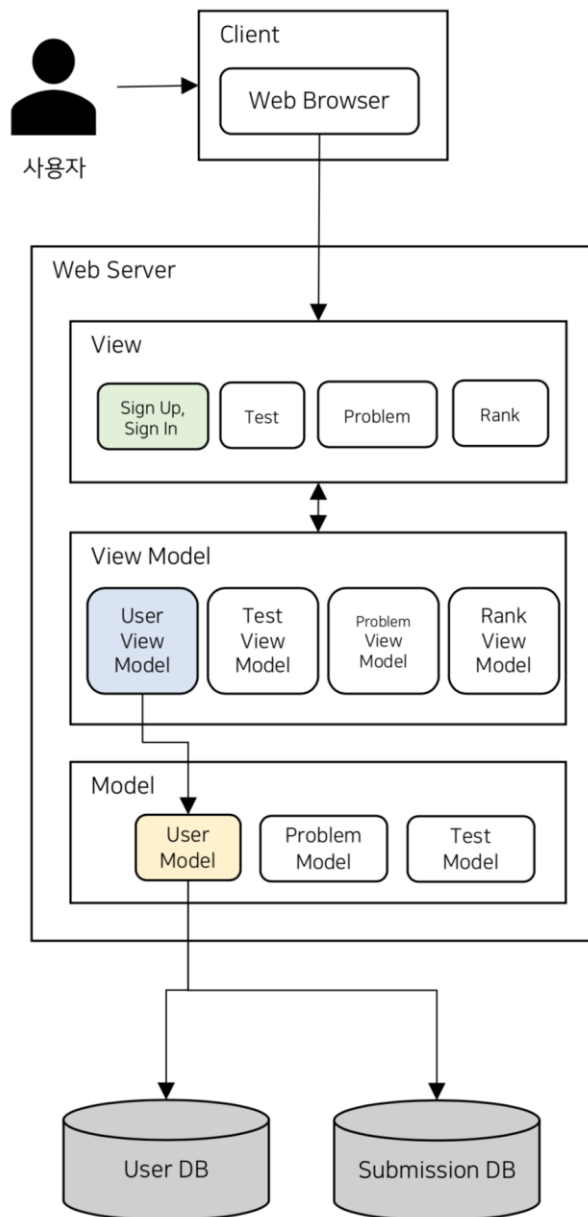
#### A. 웹(메인)서버 시스템 상세



**Diagram 2 System Architecture: 웹(메인)서버 시스템 상세**

Web Server 는 크게 View, View Model, Model (MVVM) 구조를 지니는 것을 상정한다. 각 View는 보여지는 페이지를 구성하며 View Model을 통해 데이터를 바인딩받는다. 각각의 ViewModel은 Model을 통해 기능을 수행하고 DB에 접근한다.

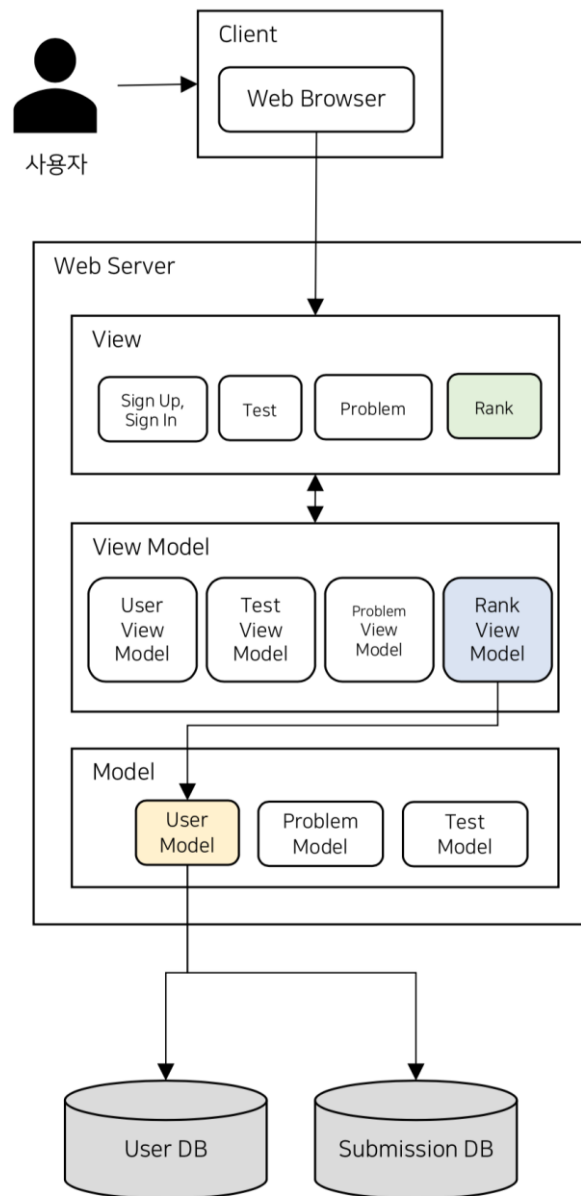
## B. 사용자 시스템



**Diagram 3 System Architecture: 사용자 시스템**

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

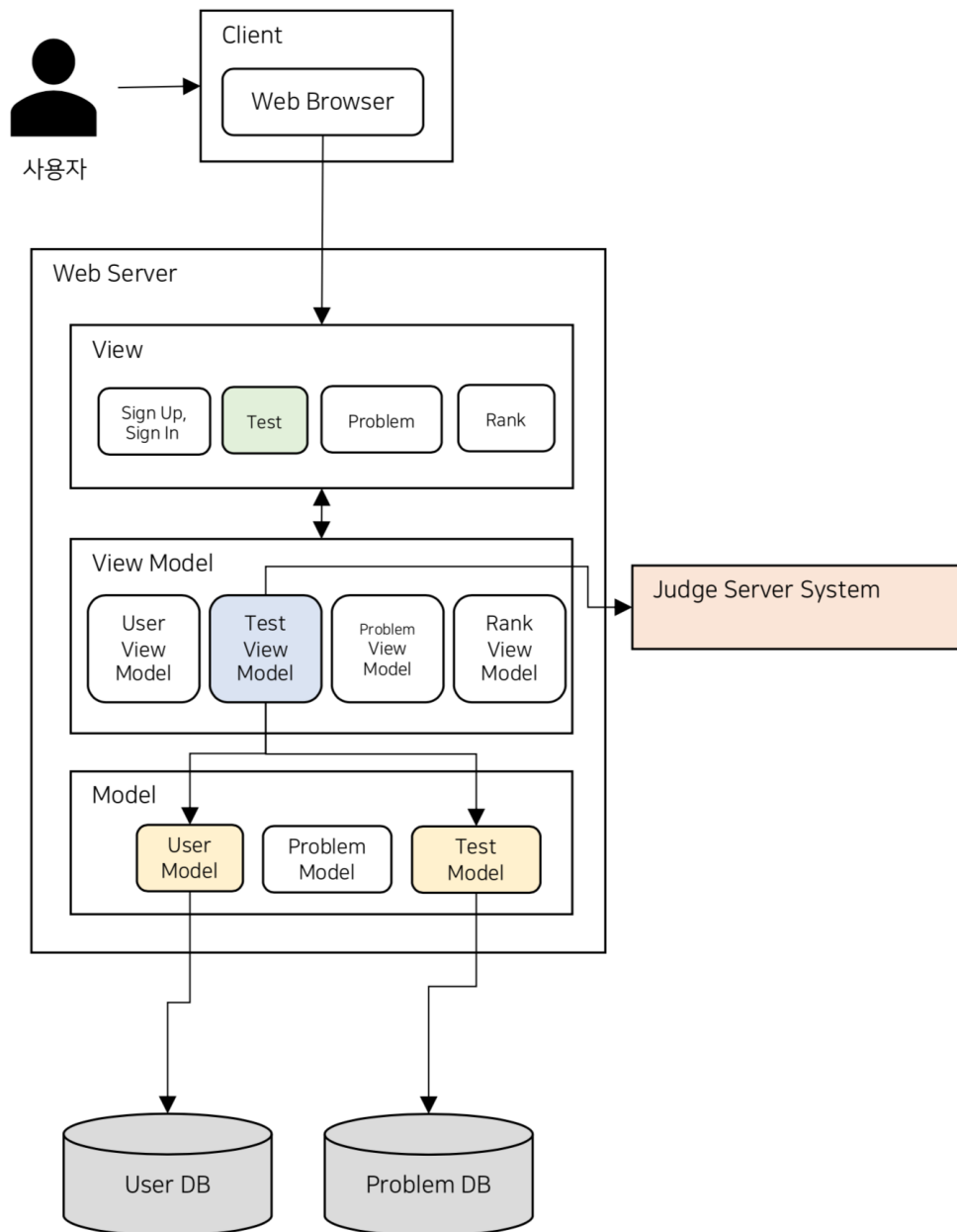
### C. 랭킹시스템



**Diagram 4 System Architecture: 랭킹 시스템**

랭킹 시스템은 Rank View를 통해 표시되며 이때 Rank ViewModel이 User Model에서 회원의 랭크를 가져와 이를 변환하고 View에 데이터를 전달한다.

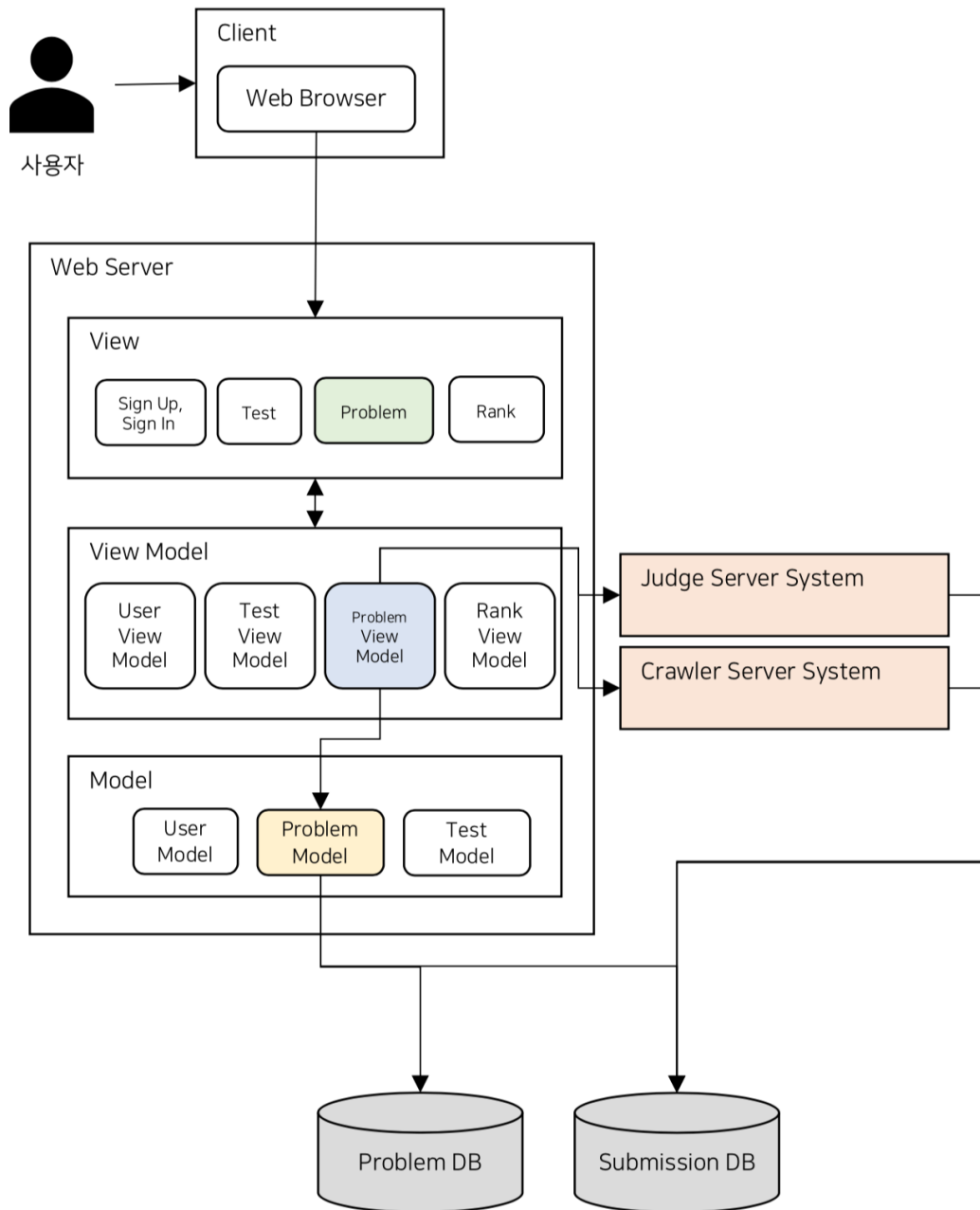
#### D. 진단고사 시스템



**Diagram 5 System Architecture: 진단고사 시스템**

진단고사 시스템은 Test View를 통해 표시되며 여기서 Test Viewmodel이 보여질 문제를 선정하여 전달한다. 최종 제출 이후 체점은 Judge Server System을 통해 이루어지고 결과를 반환받아 계산하여 최종 실력 점수를 UserDB에 저장한다.

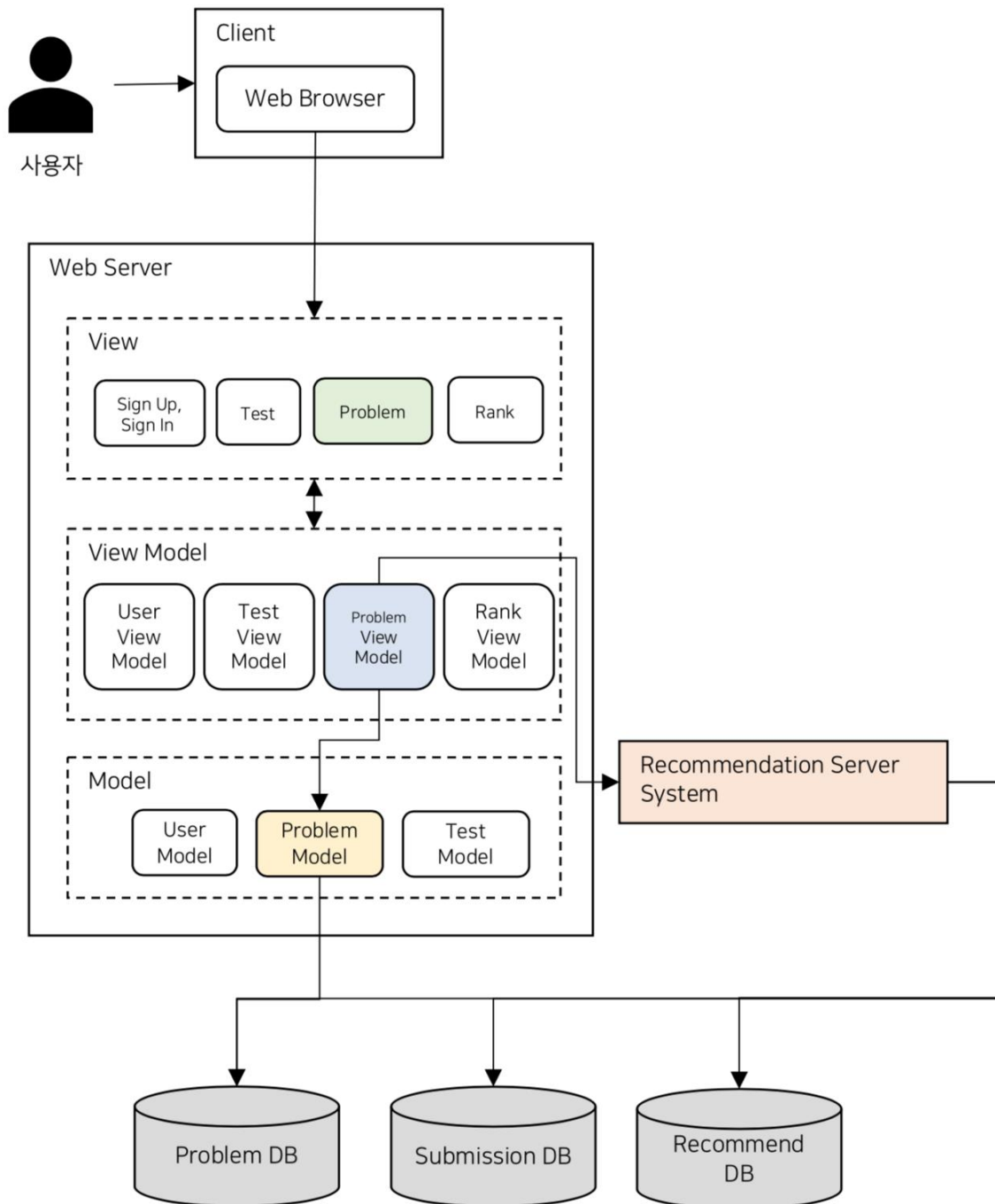
## E. 문제 풀이 시스템



**Diagram 6 System Architecture: 문제 풀이 시스템**

문제 풀이 시스템은 Problem View를 통해 표시되며 Problem View Model을 통해 문제 풀이 기능을 구현한다. 내부 문제의 경우 Judge Server System을 통해, 외부 문제의 경우 Crawler Server System을 통해 문제를 채점하고 이를 Submission DB에 저장한다.

## F. 문제 추천 시스템

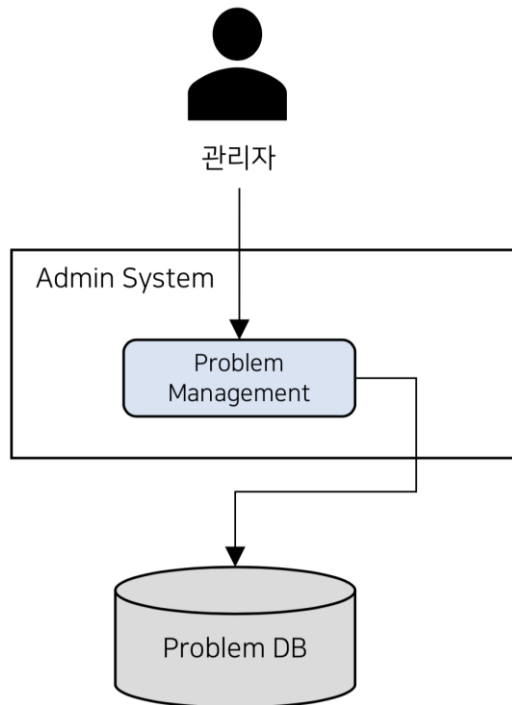


**Diagram 7 System Architecture: 문제 추천 시스템**

문제 추천 시스템은 Problem View 와 ViewModel를 통해 이루어지며 추천하기 버튼 클릭시 Recommendation Server System으로 요청을 보낸다. 시스템 내 협업 필터링 모델에서 사용자의 Submission데이터를 입력받아 문제를 추천하고 이를 문제 ID로 반환한다.



## G. 문제 관리 시스템



**Diagram 8 System Architecture: 문제 관리 시스템**

문제 관리 시스템은 관리자가 Admin System으로 접속하여 문제를 추가하고, 삭제 및 관리할 수 있으며 이러한 변경사항은 Problem DB에 반영된다.

## 6 System Requirement 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

### A. 회원가입

| 항목 | 설명   |
|----|--|
| 기능 | 회원가입   |
| 설명 | 회원 가입을 위해 사용자 정보를 입력받고, 데이터베이스에 사용자 정보를 저장한다.  |
| 입력 | 이메일, 비밀번호, 이름, 외부 알고리즘 서비스명, 외부 알고리즘 서비스 계정  |
| 출력 | 이메일 중복 체크 후 결과 메시지 출력<br>외부 알고리즘 서비스 계정 정보 존재 여부 확인 후 결과 메시지 출력<br>입력 양식을 모두 작성했는지 확인 후, 조건 충족 여부 출력<br>회원가입 결과 출력 |
| 처리 | 회원 가입 후 데이터베이스에 회원 정보 저장   |
| 조건 | 이메일: ' 아이디 문자열 @ 도메인 문자열 ' 형식이며, 중복되어서는 안된다.<br>비밀번호는 8~16자의 문자열 형식이다.<br>외부 알고리즘 서비스는 백준, Hackerrank로 한정한다.       |

**Table 3 Functional Requirement: 회원가입**

### B. 로그인

| 항목 | 설명   |
|----|--|
| 기능 | 로그인  |
| 설명 | 사용자가 서비스를 이용하기 위해 이메일과 비밀번호를 입력하고 사용 권한을 얻을 수 있어야 한다. 로그인 후에 사용자를 위한 문제를 추천받고, 풀었던 문제를 다시 볼 수 있다. 로그인 없이는 서비스를 이용할 수 없다. |
| 입력 | 사용자의 이메일과 비밀번호를 입력한다.  |
| 출력 | 로그인에 성공했을 경우 메인 화면으로 redirect한다.   |

|           |   |
|-----------|---|
|           | 로그인에 실패했을 경우, 다시 시도하라는 메시지를 출력하고 사용자 정보 입력창은 초기화된다.   |
| <b>처리</b> | 이메일 형식과 비밀번호 형식을 지켰는지 확인하고, 잘못된 입력일 경우 경고 문구를 출력한다.<br><br>형식을 지켰을 경우, 로그인을 시도한다.<br><br>데이터베이스에 저장된 회원 정보와 대조해 로그인에 성공 여부를 확인한다. |
| <b>조건</b> | 이메일은 ' 아이디 문자열 @ 도메인 문자열 ' 형식이다.<br><br>비밀번호는 4~12자의 문자열 형식이다.<br><br>여러번 로그인 시도할 경우, 최초 한 번의 시도만 서버에 요청한다.                       |

**Table 4 Functional Requirement: 로그인**

### C. 진단고사 응시

| 항목        | 설명  |
|-----------|---|
| <b>기능</b> | 진단고사 응시   |
| <b>설명</b> | 사용자의 수준에 맞는 문제를 추천하기 위해, 본 서비스를 처음 사용하는 사용자의 수준을 진단한다. 사용자가 진단고사 응시 버튼을 누르면 진단고사 난이도를 초급, 중급, 고급 중 하나를 선택할 수 있다. 사용자는 난이도를 하나 선택하여 진단고사를 응시한다. 코드를 작성하거나 선택지를 체크할 수 있다. 사용자가 진단고사에 응답한 데이터를 저장한다. 사용자의 응답이 적절한지 확인한다. |
| <b>입력</b> | 각 문제별 사용자의 응답   |
| <b>출력</b> | 사용자의 실력 수준  |
| <b>처리</b> | 사용자의 응답을 채점 서버로 전달한다.   |
| <b>조건</b> | 1. 로그인한 상태여야 한다<br><br>2. 사용자는 모든 문항에 응답하여야 한다. 공백은 응답으로 간주하지 않는다.  |

**Table 5 Functional Requirement: 진단고사 응시**

#### D. 진단고사 채점

| 항목 | 설명  |
|----|---|
| 기능 | 진단고사 채점   |
| 설명 | 진단고사 응시 기능이 전달한 사용자의 응답 데이터를 채점서버가 채점 결과를 반환한다. 채점 서버의 출력값에 따라 정답률을 계산한다. 사용자의 응답의 정답 유무를 확인한다. 객관식 문제의 경우, 사용자가 정답을 맞았는지 확인한다. 정답 선택지를 선택했을 경우에만 정답으로 인정한다. 그 외의 선택지를 선택했을 경우, 오답으로 처리한다. 텍스트로 코드를 작성한 문제의 경우, 사용자가 응답한 정보를 judger를 이용해 채점한다. Accepted가 결과로 나왔을 경우, 정답으로 인정한다. 그 외 Compile error, Runtime error, Time Limit Exceeded, Memory Limit Exceeded, Wrong Answer가 나왔을 경우 오답으로 처리한다. |
| 입력 | 채점 서버의 출력값  |
| 출력 | 사용자의 실력 수준  |
| 처리 | 문제 수와 Accepted 수의 비율을 정답률로 계산한다. 정답률에 따라 사용자의 실력 수준이 리턴된다. 진단고사 기반 추천 기능이 맞는 문제를 추천한다.  |
| 조건 | 1. 채점 서버에서 사용자의 응답을 전달받은 상태여야 한다.<br>2. 인터넷이 연결된 상태여야 한다.   |

**Table 6 Functional Requirement: 진단고사 채점**

#### E. 문제 추천 받기

| 항목 | 설명   |
|----|--|
| 기능 | 문제 추천  |
| 설명 | 로그인을 하고, 진단고사를 치룬 사용자에게 제공하는 기능이다. 사용자는 자신의 현재 수준에 맞는 문제들을 추천 받을 수 있다. 문제는 내부 문제, 외부 문제로 나누어 진다. 일정 수 이상의 문제를 다 푼 사용자는 추가적으로 문제를 추천 받을 수 있다. |

|    |  |
|----|--|
| 입력 | 진단고사 결과 + 사용자 문제풀이 데이터   |
| 출력 | 문제 5개(내부문제+외부문제)   |
| 처리 | 사용자의 진단고사 결과와 문제풀이 데이터를 추천 서버에 보낸다. 추천서버에서 해당하는 문제들을 문제 DB로부터 결과로 반환한다.  |
| 조건 | 1. 사용자는 진단고사를 본 상태여야 한다.<br>2. 외부 문제풀이 데이터를 받기 위해서는 사용자별로 해당 사이트에 계정과 식별할 수 있는 문제풀이 내역이 있어야 한다.<br>3. 사용자가 추가로 문제를 추천 받기 위해서는 3문제 이상 풀어야 한다. |

**Table 7 Functional Requirement: 문제 추천 받기**

F. 추천 서버에서 문제 추천하기

| 항목 | 설명   |
|----|--|
| 기능 | 추천 서버에서 문제 추천하기  |
| 설명 | 추천 서버에서 협업 필터링 모델을 이용해 사용자에게 맞춤형 문제를 제공한다. 협업 필터링은 문제 풀이 데이터를 이용해 하루주기로 학습과 업데이트를 진행하며 이를 기반으로 사용자의 문제 풀이 데이터를 입력 받아 추천한다. 이때 이전 문제 풀이 데이터를 탐색해 해당 난이도에서 10문제 이상을 맞췄을 경우 다음 난이도의 문제를 추천한다. |
| 입력 | 사용자의 문제 풀이 데이터   |
| 출력 | 추천 문제 5개의 문제DB ID  |
| 처리 | 해당 문제 풀이 데이터와 이전 문제 풀이 데이터를 조합하여 등급에 맞는 새로운 문제를 추천한다.  |
| 조건 | 문제 풀이 데이터는 최소 3개이상 있어야 함   |

**Table 8 Functional Requirement: 추천 서버에서 문제 추천하기**

#### G. 진단고사 기반 추천

| 항목 | 설명   |
|----|--|
| 기능 | 진단고사 기반 추천   |
| 설명 | 문제 풀이 데이터가 존재하지 않고 진단고사를 푼 이후 실력수준만 나온 경우, 그 수준에 맞게 문제 5개를 골라 추천해준다. 이때 문제는 미리 선정되어진 문제를 바탕으로 하며 같은 수준으로 나온 사용자는 같은 문제를 추천받게 된다. |
| 입력 | 진단고사에서 나온 실력수준   |
| 출력 | 추천 문제 5개의 문제DB ID  |
| 처리 | 실력 수준에 맞게 문제 DB에서 문제 DB ID를 뽑아 출력한다.   |
| 조건 | <ol style="list-style-type: none"> <li>1. 이전 문제 풀이 데이터가 없어야함</li> <li>2. 진단 고사 점수가 존재해야함</li> </ol>                              |

**Table 9 Functional Requirement: 진단고사 기반 추천**

#### H. 내부 문제 풀기

| 항목 | 설명  |
|----|---|
| 기능 | 내부 문제 풀기  |
| 설명 | 사용자가 추천받은 문제들 중에서 Quirious 자체 문제를 골라 이를 직접 풀어보고 채점 받을 수 있는 기능이다. Quirious 자체 문제를 클릭하면 사용자는 문제 풀기 페이지로 넘어가고 거기서 문제 설명, 입력 값, 출력 값을 보고 코드를 직접 쳐서 문제를 풀게 된다. 이때 문제를 풀 때 코드 에디터를 통해 편리하게 코드를 작성할 수 있고 제출하기 버튼을 누르면 결과화면을 통해 자신의 코드에 대한 진단을 받을 수 있다. |
| 입력 | 해당 문제 클릭하고 Python3 코드를 입력   |

|    |  |
|----|--|
| 출력 | 성공, 실패,에러(컴파일 에러 / 런타임 에러)                               |
| 처리 | 입력한 답안을 채점서버에 보낸다. 채점 서버에서 해당 코드를 컴파일 및 검사를 하고 결과를 반환한다. |
| 조건 | 사용자는 진단 고사를 본 상태로 진단 점수가 있어야 한다.                         |

**Table 10 Functional Requirement: 내부 문제 풀기**

#### I. 외부 문제 풀기

| 항목 | 설명   |
|----|--|
| 기능 | 외부문제 풀기  |
| 설명 | 사용자의 실력에 맞춰 외부 문제은행 사이트의 문제를 추천받는 기능이다. 한번 추천받은 문제는 바뀌지 않으며, 3문제 이상 풀 경우에 새로운 문제들을 추천받을 수 있다.  |
| 입력 | 사용자가 진단고사를 풀고 온다(사용자의 등급 정보가 있다).<br>사용자가 문제를 풀면 체크박스를 클릭한다.   |
| 출력 | 사용자가 진단고사를 풀지 않았을 경우 '진단고사를 풀면 문제를 추천받을 수 있습니다'라는 문구를 출력한다.<br><br>외부문제 데이터베이스에서 사용자의 등급과 일치하는 등급의 문제들을 추천 문제로 하며, 추천문제로 연결되는 링크를 출력한다.<br><br>추천한 외부 문제를 푼 경우 체크표시가 성공적으로 된다. 문제를 풀지 않은 경우 문제를 풀고 오라는 경고창을 띄운다. |
| 처리 | 사용자가 푼 외부 문제를 채점 데이터베이스에 저장한다.<br><br>문제를 추천할 때, 채점 데이터베이스에 있는 문제는 추천에서 제외한다.<br><br>사용자가 체크박스를 클릭하면 실제 문제를 풀었는지 확인한다. 풀었다면 체크박스는 정상적으로 체크된다.  |
| 조건 | 사용자의 진단고사 결과가 있다.<br><br>사용자 데이터베이스에 해당 외부문제은행 계정 정보(아이디, 비밀번호)가 있   |

|  |    |
|--|----|
|  | 다. |
|--|----|

**Table 11 Functional Requirement: 외부 문제 풀기**

J. 마이페이지

| 항목 | 설명   |
|----|--|
| 기능 | 마이페이지  |
| 설명 | 비밀번호 변경 버튼을 누르면 계정 비밀번호를 수정할 수 있다. 변경 버튼을 눌러 비밀번호 변경 요청을 하면, 비밀번호 조건을 확인한 후 사용자 데이터 베이스에 변경된 비밀번호가 저장된다.<br><br>사용자는 푼 문제들을 확인할 수 있다. 문제 리스트를 누르면, 푼 문제와 시도 횟수를 표시한다. 푼 문제는 최신순으로 정렬되어 있다. |
| 입력 | 마이페이지 클릭   |
| 출력 | 마이페이지 화면   |
| 처리 |  |
| 조건 | 새로운 비밀번호는 비밀번호 조건(문자열 4~12자)을 만족시켜야 한다.  |

**Table 12 Functional Requirement: 마이페이지**

K. 랭킹확인

| 항목 | 설명  |
|----|---|
| 기능 | 랭킹 확인   |
| 설명 | 사용자가 개인 랭킹과 전체 랭킹을 확인할 수 있는 기능이다.   |
| 입력 | 랭킹 확인 페이지를 클릭한다.  |
| 출력 | 사용자의 등급 정보가 있을 경우, 개인 랭킹 정보를 출력한다. 없을 경우 '진단고사를 풀면 자신의 랭킹을 확인할 수 있습니다' 라는 문구를 출력한다. |
| 처리 | 개인 랭킹은 같은 등급인 사용자들의 수와 그 중 로그인한 회원의 등수와 출   |



|           |   |
|-----------|---|
|           | <p>력한다.</p> <p>전체 랭킹은 사용자의 등급 유무 상관없이 출력한다. Quirious 전체 사용자의 등수, 사용자 닉네임, 푼 문제 수, 제출한 문제 수, 정답 비율을 출력한다.</p> <p>전체 사용자의 푼 문제 수 데이터베이스를 참조한다.</p> <p>사용자의 푼 문제 수를 내림차순 하여 등수를 매긴다.</p> |
| <b>조건</b> | <p>Quirious의 사용자가 내부 문제를 풀 때마다 랭킹은 바뀔 수 있다.</p> <p>사용자가 Quirious에서 푼 문제가 있어야한다.</p>   |

**Table 13 Functional Requirement: 랭킹 확인**

#### L. 내부 문제 관리

| 항목        | 설명  |
|-----------|---|
| <b>기능</b> | 내부 문제 관리하기  |
| <b>설명</b> | 관리자는 내부 문제를 추가하고 수정하고 삭제할 수 있는 권한이 있으며 이를 어드민 페이지에서 할 수 있다. 어드민 페이지에서 내부 문제 추가하기 페이지를 열면 문제에 대한 설명, 입력값에 대한 설명, 출력값에 대한 설명, 입력 예시, 출력 예시를 입력 해야하고 각종 제약사항(메모리, 실행시간)을 넣을 수 있다. 해당 문제가 어떤 난이도를 지니는지 입력해야한다. 테스트 케이스를 만들어 넣어야하며 이때 테스트 케이스는 zip파일 형태로 압축되어 있어야한다. |
| <b>입력</b> | 문제ID, 문제 설명, 입출력에 대한 설명, 예시 입력, 예시 출력, 테스트 케이스 zip파일  |
| <b>출력</b> | 성공, 에러 메시지  |
| <b>처리</b> | 해당 문제를 문제 DB에 저장하고, 추천 서버에서 이 문제를 추천할 수 있도록 한다.   |
| <b>조건</b> | 문제 ID가 이전에 등록된 적이 없어야 한다.   |

**Table 14 Functional Requirement: 내부 문제 관리**

## M. 외부 문제 관리

| 항목 | 설명   |
|----|--|
| 기능 | 외부 문제 관리   |
| 설명 | 관리자는 어드민 페이지 내 외부 문제 추가하기를 열면 먼저 추가된 문제 들을 리스트 형식으로 볼 수 있다. 상단에 문제 추가하기 버튼을 누르면 문제를 추가하는 페이지로 이동한다. 페이지는 문제 제목, 문제 URL, 문제 유형, 문제의 난이도 등 문제에 대한 정보를 입력할 수 있는 입력란이 있다. 해당 입력란을 기입한 후 추가하기 버튼을 누르면 문제가 정상적으로 추가되고 추가된 문제는 문제 데이터 베이스에 추가되고 추천 서버에서 활용할 수 있다. |
| 입력 | 문제의 출처, 난이도, 분류, URL, 문제 번호, 문제 설명   |
| 출력 | 성공, 에러 메시지   |
| 처리 | 입력값을 외부 문제 DB에 저장하고  |
| 조건 | 추가하려는 문제는 이전에 추가되지 않은 문제여야 한다. 문제 추가는 관리자만 할 수 있다.   |

**Table 15 Functional Requirement: 외부 문제 관리**

## 6.3 Nonfunctional Requirement

### A. Product Requirement

#### A.1. usability requirement (UI)

직관적이고 이해하기 쉬운 UI를 갖추어야 한다. 진단고사의 경우 한 문제씩 제시해 문제 하나에 집중할 수 있도록 해야 한다. 정답 입력하는 UI는 가독성을 위해 배경색은 흰색, 코드는 검은색을 기본으로 하며 syntax highlighting을 지원해야 한다. 문제 리스트를 보여줄 때는 테이블 구조로 여러 문제를 한 번에 가독성 있게 볼 수 있도록 해야한다.

#### A.2 Efficiency Requirement

문제 추천 모델은 하루에 한 번 업데이트가 이루어져야한다. 진단고사를 본 후에 문제가 추천되어 사용자에게 보여지는데 걸리는 시간은 4sec를 넘지 않아야 한다. 진단

고사 추천을 제외한 모든 페이지의 응답속도는 3sec를 넘겨서는 안된다.

### A.3 Dependability Requirement

회원가입시 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 추천 데이터로 사용이 되어야 한다. 추천되는 문제는 사용자의 실력수준과 2단계 이상 벗어나지 않는 선에서 제공되어야 한다.

### A.4 Security Requirement

사용자의 개인정보는 암호화되어 보관해야 한다. 사용자의 비밀번호는 curious측에서도 정확하게 알 수 없으며, 로그인시에 사용자를 식별하는데만 사용된다. 사용자의 개인정보는 다른 사용자에게 노출되지 않아야 한다.

외부 접근으로 인해 사용자의 정보가 유출되는 일이 없어야 한다. 각 문제의 test case는 사용자가 test case에만 맞춘 코드를 작성하지 않도록 일부만 노출되어야 한다.

## B. Organization Requirement

### B.1 Environmental Requirement

- Curious 서비스는 온라인 웹서비스이며, PC와 모바일 모두 이용할 수 있다. 사용자는 인터넷이 연결된 상태에서 사용하여야 한다. 크롬, 익스플로러에서 사용할 수 있다. 사용자의 사용 환경에 관계 없이 이용에 불편이 없도록 하는 것이 목표이나, 주 사용 환경은 PC를 전제로 한다. 개발 시 우선순위도 PC 사용자를 우선순위로 둔다.

### B.2 Operational Requirement

Vue.js, Django, Docker를 이용해 개발한다. 개발 언어는 Python, JavaScript이며, Qingdao University의 Opensource를 채점과 개발에 이용한다.

### B.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. Django 프레임워크를 통해 호환성을 높인다.

## C. External Requirement

### C.1 Regulatory requirement / Legislative requirement

- (저작권을 확인해야 한다. 회원 정보를 외부에 제공할 경우 정보 제공 동의를 얻어야 한다.)
- 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. 사용자 정보는 협업 필터링에 사용될 수 있으나 익명으로 처리된다.

### C.2 Ethical Requirement

본 서비스는 이용 연령 제한이 없다.

### C.3 Safety/Security Requirement

- 개인정보보호법에 따라 고객 개인정보를 안전하게 보관하고 보호하여야 한다.

## 6.4 Scenario

### A. 회원가입

#### A.1 Initial Assumption

사용자는 적어도 한 개 이상의 이메일 계정을 가지고 있어야 하며, 백준 혹은 hackerrank의 계정을 가지고 있어야 한다.

## A.2 Normal Flow of Events

사용자는 로그인 되지 않은 상태에서 메인 페이지에 접근할 경우, 로그인을 요청받는다.

로그인 화면에서 회원 가입 버튼을 클릭해 회원 가입 화면으로 이동한다.

회원 가입 화면에서 이메일, 비밀번호, 이름, 외부 알고리즘 서비스명, 외부 알고리즘 서비스 계정을 입력한다.

모든 정보를 입력한 후에 회원가입을 요청한다.

## A.3 What can go wrong

중복된 이메일이 있을 경우, 해당 이메일로는 가입하지 못한다.

이메일 형식을 지키지 못한 경우, 올바른 형식을 입력하도록 유도한다.

비밀번호 형식을 지키지 못한 경우, 올바른 형식을 입력하도록 유도한다.

백준, hackerrank 외의 외부 알고리즘 서비스를 입력하려 할 경우, 다시 입력하도록 한다.

외부 알고리즘 서비스에 등록되지 않은 계정을 입력하려 할 경우, 다시 입력하도록 한다.

## A.4 System State on Completion

모든 정보가 올바르게 입력되고 서버에서 검증작업을 완료한 후에 사용자 데이터베이스에 저장된다.

## B. 로그인

### B.1 Initial Assumption

사용자는 해당 서비스에 회원 가입을 한 상태여야 한다.

## B.2 Normal Flow of Events

사용자는 로그인 되지 않은 상태에서 메인 페이지에 접근할 경우, 로그인을 요청받는다.

로그인 화면에서 이메일, 비밀번호를 입력한다.

입력 후 로그인 버튼을 클릭해 로그인을 요청한다.

## B.3 What can go wrong

이메일 형식을 지키지 못한 경우, 올바른 형식을 입력하도록 유도한다.

비밀번호 형식을 지키지 못한 경우, 올바른 형식을 입력하도록 유도한다.

## B.4 System State on Completion

모든 정보가 올바르게 입력되고 요청을 보내면, 서버에서 사용자 데이터베이스의 정보와 사용자가 요청한 계정 정보를 대조해, 로그인 상태로 저장한다. 사용자에게 access token 을 발급해 서비스를 이용할 수 있도록 한다.

# C. 레벨 테스트

## C.1 Initial Assumption

사용자는 로그인을 한 상태여야 한다. 사용자의 DB에 사용자의 실력 수준에 대한 정보는 없어야 한다. 진단고사의 난이도 선택은 사용자가 스스로 자의적으로 선택한다.

## C.2 Normal Flow of Events

'진단고사 응시하기'를 누르면 진단고사를 응시한다. 사용자는 객관식 혹은 주관식에 맞춰 진단고사를 응시한다. 진단고사 문제에 모두 답을 작성하고 하단의 제출하기 버튼을 누르면 진단고사 응시가 완료된다.

사용자의 응시가 완료되면, 사용자의 응답의 정답 유무를 확인한다. 객관식 문제의 경우, 사용자가 정답을 맞혔는지 확인한다. 정답 선택지를 선택했을 경우에만 정답으로 인정한다. 그 외의 선택지를 선택했을 경우, 오답으로 처리한다. 텍스트로 코드를 작성한 문제

의 경우, 사용자가 응답한 정보를 judger를 이용해 채점한다. Accepted가 결과로 나왔을 경우, 정답으로 인정한다. Compile error, Runtime error, Time Limit Exceeded, Memory Limit Exceeded, Wrong Answer가 나왔을 경우 오답으로 처리한다.

채점이 완료되면, 사용자 정보에 응시한 진단고사의 수준과 정답률, 실력 수준을 저장한다. 추천 기준에는 초급, 중급, 고급과 1,2,3단계를 저장할 수 있다. 추천 기준의 default 값은 사용자가 응시한 진단고사의 수준의 2단계로 한다.

진단고사의 정답률에 따른 실력 수준은 Table1의 기준을 따른다.

사용자에게 추천 기준을 사용자의 현재 실력 수준이라는 결과값으로 보여준다. 실력 수준을 기준으로 기존에 정리해둔 외부 문제들을 난이도별로 추천해준다.

### C.3 What can go wrong

사용자가 진단고사 응시 중 페이지를 나가는 경우, 사용자가 응답한 결과는 사라진다. 사용자는 다시 진단고사를 처음부터 진행하여야 한다.

사용자가 응답을 빠뜨린 문항이 있을 경우, 진단고사가 완료되지 않고 제출버튼을 눌러도 넘어가지 않는다. 이 경우, 모든 문항이 응답되지 않았으므로 성실한 응답을 요구하는 경고창을 띄운다.

사용자가 코드를 작성하는 문제에 응답을 하였으나 스페이스바를 이용한 공백만 작성한 경우, 작성한 답으로 인정하지 않는다. 응답을 빠뜨린 것으로 간주하고 경고창을 띄운다.

### C.4 System State on Completion

사용자의 수준이 사용자 DB에 저장된다.

## D. 문제 추천 받기

### D.1 Initial Assumption

사용자는 로그인을 하고, 진단고사를 본 상태여야 한다.

## D.2 Normal Flow of Events

사용자의 진단고사 결과와 문제풀이 데이터가 추천 서버에 보내져서, 협업 필터링에 기반한 추천 모델에 의해 사용자에게 적합한 문제들이 추천된다. 추천되는 문제들은 각각 자체 문제 태그, 백준 태그, 해커랭크 태그로 나뉘어 진다. 사용자가 3문제 이상을 풀었을 때, '추천 받기' 버튼이 활성화된다. 사용자가 '추천 받기' 버튼을 누르면 새로운 5문제를 추천 받을 수 있다. 사용자가 5문제를 풀면, 버튼 클릭 없이도 자동으로 새로운 5문제를 추천 받을 수 있다.

## D.3 What can go wrong

사용자가 진단고사를 보지 않은 상태이면, 메인 페이지에 '진단고사를 보고, 문제 추천을 받으세요!'라고 표시된다.

## D.4 System State on Completion

메인 페이지에 사용자별로 수준에 맞는 문제들이 5개 표시된다. 가장 위에 나오는 문제가 사용자에게 가장 적합한 문제로, 우선순위가 높은 순으로 내림차순으로 정렬된다.

## E. 내부 문제 풀기

### E.1 Initial Assumption

사용자는 로그인한 상태여야 한다. 사용자는 진단 고사를 본 상태로 진단 점수가 있어야 한다.

### E.2 Normal Flow of Events

사용자는 추천 받은 문제들 중에서 자체 문제 태그로 된다. 이 문제를 클릭하면 문제 풀기 페이지로 이동한다. 해당 페이지에선 문제에 대한 설명과 입력값, 출력값이 적혀있고 사용자가 직접 코드를 작성할 수 있는 코드 에디터가 존재한다. 지원하는 언어는 Python 이고 사용자가 코드를 입력하면 예약어에 대해선 다른 색깔로 하이라이트 표시가 된다. 정답에 맞는 코드를 치고 제출하기 버튼을 누르면 성공했습니다라는 메시지와 함께 문제



추천 페이지로 이동하게 된다. 실패하게 되면 다시 풀어보세요라는 메시지와 함께 문제 풀기 페이지로 리다이렉션된다.

### E.3 What can go wrong

코드를 입력하지 않고 제출하기를 누를 경우 코드를 입력하라는 경고 메시지가 뜬다.

### E.4 System State on Completion

코드 결과창이 보여진다.

## F. 외부 문제 풀기

### F.1 Initial Assumption

사용자는 자신의 실력에 맞는 외부 문제은행의 문제를 추천받을 수 있으며, 진단고사를 푼 사용자만 추천받을 수 있다. 한번 추천받은 문제는 바뀌지 않고, 추천 문제 중 세 문제 이상 풀 경우 새로 추천받을 수 있다.

### F.2 Normal Flow of Events

사용자의 상태를 2가지로 나눌 수 있다.

#### 1) 진단고사를 푼 사용자

진단고사를 푼 사용자는 등급정보가 존재한다. 외부 문제은행 데이터베이스에서 사용자의 등급과 일치하는 등급의 문제들을 추천한다. 풀고 다시 페이지로 돌아와서 체크버튼을 클릭한 경우에만 체크가 된다. 추천 한 회당 5개의 문제를 추천하며, 3문제 이상 풀었을 경우 [새로 추천받기] 버튼이 활성화된다. [새로 추천받기]버튼을 누르면 다시 사용자 등급에 맞는 수준의 문제를 추천한다.

#### 2) 진단고사를 풀지 않은 사용자

외부 문제를 추천받고 풀 수 없으며 대시보드에는 '진단고사를 풀고 오면 문제를 추천받을 수 있습니다'라는 문구를 출력한다.

### F.3 What can go wrong

사용자가 추천받은 문제를 풀지 않았는데 체크박스에 체크할 경우, '문제를 풀고 오세요.' 라는 안내창을 띄운다.

사용자가 회원가입 시 입력한 외부 문제 은행 계정 아이디가 존재하지 않을 경우, '입력한 아이디가 존재하지 않습니다' 라는 안내창을 띄운다.

### F.4 System State on Completion

문제를 풀고 와서 체크버튼을 클릭할 경우, 푼 문제들은 채점 데이터베이스에 저장된다.

## G. 마이페이지

### G.1 Initial Assumption

사용자가 로그인을 한 상태여야 한다.

### G.2 Normal Flow of Events

마이페이지 버튼을 누르면 마이페이지로 이동한다.

비밀번호 변경 버튼을 누르면 계정 비밀번호를 수정할 수 있다. 수정 전 다시 현재 비밀번호를 다시 입력한다. 새로운 비밀번호를 입력 후, 새로운 비밀번호 확인 칸에 다시 같은 비밀번호를 입력한다. 변경 버튼을 눌러 비밀번호 변경 요청을 하면 사용자 데이터베이스에 변경된 비밀번호가 저장된다.

사용자는 푼 문제들을 확인할 수 있다. 문제 리스트를 누르면, 푼 문제와 시도 횟수를 표시한다. 푼 문제는 최신순으로 정렬되어 있다.

### G.3 What can go wrong

마이페이지로 이동하는 도중 로그인 상태가 해제되면 메인 페이지로 redirect하고 로그인을 다시 요구하는 에러 메시지를 띄운다.

비밀번호 형식이 4~12자의 문자열 형식이 아닐 경우, 다른 비밀번호를 입력하는 것을 요청한다.

#### G.4 System State on Completion

비밀번호가 변경되어 사용자 DB에 저장된다.

### H. 랭킹 확인

#### H.1 Initial Assumption

사용자는 자신과 다른 사람들의 랭킹을 확인할 수 있다. 개인랭킹은 진단고사를 푼 사용자만 확인할 수 있으며, 전체 랭킹은 진단고사에 상관없이 모든 사용자들이 확인할 수 있다.

#### H.2 Normal Flow of Events

사용자가 접근한 경우를 2가지로 나눌 수 있다.

##### 1) 진단고사를 푼 사용자

개인 랭킹은 로그인한 사용자와 같은 등급의 사용자들의 데이터베이스를 참조한다. 개인 랭킹은 같은 등급인 사용자들의 수와 그 중 로그인한 사용자의 등수와 출력한다. 등수는 푼 내부 문제를 기준으로 내림차순하여 매긴다.

##### 2) 진단고사를 풀지 않은 사용자

개인랭킹이 출력되는 위치에 '진단고사를 풀면 자신의 랭킹을 확인할 수 있습니다' 라는 문구를 출력한다.

전체 랭킹은 Qurious의 전체 사용자들의 데이터베이스를 참조한다. 전체 사용자의 푼 내 부문제 개수를 기준으로 내림차순하여 등수를 매긴다. 사용자는 전체 랭킹에서는 등수, 사용자의 닉네임, 푼 내부 문제 개수, 전체 내부분제 개수, 정답 비율을 확인할 수 있다. 사용자의 진단고사 결과유무와 상관없이 랭킹 정보를 출력한다.

### H.3 What can go wrong

문제 푼 기록이 없을 경우, 본인의 랭킹이 표시되지 않을 수 있다.

### H.4 System State on Completion

Curious의 전체 사용자 누구라도 내부 문제를 풀면 랭킹은 갱신된다.

## I. 내부 문제 관리

### I.1 Initial Assumption

관리자는 어드민 권한을 가지고 있는 사용자를 의미한다. 관리자로 로그인된 상태여야한다.

### I.2 Normal Flow of Events

관리자는 내부 문제를 추가하고 수정하고 삭제할 수 있는 권한이 있으며 이를 어드민 페이지에서 할 수 있다. 어드민 페이지에서 내부 문제 추가하기 페이지를 열면 문제에 대한 설명, 입력값에 대한 설명, 출력값에 대한 설명, 입력 예시, 출력 예시를 입력 해야하고 각종 제약사항(메모리, 실행시간)을 넣을 수 있다. 해당 문제가 어떤 난이도를 지니는지 입력해야한다. 테스트 케이스를 만들어 넣어야하며 이때 테스트 케이스는 zip파일 형태로 압축되어 있어야한다.

### I.3 What can go wrong

형식에 맞지 않는 Zip파일을 업로드한 경우 경고 메시지가 출력된다. 입력 폼 중에 빈공간이 있는 경우 해당 폼을 작성해야 문제를 추가할 수 있다. 문제 제목이 이미 있는 경우 문제를 추가할 수 없다.

### I.4 System State on Completion

내부 문제 DB에 해당 문제가 저장된다. 문제 관리창에서 해당 문제를 확인할 수 있다.

## J. 외부 문제 관리

### J.1 Initial Assumption

관리자는 어드민 권한을 가지고 있는 사용자를 의미한다. 관리자로 로그인된 상태여야한다.

### J.2 Normal Flow of Events

관리자는 어드민 페이지 내 외부 문제 추가하기를 열면 먼저 추가된 문제 들을 리스트 형식으로 볼 수 있다. 상단에 문제 추가하기 버튼을 누르면 문제를 추가하는 페이지로 이동한다. 페이지는 문제 제목, 문제 URL, 문제 유형, 문제의 난이도 등 문제에 대한 정보를 입력할 수 있는 입력란이 있다. 해당 입력란을 기입한 후 추가하기 버튼을 누르면 문제가 정상적으로 추가되고 추가된 문제는 문제 데이터 베이스에 추가되고 추천 서버에서 활용할 수 있다.

### J.3 What can go wrong

이미 있는 문제 번호를 입력한 경우 이미 있는 문제라는 경고 메시지가 뜬다. URL이 잘못된 형식인 경우 URL 오류 메시지가 뜬다.

### J.4 System State on Completion

외부문제 DB에 해당 문제가 저장된다. 문제 관리창에서 해당 문제를 확인할 수 있다.

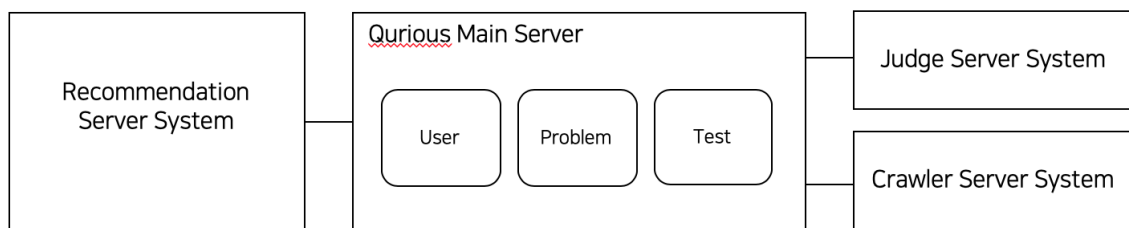
## 7 System Model

### 7.1 Objective

System Requirement Model에서는 Context Models, 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

#### A. Context Model



**Diagram 9 Context Model**

#### B. Process Model

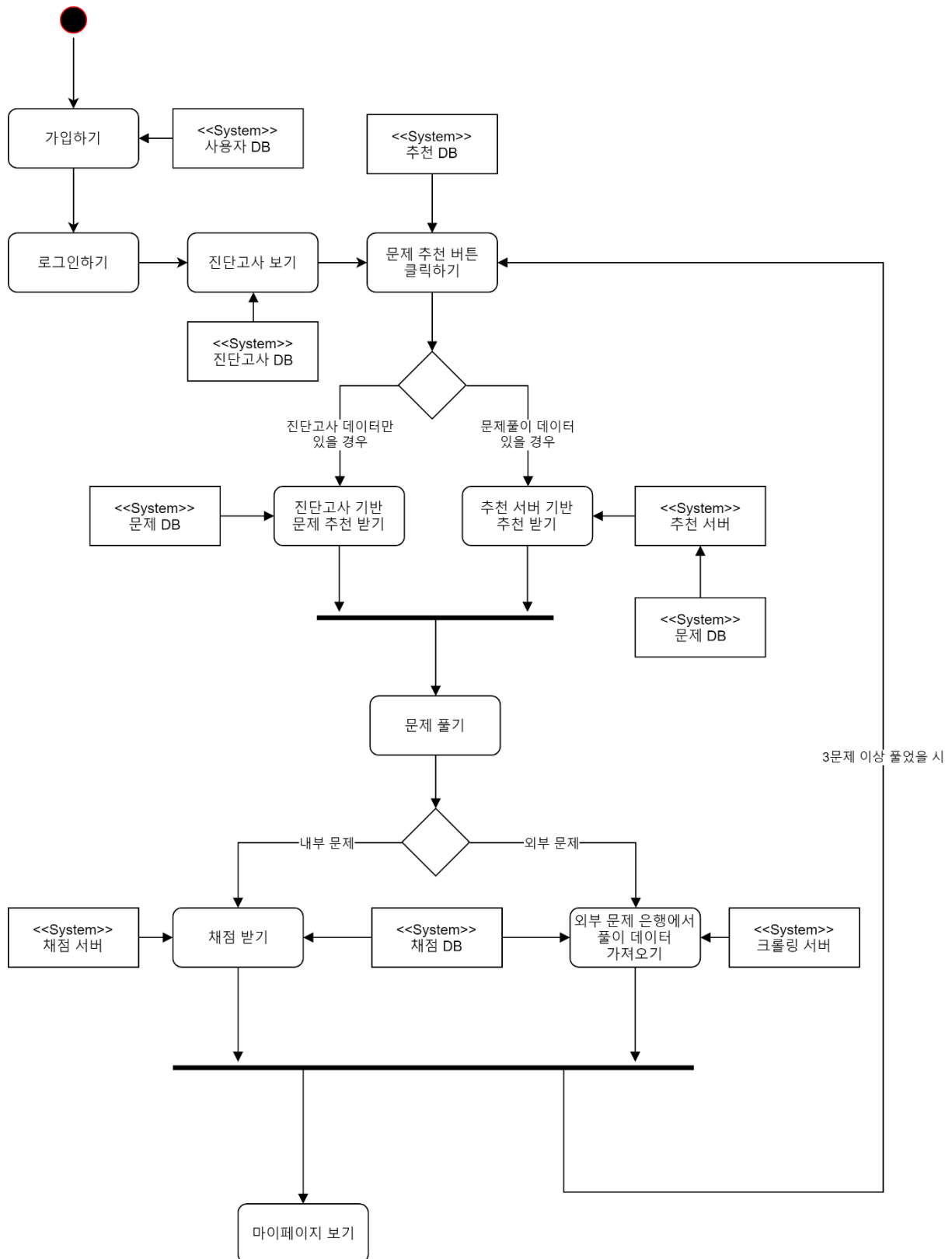


Diagram 10 Process Model

## 7.3 Interaction Models

### A. Use Case Model

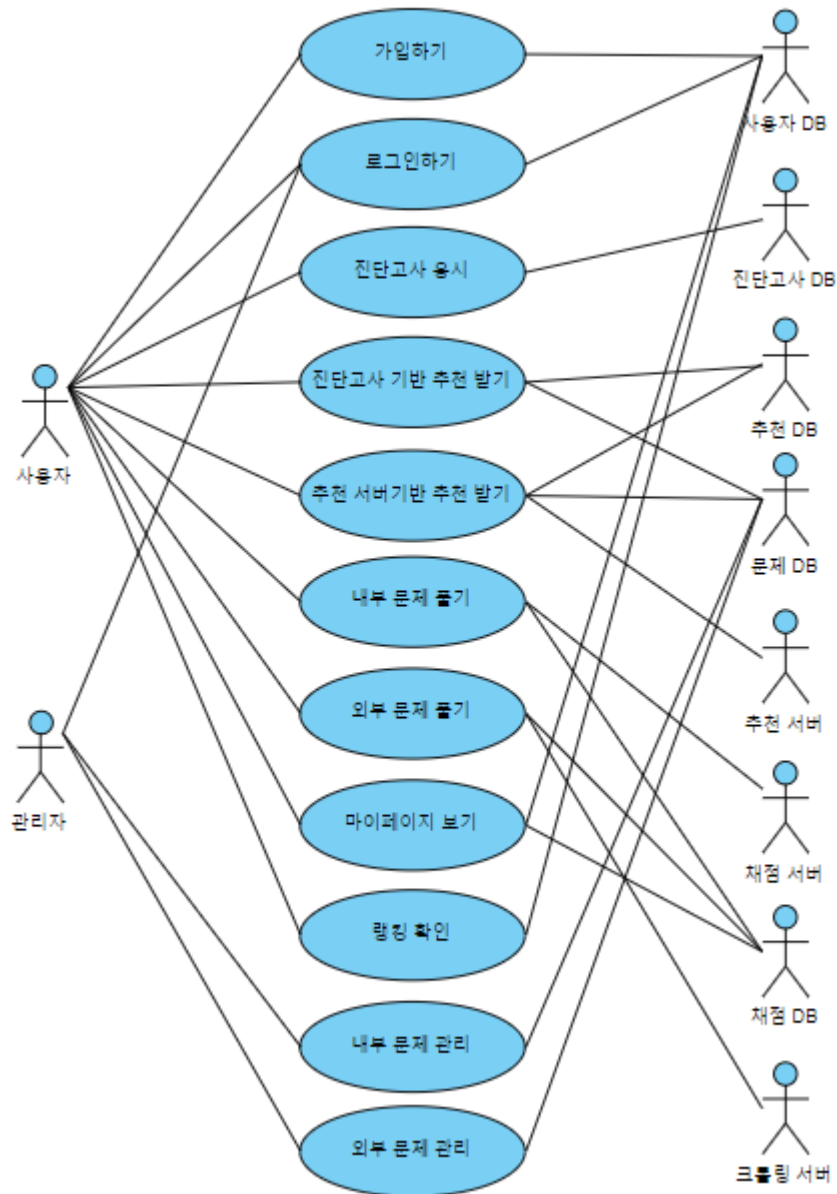


Diagram 11 Use Case Model



## B. Tabular Description

### B.1 가입하기

| USE CASE    | 가입하기   |
|-------------|--|
| ACTOR       | 사용자, 사용자 DB                                    |
| DESCRIPTION | 회원 가입을 위해 사용자 정보를 입력 받고, 데이터베이스에 사용자 정보를 저장한다. |
| STIMULUS    | 사용자가 사용자 정보를 입력하고 회원가입 버튼을 누른다.                |
| RESPONSE    | 사용자의 회원가입 정보를 검증한뒤 DB에 사용자 정보를 등록한다.           |
| COMMENTS    |  |

Table 16 Tabular Description: 가입하기

### B.2 로그인하기

| USE CASE    | 로그인하기   |
|-------------|---|
| ACTOR       | 사용자, 관리자, 사용자 DB                                |
| DESCRIPTION | 사용자가 서비스 이용을 위해 이메일과 비밀번호를 입력하고 사용 권한을 얻는다.     |
| STIMULUS    | 사용자가 이메일과 비밀번호를 입력하고 로그인 버튼을 누른다.               |
| RESPONSE    | 사용자의 로그인 정보와 DB에 저장된 사용자 정보를 대조하여 로그인 결과를 반환한다. |
| COMMENTS    |   |

Table 17 Tabular Description: 로그인하기

### B.3 진단고사 응시

| USE CASE    | 진단고사 응시  |
|-------------|--|
| ACTOR       | 사용자, 진단고사 DB   |
| DESCRIPTION | 사용자DB는 비어있기 때문에, 사용자에게 문제를 추천할 때 사용할 실력 수준을 알아내기 위해 진단고사를 응시한다. 사용자는 문제마다 응답을 제시하고, 본 서비스는 응답을 모아 채점서버로 전송한다. 채점서버에서 받은 실력수준 데이터를 진단고사 기반 추천으로 전달한다. |

|                 |   |
|-----------------|---|
| <b>STIMULUS</b> | 사용자가 로그인을 한 후, 진단고사 응시 버튼을 누르고 난이도를 선택한 후 문제를 푼다. |
| <b>RESPONSE</b> | 사용자 데이터베이스에 사용자의 실력수준이 추가된다.                      |
| <b>COMMENTS</b> |   |

**Table 18 Tabular Description: 진단고사 응시**

#### B.4 진단고사 기반 추천 받기

|                    |                                      |
|--------------------|--------------------------------------|
| <b>USE CASE</b>    | <b>진단고사 기반 추천 받기</b>                 |
| <b>ACTOR</b>       | 사용자, 문제 DB                           |
| <b>DESCRIPTION</b> | 진단고사 데이터만 보유한 사용자가 문제를 추천 받는 기능이다.   |
| <b>STIMULUS</b>    | 진단고사를 보고 나서, '문제 추천 받기' 버튼을 클릭하면 된다. |
| <b>RESPONSE</b>    | 문제 DB에서 사용자의 실력 수준에 맞는 문제를 5개 추천해준다. |
| <b>COMMENTS</b>    |                                      |

**Table 19 Tabular Description: 진단고사 기반 추천 받기**

#### B.5 추천 서버 기반 추천 받기

|                    |   |
|--------------------|---|
| <b>USE CASE</b>    | <b>추천 서버 기반 추천 받기</b>   |
| <b>ACTOR</b>       | 사용자, 추천 서버, 문제 DB   |
| <b>DESCRIPTION</b> | 진단고사 데이터와 문제풀이 데이터를 둘 다 보유한 사용자가 문제를 추천 받는 기능이다.                      |
| <b>STIMULUS</b>    | 문제를 3문제 이상 풀고나서, '문제 추천 받기' 버튼을 클릭하면 된다.                              |
| <b>RESPONSE</b>    | 사용자 실력 수준 데이터와, 1일 마다 갱신되는 협업 필터링 모델을 통해 사용자에게 가장 적합한 문제를 5개 추천해준다.   |
| <b>COMMENTS</b>    | 사용자가 추천 받은 문제를 모두 풀었을 때까지, '문제 추천 받기' 버튼을 누르지 않는다면 자동으로 문제를 5개 추천해준다. |

**Table 20 Tabular Description 추천 서버 기반 추천 받기**

#### B.6 내부 문제 풀기

|                 |                 |
|-----------------|-----------------|
| <b>USE CASE</b> | <b>내부 문제 풀기</b> |
| <b>ACTOR</b>    | 사용자             |

|                    |   |
|--------------------|---|
| <b>DESCRIPTION</b> | 사용자는 추천 받은 문제들 중에서 자체 문제 태그로 된다. 이 문제를 클릭하면 문제 풀기 페이지로 이동한다. 해당 페이지에선 문제에 대한 설명과 입력값, 출력값이 적혀있고 사용자가 직접 코드를 작성할 수 있는 코드 에디터가 존재한다. 지원하는 언어는 Python 이고 사용자가 코드를 입력하면 예약어에 대해선 다른 색깔로 하이라이트 표시가 된다. |
| <b>STIMULUS</b>    | 해당 문제 클릭하고 Python3 코드를 입력   |
| <b>RESPONSE</b>    | 성공, 실패,에러(컴파일 에러 / 런타임 에러)  |
| <b>COMMENTS</b>    |   |

Table 21 Tabular Description: 내부 문제 풀기

## B.7 외부 문제 풀기

| USE CASE           | 외부 문제 풀기   |
|--------------------|--|
| <b>ACTOR</b>       | 사용자, 크롤링 서버, 채점 DB   |
| <b>DESCRIPTION</b> | 사용자가 외부 문제 은행의 문제를 추천 받는 기능이다.   |
| <b>STIMULUS</b>    | 사용자가 진단고사를 풀고 외부 문제 추천 받기 버튼을 누른다. 추천된 문제로 이동하는 링크가 대시보드에 출력되고 링크를 클릭하면 해당 문제로 이동한다. |
| <b>RESPONSE</b>    | 진단고사 결과가 없을 경우, 진단고사를 풀고 오라는 안내창을 띄운다.<br>외부 문제 DB에서 사용자의 실력수준과 일치하는 문제를 가져온다.       |
| <b>COMMENTS</b>    |  |

Table 22 Tabular Description: 외부 문제 풀기

## B.8 마이페이지 보기

| USE CASE           | 마이페이지 보기   |
|--------------------|--|
| <b>ACTOR</b>       | 사용자, 사용자 DB, 채점 DB   |
| <b>DESCRIPTION</b> | 로그인 후 마이페이지 버튼을 눌러 들어간다. 비밀번호 변경 버튼을 누르면 새로운 비밀번호를 입력하고 변경할 수 있다. 푼 문제 리스트들을 볼 수 있다. |
| <b>STIMULUS</b>    | 사용자가 로그인을 한 후 마이페이지 버튼을 눌러 들어간다.   |

|                 |                |
|-----------------|----------------|
| <b>RESPONSE</b> | 마이페이지 화면이 보인다. |
| <b>COMMENTS</b> |                |

**Table 23 Tabular Description: 마이페이지 보기**

#### B.9 랭킹 확인

|                    |  |
|--------------------|--|
| <b>USE CASE</b>    | <b>랭킹 확인</b>   |
| <b>ACTOR</b>       | 사용자, 채점 DB   |
| <b>DESCRIPTION</b> | 사용자가 자신의 랭킹과 Qurious의 전체 사용자의 랭킹을 확인하는 기능이다.   |
| <b>STIMULUS</b>    | 사용자가 랭킹확인 페이지를 들어간다.   |
| <b>RESPONSE</b>    | 전체 사용자의 채점 DB에서 푼 문제 수를 확인한다.<br>진단고사 결과가 없을 경우 진단고사를 풀고 오면 개인랭킹을 확인할 수 있다는 문구를 띄운다. |
| <b>COMMENTS</b>    |  |

**Table 24 Tabular Description: 랭킹 확인**

#### B.10 내부 문제 관리

|                    |   |
|--------------------|---|
| <b>USE CASE</b>    | <b>내부 문제 관리</b>   |
| <b>ACTOR</b>       | 관리자   |
| <b>DESCRIPTION</b> | 관리자는 내부 문제를 추가하고 수정하고 삭제할 수 있는 권한이 있으며 이를 어드민 페이지에서 할 수 있다. |
| <b>STIMULUS</b>    | 내부 문제 추가하기 버튼 클릭  |
| <b>RESPONSE</b>    | 성공, 에러 메시지  |
| <b>COMMENTS</b>    |   |

**Table 25 Tabular Description: 내부 문제 관리**

#### B.11 외부 문제 관리

|                 |                 |
|-----------------|-----------------|
| <b>USE CASE</b> | <b>외부 문제 관리</b> |
| <b>ACTOR</b>    | 관리자             |

|                    |  |
|--------------------|--|
| <b>DESCRIPTION</b> | 관리자는 어드민 페이지 내 외부 문제 추가하기를 열면 먼저 추가된 문제 들을 리스트 형식으로 볼 수 있다. 상단에 문제 추가하기 버튼을 누르면 문제를 추가하는 페이지로 이동한다. 페이지는 문제 제목, 문제 URL, 문제 유형, 문제의 난이도 등 문제에 대한 정보를 입력할 수 있는 입력란이 있다. 해당 입력란을 기입한 후 추가하기 버튼을 누르면 문제가 정상적으로 추가된다. |
| <b>STIMULUS</b>    | 외부 문제 추가하기버튼 클릭  |
| <b>RESPONSE</b>    | 성공, 에러 메시지   |
| <b>COMMENTS</b>    |  |

Table 26 Tabular Description: 외부 문제 관리

## C. Sequence Diagram

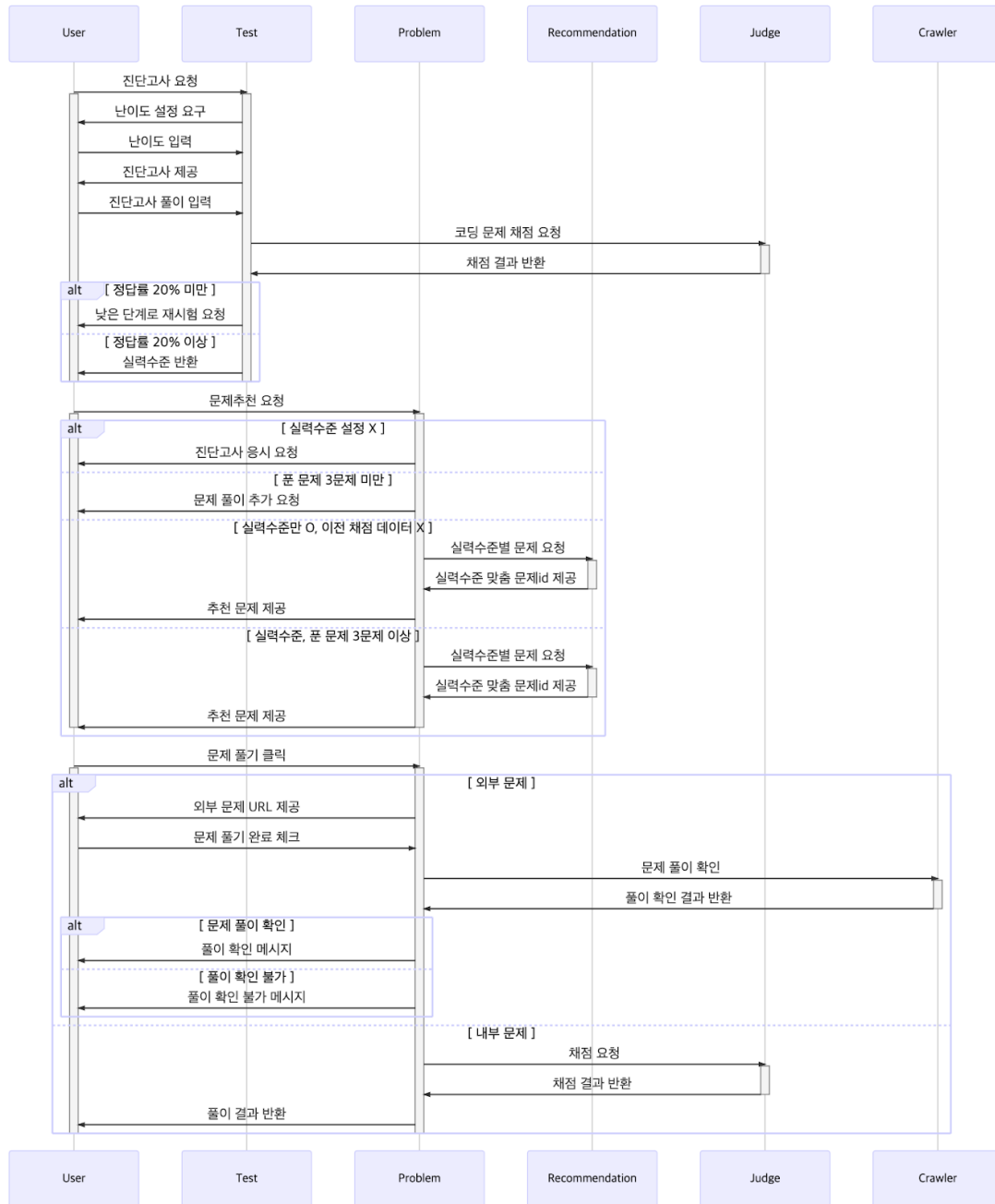


Diagram 12 Sequence Diagram: 전체

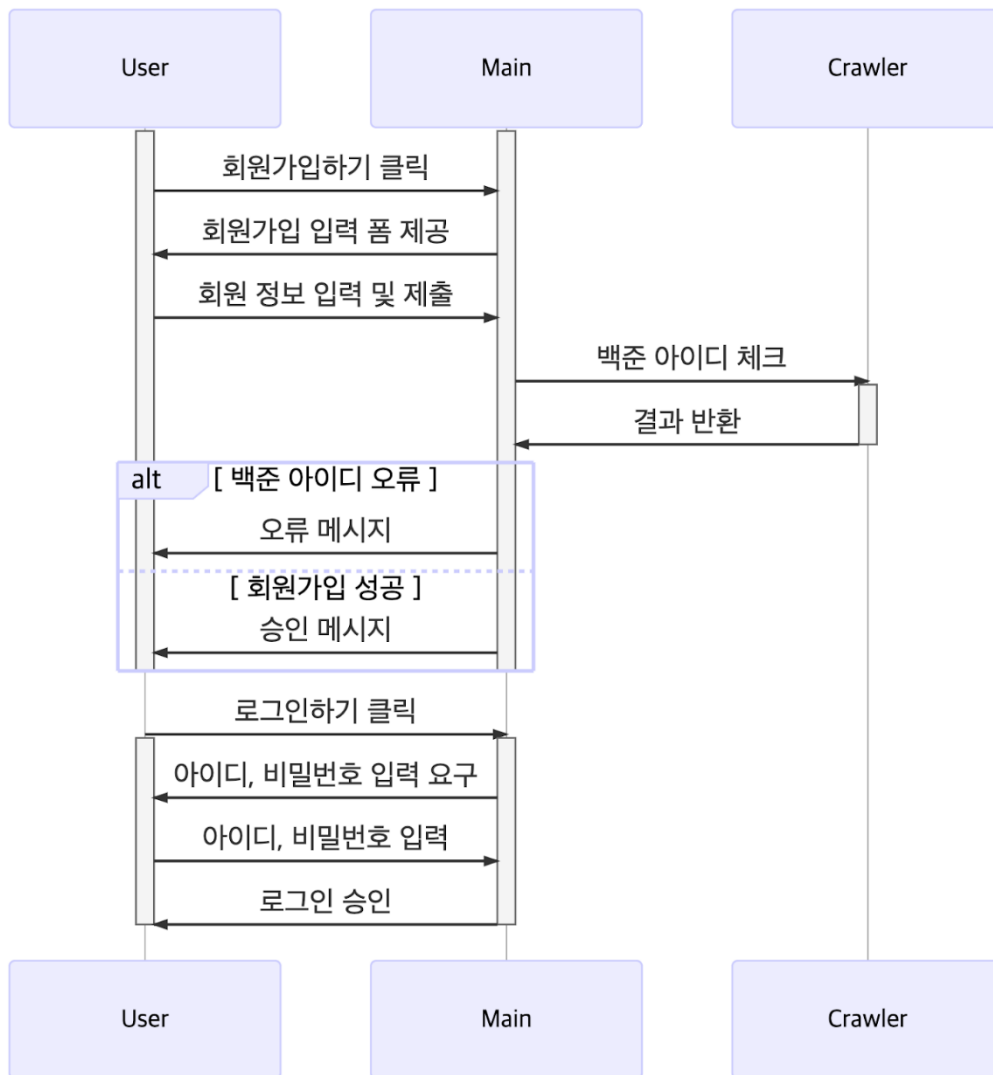


Diagram 13 Sequence Diagram: 회원가입

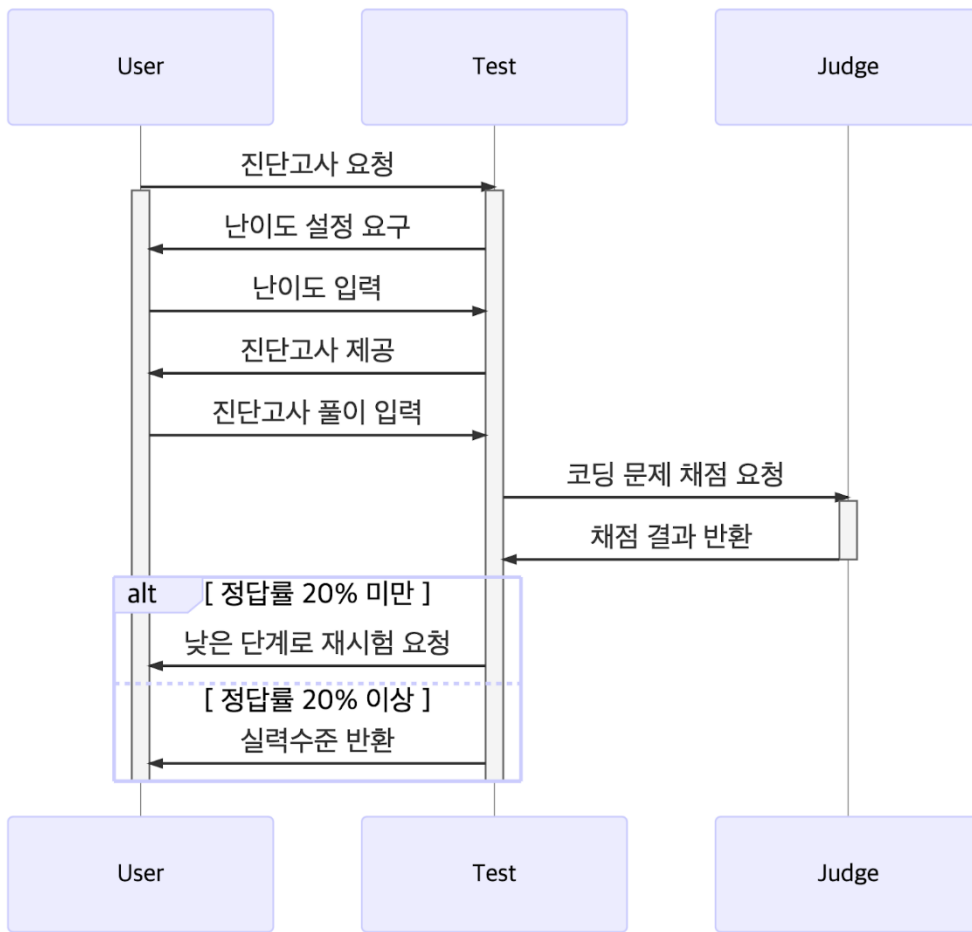


Diagram 14 Sequence Diagram: 채점



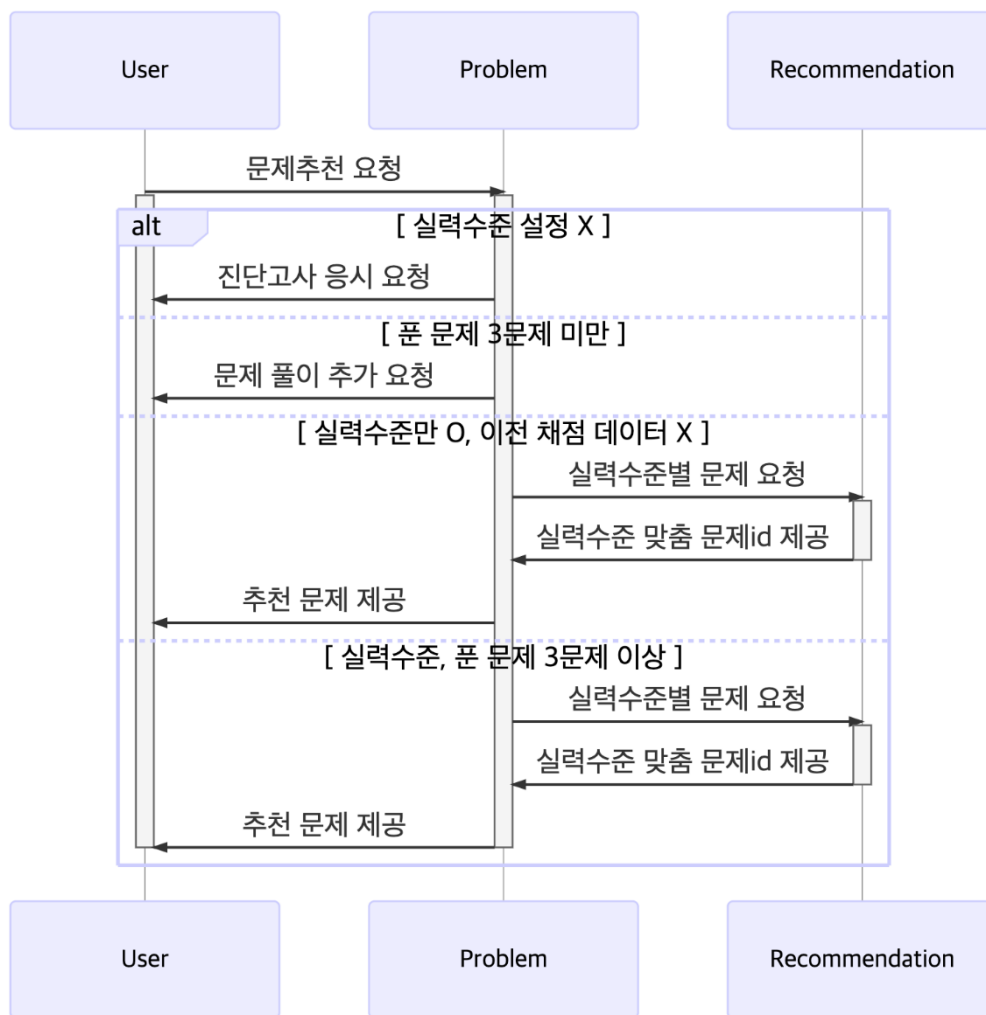


Diagram 15 Sequence Diagram: 추천

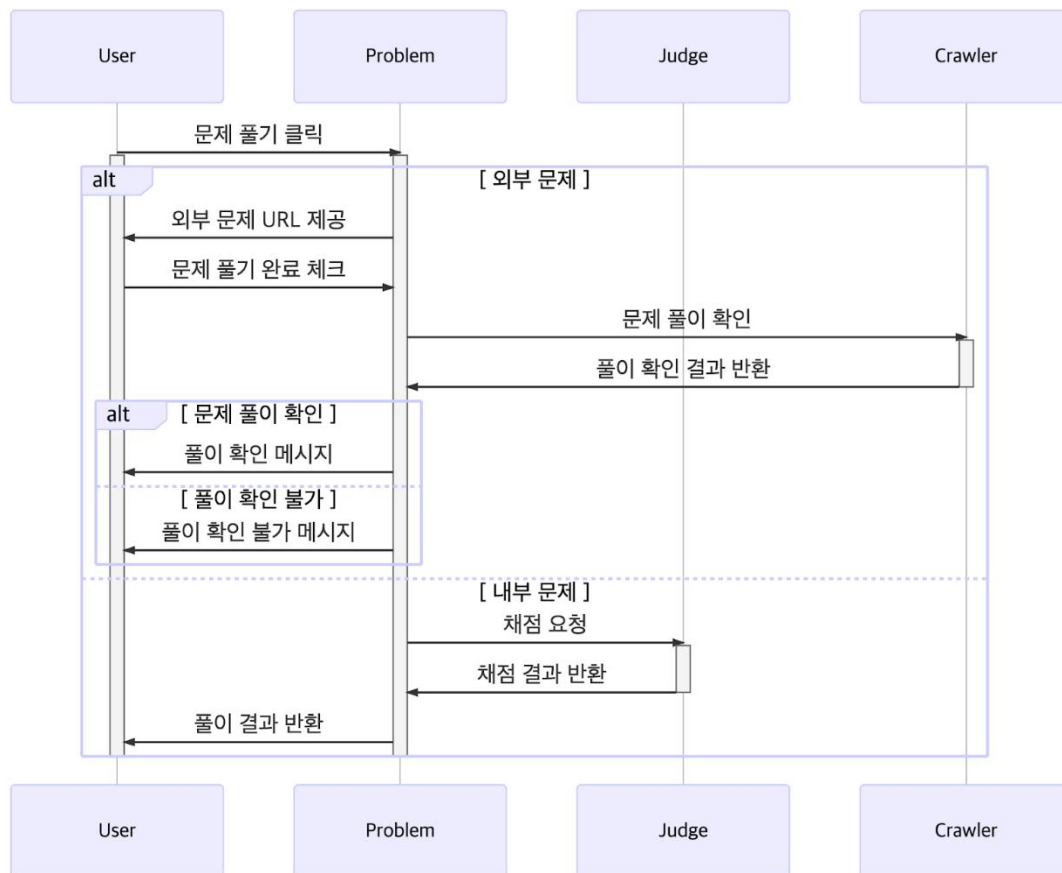


Diagram 16 Sequence Diagram: 외부 문제 풀이

## 7.4 Structural Models

### A. Class Diagram

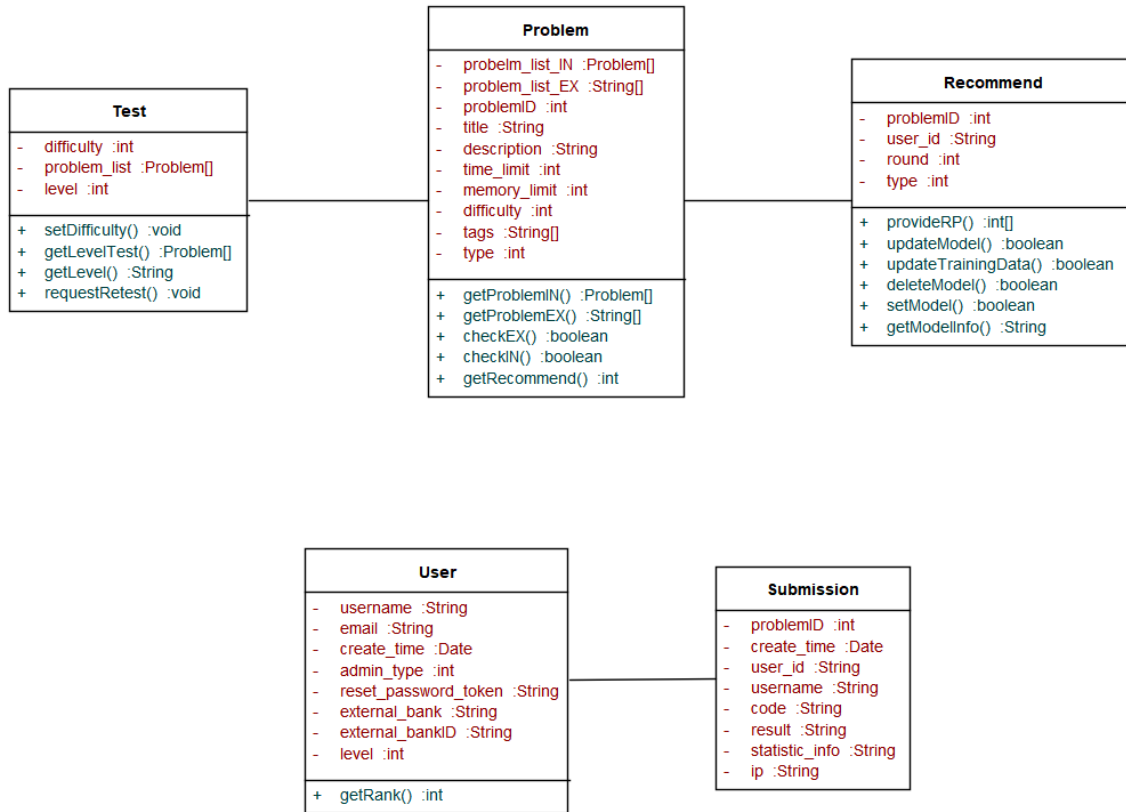


Diagram 17 Class Diagram

## 7.5 Behavioral Models

### A. Event-driven Diagram

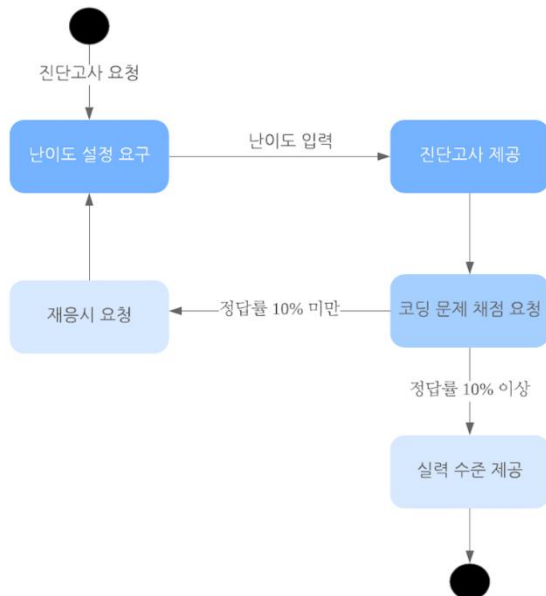


Diagram 18 Event Driven Diagram: 진단고사

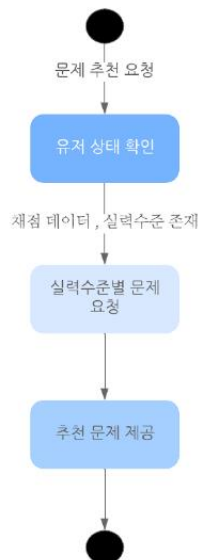


Diagram 19 Event Driven Diagram: 문제 추천

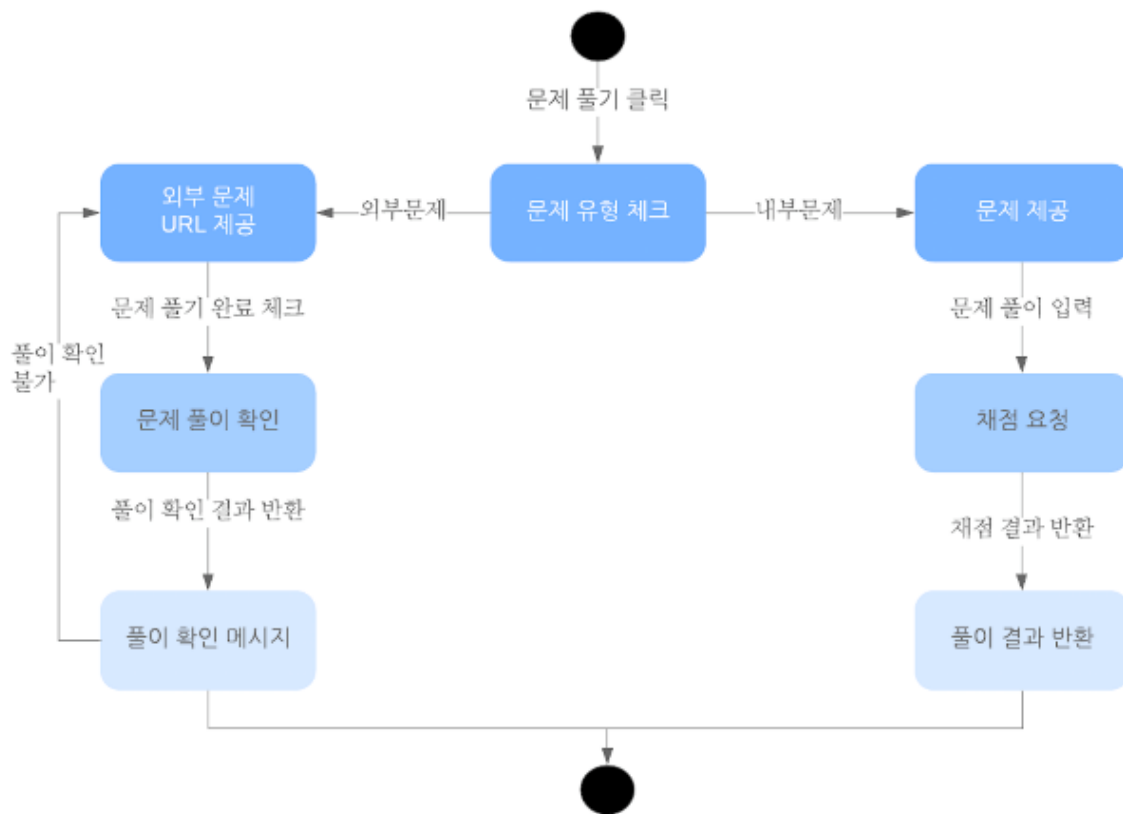


Diagram 20 Event Driven Diagram: 문제 풀이

## 8 System Requirement Evolution

### 8.1 Objective

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

### 8.2 Limitation

#### A. 외부 문제 의존성

Quirious는 자체 제작한 내부 문제와 더불어 외부 알고리즘 사이트의 문제를 함께 제공한다. 이에 따라 발생하는 몇가지 한계점이 존재한다.

첫째로, 자체 제작하는 문제 수가 타 플랫폼에 비해 적다. 충분한 콘텐츠를 제공하지 못해 추천 알고리즘의 성능에도 문제를 줄 수 있고, 이에 따라 사용자에게 신뢰를 주지 못

하는 상황이 발생할 수 있다.

외부 문제를 가져올 때, 검증에 대한 요구가 발생하게 된다. 자체 제작하는 문제가 아니기 때문에 문제의 질을 보장할 수 없고, 해당 문제가 어떤 난이도에 속하는지도 확인해야 한다.

외부 플랫폼으로의 이탈 가능성도 존재한다. 외부 문제를 제공하고, 사용자가 Quirious 플랫폼이 아닌 타 플랫폼에 만족을 느껴 이탈할 수 있다. 진단 및 추천 시스템을 원하지 않고 문제 풀이 서비스만을 원하는 사용자는 상대적으로 다양한 문제 pool을 확보하고 있는 외부 서비스로 이탈할 가능성이 높다.

#### B. 추천 정확도 문제

사용자가 초기에 서비스를 이용하고자 할 경우, 사용자의 문제 풀이 데이터가 없어 진단고사 외에는 추천할 수 있는 데이터 수 자체가 적다. 이에 따라 cold start 문제가 발생한다. 또한 알고리즘 문제의 특성상, 하나의 문제에 대한 여러가지 해답이 존재한다. 따라서 한 문제에 대한 feature를 모두 특징짓기 어렵다. 만약 문제 수의 부족으로 같은 문제가 반복하여 추천되거나, 추천 문제가 사용자의 기대에 부응하지 못할 경우 사용자의 이탈로 이어질 수 있다.

### 8.3 System Evolution

#### A. Evolution for AI tutor

인공지능 기술의 발전으로 단순히 문제 추천을 해주는 것 뿐만 아니라, 학습 과정 전반에 걸쳐 tutor의 역할을 해주는 서비스가 늘어나고있다. 진단 테스트부터 문제 추천, 학습 스타일 가이드, 강의 추천까지 AI를 이용한 솔루션을 제공한다. Quirious의 경우 코드를 line by line으로 검사하여 코딩 스타일을 가이드하며, 각 언어 특성에 맞는 구조와 syntax를 사용할 수 있도록 지도해야 한다. 거기서 그치지 않고, 학습할 수 있는 자료와 강의까지 추천해주어 AI tutor의 역할을 할 수 있어야 한다.

#### B. Evolution for Mobile

웹 서비스를 모바일 기기로 이용하는 사용자가 pc보다 더 많아졌다. 하지만 코딩의 특성상 모바일 환경에서는 키보드를 사용할 수 없기 때문에, 접근성이 떨어질 수 있다. 모바일

일에서 코딩이 가능하도록 하는 방안으로 모바일 가상 키보드를 제공한다. 각 키보드에는 프로그래밍 언어별 예약어를 블록으로 제시하여 키보드 입력을 최소화한다. 또한 따로 어플리케이션을 설치하지 않아도 웹에서 동작하는 IDE를 제공해 모바일에서도 코딩이 가능하도록 한다.

### C. Evolution for User Requirement

사용자의 학습 목적은 단순히 프로그래밍 역량 강화에서 그치지 않고, 알고리즘 대회 참가, 컴퓨팅 사고 능력 증진 등 다양한 목적이 있을 수 있다. 사용자에게 목적에 맞는 문제 pool을 제공하고, 적절한 시스템을 제공해야 한다. 또한 사용자의 유형이 학습자뿐만 아니라 교수자일수도 있다. 교수자에게는 학생들의 수준을 파악하기 위해, 클래스를 구성해 문제를 출제하고 학생들의 풀이 내역을 확인할 수 있는 기능을 제공한다. 또한 문제들을 pdf로 변환해 수업 자료로 사용할 수 있도록 해야한다.

## 9 Appendix

### 9.1 Objectives

Appendix에서는 개발되는 시스템에서 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, **Relationship type**은 굵은 이탤릭체, Constraint는 밑줄로 표시한다.

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

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

**제출 데이터베이스**는 id를 필수적으로 가져야 하고, 추가적으로 *problem, user\_id, username, code, result*를 attribute로 가질 수 있다.

**문제 데이터베이스**는 id를 필수적으로 가져야 하고, *title, description, input, output, test\_case\_id, difficulty*를 attribute로 가질 수 있다.

**추천 데이터베이스**는 problem\_id, user\_id, round, problem\_type를 attribute로서 가져야 한다.

### 9.3 Database Detail

**사용자 데이터베이스** Qurious에 사용자가 가입할 때 입력되며, **제출 데이터베이스**, **추천 데이터베이스**에 참조된다. **제출 데이터베이스**는 사용자가 내부 문제를 제출하면 제출 결과를 저장한다. 저장된 제출 결과에 따라 **사용자 데이터베이스** *level* 속성이 변경될 수 있다. **추천 데이터베이스**는 **사용자 데이터베이스**와 **문제 데이터베이스**를 참조하여 사용자가 제출한 문제들의 데이터를 저장한다.

## 10 Index

### 10.1 Objectives

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

### 10.2 Table Index

|  |    |
|--|----|
| Table 1 용어 정의 .....                                  | 11 |
| Table 2 약어.....                                      | 11 |
| Table 3 Functional Requirement: 회원가입.....            | 25 |
| Table 4 Functional Requirement: 로그인 .....            | 26 |
| Table 5 Functional Requirement: 진단고사 응시 .....        | 26 |
| Table 6 Functional Requirement: 진단고사 채점 .....        | 27 |
| Table 7 Functional Requirement: 문제 추천 받기.....        | 28 |
| Table 8 Functional Requirement: 추천 서버에서 문제 추천하기..... | 28 |
| Table 9 Functional Requirement: 진단고사 기반 추천.....      | 29 |



|   |    |
|---|----|
| Table 10 Functional Requirement: 내부 문제 풀기 .....   | 30 |
| Table 11 Functional Requirement: 외부 문제 풀기 .....   | 31 |
| Table 12 Functional Requirement: 마이페이지 .....      | 31 |
| Table 13 Functional Requirement: 랭킹 확인 .....      | 32 |
| Table 14 Functional Requirement: 내부 문제 관리 .....   | 32 |
| Table 15 Functional Requirement: 외부 문제 관리 .....   | 33 |
| Table 16 Tabular Description: 가입하기 .....          | 48 |
| Table 17 Tabular Description: 로그인하기 .....         | 48 |
| Table 18 Tabular Description: 진단고사 응시 .....       | 49 |
| Table 19 Tabular Description: 진단고사 기반 추천 받기 ..... | 49 |
| Table 20 Tabular Description 추천 서버 기반 추천 받기 ..... | 49 |
| Table 21 Tabular Description: 내부 문제 풀기 .....      | 50 |
| Table 22 Tabular Description: 외부 문제 풀기 .....      | 50 |
| Table 23 Tabular Description: 마이페이지 보기 .....      | 51 |
| Table 24 Tabular Description: 랭킹 확인 .....         | 51 |
| Table 25 Tabular Description: 내부 문제 관리 .....      | 51 |
| Table 26 Tabular Description: 외부 문제 관리 .....      | 52 |

### 10.3 Figure Index

|                                     |   |
|-------------------------------------|---|
| Figure 1 SW중심사회를 위한 인재양성 추진계획 ..... | 8 |
| Figure 2 주요 개편 방향 .....             | 8 |
| Figure 3 사교육 현황 .....               | 8 |

#### 10.4 Diagram Index

|   |    |
|---|----|
| Diagram 1 System Architecture: Curious 전체 .....     | 17 |
| Diagram 2 System Architecture: 웹(메인)서버 시스템 상세 ..... | 18 |
| Diagram 3 System Architecture: 사용자 시스템 .....        | 19 |
| Diagram 4 System Architecture: 랭킹 시스템 .....         | 20 |
| Diagram 5 System Architecture: 진단고사 시스템 .....       | 21 |
| Diagram 6 System Architecture: 문제 풀이 시스템 .....      | 22 |
| Diagram 7 System Architecture: 문제 추천 시스템 .....      | 23 |
| Diagram 8 System Architecture: 문제 관리 시스템 .....      | 24 |
| Diagram 9 Context Model.....                        | 45 |
| Diagram 10 Process Model.....                       | 46 |
| Diagram 11 Use Case Model .....                     | 47 |
| Diagram 12 Sequence Diagram: 전체 .....               | 53 |
| Diagram 13 Sequence Diagram: 회원가입 .....             | 54 |
| Diagram 14 Sequence Diagram: 채점 .....               | 55 |
| Diagram 15 Sequence Diagram: 추천 .....               | 56 |
| Diagram 16 Sequence Diagram: 외부 문제 풀이 .....         | 57 |
| Diagram 17 Class Diagram .....                      | 58 |
| Diagram 18 Event Driven Diagram: 진단고사 .....         | 59 |
| Diagram 19 Event Driven Diagram: 문제 추천 .....        | 59 |
| Diagram 20 Event Driven Diagram: 문제 풀이 .....        | 60 |

## 11 Reference

김민석, *SW인재 양성을 위한 국내외 SW교육 현황 및 시사점*, (정보통신산업진흥원: SW 산업기획팀 소프트웨어산업진흥본부, 2018.12.31), 8, 9.

남지원, "월 60만원짜리 '코딩학원'...코딩교육 열풍에 학원·교습소 급증", *경향신문*, 2017.12.19, [http://news.khan.co.kr/kh\\_news/khan\\_art\\_view.html?art\\_id=201712121522001](http://news.khan.co.kr/kh_news/khan_art_view.html?art_id=201712121522001)

최종경, 코딩 이슈리포트, *BNK투자증권*, 2017.12.19