Software Requirements Specification

**for**

**StoryTelling-based**

**Personalized Learning Platform**



**2018314952 박성민**

**2018312672 김민석**

**2017310277 서보현**

**2017314533 유현택**

**2018314980 이재필**

**2018312863 조성원**

**2019313156 채서영**

**소프트웨어공학개론 6조**

**2023.04.30**

**Table of Contents**

[**1. INTRODUCTION 6**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.gjdgxs)

[**1.1. PURPOSE 6**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.30j0zll)

[**1.2. SCOPE 6**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.1fob9te)

[**1.3. Definition, Acronyms, and Abbreviation 7**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.3znysh7)

[**1.4. References 8**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.1t3h5sf)

[**1.5. Overview 8**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.4d34og8)

[**2. Overall Description 9**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2s8eyo1)

[**2.1. Product perspective 9**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.17dp8vu)

[**2.2. Product Functions 9**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.3rdcrjn)

[2.2.1. MAIN PAGE 11](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.lnxbz9)

[2.2.2. RUN RESULT PAGE 11](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.35nkun2)

[2.2.3. GRADING RESULT PAGE 11](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.1ksv4uv)

[2.2.4. SUBMIT & RESULT ANALYSIS PAGE 11](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.44sinio)

[2.2.5. SUBMIT & RESULT ANALYSIS PAGE 11](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.44sinio)

[2.2.6. SUBMIT & RESULT ANALYSIS PAGE 12](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.44sinio)

[2.2.7. SUBMIT & RESULT ANALYSIS PAGE 12](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.44sinio)

[2.2.8. SUBMIT & RESULT ANALYSIS PAGE 12](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.44sinio)

[2.2.9. SUBMIT & RESULT ANALYSIS PAGE 12](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.44sinio)

[2.2.10. SUBMIT & RESULT ANALYSIS PAGE 12](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.44sinio)

[**2.3. USER CLASSES AND CHARACTERISTICS 12**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2jxsxqh)

[2.3.1. SYSTEM ADMINISTRATOR 12](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.z337ya)

[2.3.2. STUDENT 13](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.3j2qqm3)

[2.3.3. LECTURER & TA 13](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.1y810tw)

[**2.4. OPERATING ENVIRONMENT 13**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.4i7ojhp)

[**2.5. Design and Implementation Constraints 13**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.4i7ojhp)

[**2.6. Assumptions and Dependencies 14**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2xcytpi)

[**3. EXTERNAL INTERFACE REQUIREMENTS 14**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.1ci93xb)

[**3.1. USER INTERFACE 14**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.3whwml4)

[**3.2. HARDWARE INTERFACE 24**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.ihv636)

[**3.3. software interfaces 25**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.41mghml)

[**3.4. Communication Interfaces 26**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.vx1227)

[**4. SYSTEM REQUIREMENTS 27**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.1v1yuxt)

[**4.1. FUNCTIONAL REQUIREMENTS 27**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.4f1mdlm)

[4.1.1. USE CASE 27](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2u6wntf)

[4.1.2. USE CASE DIAGRAM 36](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2zbgiuw)

[4.1.3. DATA FLOW DIAGRAM](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2zbgiuw) 38

[**4.2. NON FUNCTIONAL REQUIREMENTS 39**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2dlolyb)

[4.2.1. PRODUCT REQUIREMENT 39](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.sqyw64)

[4.2.2. ORGANIZATIONAL REQUIREMENT 39](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.3cqmetx)

[4.2.3. EXTERNAL REQUIREMENT 40](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.1rvwp1q)

[**4.3. ORGANIZING SYSTEM FLOW 40**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.4bvk7pj)

[4.3.1. CONTEXT MODEL 40](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2r0uhxc)

[4.3.2. PROCESS MODEL 41](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.3q5sasy)

[4.3.3. INTERACTION MODEL 41](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.kgcv8k)

[4.3.4. BEHAVIOR MODEL 41](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.34g0dwd)

[**4.4. SYSTEM ARCHITECTURE 42**](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.43ky6rz)

**그림 목차**

[Figure 1. Main Page Diagram 16](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.qsh70q)

[Figure 2. Use Case Diagram - System 36](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.49x2ik5)

[Figure 3. Use Case Diagram - Survey 36](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.147n2zr)

[Figure 4. Use Case Diagram - Learning 37](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.qsh70q)

[Figure 5. Use Case Diagram – Question Solving 37](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.49x2ik5)

[Figure 6. Data Flow Diagram – Question Solving 38](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.147n2zr)

[Figure 7. Data Flow Diagram – Learning 38](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.qsh70q)

[Figure 8. Data Flow Diagram - System 38](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.49x2ik5)

[Figure 9. Data Flow Diagram – Survey 38](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.147n2zr)

[Figure 10. Context Model 40](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.qsh70q)

[Figure 11. Process Model 41](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.49x2ik5)

[Figure 12. Sequence Model 42](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.147n2zr)

[Figure 13. System Architecture 42](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.qsh70q)

**표 목차**

[Table 1. Acronyms/Abbreviation 7](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2et92p0)

[Table 2. Terms and definitions 7](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.3dy6vkm)

[Table 3. Table of Product Functions 9](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.26in1rg)

[Table 4. User Interface of Main Page 14](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2bn6wsx)

[Table 5. User Interface of Main Page - Login 15](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2et92p0)

[Table 6. User Interface of Main Page - Register 16](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.3dy6vkm)

[Table 7. User Interface of Student Level Test Page](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.26in1rg) 17

[Table 8. User Interface of Student Code Test Page 1](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2bn6wsx)7

[Table 9. User Interface of Student Curriculum Page 18](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2et92p0)

[Table 10. User Interface of Teacher Mode Page 20](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.3dy6vkm)

[Table 11. User Interface of Teller Select Page 21](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.26in1rg)

[Table 12. User Interface of Storytelling Page 22](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2bn6wsx)

[Table 13. Hardware Interface for User 24](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2et92p0)

[Table 14. Software Interface of DB 25](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.3dy6vkm)

[Table 15. Software Interface of Front-end 25](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.26in1rg)

[Table 16. Software Interface of Back-end 25](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2bn6wsx)

[Table 17. Communication Interfaces for Web Platform 26](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2et92p0)

[Table 18. Functional requirements abstraction 27](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.3dy6vkm)

[Table 19. Use Case of Curriculum 28](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.26in1rg)

[Table 20. Use Case of Progress 29](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2bn6wsx)

[Table 21. Use Case of Storytelling 29](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2et92p0)

[Table 22. Use Case of Question Generation 30](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.3dy6vkm)

[Table 23. Use Case of Code Edit & Run 30](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.26in1rg)

[Table 24. Use Case of Code Submt 31](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2bn6wsx)

[Table 25. Use Case of Code Grading 31](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2et92p0)

[Table 26. Use Case of Code Review 32](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.3dy6vkm)

[Table 27. Use Case of Level Test 32](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.26in1rg)

[Table 28. Use Case of Teller Select 3](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2bn6wsx)3

[Table 29. Use Case of Language Select 34](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2et92p0)

[Table 30. Use Case of Login 34](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.3dy6vkm)

[Table 31. Use Case of Register 35](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.26in1rg)

[Table 32. Use Case of Mode Select 35](https://docs.google.com/document/d/1NEwvSEo45oAHdh4MSEY3amr9iNW96ouy/edit#heading=h.2bn6wsx)

**1. INTRODUCTION**

* 1. **. PURPOSE**

본 문서는 “StoryTelling-based Personalized Learning Platform” 서비스를 위한 소프트웨어 요구명세서이다. 2023학년도 1학기 성균관대학교 소프트웨어공학개론 수업의 수강생으로 이루어진 Team6이 해당 프로젝트를 설계 및 구현하는 것으로 본 문서는 해당 서비스의 개발 및 구현에 필요한 요구사항을 분석하고 정리하는데 의의를 둔다. Team6는 본 문서에 기재된 내용에 기반하여 본 시스템을 개발한다.

본 문서는 "StoryTelling-based Personalized Learning Platform" 시스템을 개발하기 위한 요구사항을 명시하는 것을 목적으로 한다. 이 시스템은 스토리텔링을 기반으로 한 학습 플랫폼으로, 사용자에게 개인화된 학습 경험을 제공한다. 사용자는 자신이 원하는 캐릭터를 선택하고, 스토리텔러가 이야기를 들려주면서 캐릭터와 함께 학습할 수 있다는 점이 큰 차별화된 특징이다. 본 문서는 해당 시스템의 개요를 설명하고, UI/UX, 서버, 데이터베이스 등을 구성하는 요소들에 대한 요구사항을 정리하며, 이를 토대로 프로젝트를 진행할 수 있도록 한다.

본 문서는 주로 Team6을 비롯한 소프트웨어공학개론 강의의 교수, 조교, 수강자들을 대상으로 하며, 교육 및 비상업적 목적으로 열람 및 사용할 수 있다.

* 1. **SCOPE**

“StoryTelling-based Personalized Learning Platform” 시스템은 사용자에게 개인화된 학습 경험을 제공하기 위한 스토리텔링 기반의 학습 플랫폼이다. 본 시스템은 캐릭터 선택 및 이야기 속 캐릭터와 함께 학습하는 기능을 제공한다. 이러한 기능을 효과적으로 구현하기 위해서, front-end와 back-end에 대한 전반적인 기능이 필요하다.

이에 따라, “StoryTelling-based Personalized Learning Platform” 시스템은 직관적이고 사용하기 쉬운 UI/UX를 제공해야 한다. 또한, 기본적으로 사용자가 시스템 전반을 사용함에 있어서 학습에 방해가 되거나 불편을 느낄 수 있는 기능적인 오류는 있어서는 안 된다. 또한, 사용자 맞춤형 추천모델을 활용하여 사용자의 실력을 향상시킬 수 있는 방식으로 학습 능률을 올릴 수 있는 학습 자료 및 문제, 그리고 그에 따른 풀이를 제공하는 것도 중요하다. 이하 문서에서는 “StoryTelling-based Personalized Learning Platform” 시스템의 front-end와 back-end에 대한 기능 상세를 설명한다.

* 1. **DEFINITION, ACRONYMS, AND ABBREVIATION**

아래의 표는 본 문서에서 사용된 약어와 그 설명을 나타낸다.

**Table 1. Acronyms/Abbreviation**

|  |  |
| --- | --- |
| Acronyms/Abbreviation | Description |
| UI/UX | User Interface/User Experience |
| GUI | Graphic User Interface |
| OS | Operating System |
| DB | Database |
| URL | User Resource Locator |
| HTTP | HyperText Transfer Protocol |
| API | Application Programing Interface |
| AI | Artificial Intelligence |

아래의 표는 본 문서에서 사용한 용어와 그 정의를 나타낸다.

**Table 2. Terms and definitions**

|  |  |
| --- | --- |
| Term | Definition |
| Front-end | Front-end란 사용자가 이용하기 쉽도록 볼 수 있게 해 주는 UI를 뜻한다. Front-end는 UI와 UX 최적화에 초점을 맞춘다. |
| Back-end | Back-end란 서버와 데이터베이스 등 사용자가 직접 보지 못하는 영역을 말한다. 즉, 프론트엔드 사용자의 interaction을 처리하기 위해 DB 작업 등이 포함된다. |
| Django | Django는 웹사이트 개발을 위한 python 웹 프레임워크다. 무료 오픈소스이다. |
| Firebase | Firebase는 구글에서 제공하는 웹 호스팅 개발 플랫폼이다. Firebase JavaScript SDK로 웹 어플리케이션이 구현 가능하다. |
| ChatGPT | ChatGPT는 Pre-trained Transformer(GPT)와 Chat의 합성어로, OpenAI에서 개발한 프로토타입 대화형 인공지능 챗봇이다. 지도학습과 강화학습을 모두 사용해 파인 튜닝되었다. |
| KoAlpaca-LoRA | Korean Alpaca Model based on Stanford Alpaca로, 기본적으로 Stanford Alpaca와 동일한 학습을 진행하면서도 한국어를 이해할 수 있는 Alpaca 모델이다. |
| Actor | Actor는 use case에서 시스템과 상호작용하는 user, device, program이다. |
| Stimulus | 시스템에 입력되는 event 또는 input이다. |
| Response | Stimulus에 대한 응답이다. |

* 1. **REFERENCES**
* “비전공자도 이해할 수 있는 프론트엔드, 백엔드의 모든 것 - 입문편”, 스파르타 코딩클럽, <https://spartacodingclub.kr/blog/frontend-backend-1>
* “Django 소개”, <https://developer.mozilla.org/ko/docs/Learn/Server-side/Django/Introduction>
* “JavaScript 프로젝트에 Firebase 추가”, Firebase, <https://firebase.google.com/docs/web/setup?hl=ko>
* “ChatGPT”, 위키백과, <https://ko.wikipedia.org/wiki/ChatGPT>
* “KoAlpaca”, <https://github.com/Beomi/KoAlpaca>
* “Use Case Analysis: How to Identify Actors?”, <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/how-to-identify-actors/>
  1. **OVERVIEW**

본 문서는 1장에서 “StoryTelling-based Personalized Learning Platform” 서비스의 목적과 범위, 그리고 용어에 대해 정리하였다. 2장에서는 product의 전반적인 description 및 function들을 소개한다. 더 나아가 3장은 external requirements를 명세하며, User inferface와 hardware inferface를 정리한다. 마지막으로 4장 system requirements에서는 functional 및 non-functional requirements 등의 섹션에서 다양한 관점의 diagram을 통하여 시스템의 구조를 상세화한다.

1. **OVERALL DESCRIPTION**
   1. **PRODUCT PERSPECTIVE**

4차 산업혁명. 인공지능, 빅데이터, 사물인터넷, 클라우드 컴퓨팅 등의 첨단 기술이 혁신적으로 결합하여 새로운 가치를 창출하는 시대가 도래했다. 이러한 시대에서 코딩을 배우는 것은 단순히 컴퓨터로 프로그래밍을 하는 방법을 익히는 것이 아니라, 창의적이고 문제해결적인 사고방식을 갖추는 것이다. 코딩은 미래의 모든 직업과 관련된 기본적인 역량이 되어가고 있다. 코딩을 배우는 것은 자신의 필드에서, 전문성과 다양성을 강화시키고, 새로운 기회와 도전에 대응할 수 있는 사람이 되는 데에 도움이 된다. 이러한 시대 흐름에 맞게, 코딩 교육을 배우는 사람들이 날로 늘어나고 있다.

하지만 단순히 강의를 듣거나 문제를 풀기만 하는 것은 지루하고 효과적이지 않을 수 있다. 강의를 듣는 것은 수강자마다 다른 수준이 고려되지 않고, 문제만을 푸는 것은 충분한 학습과 피드백이 없다면 진행하기도 어렵고 흥미를 잃기 쉽다.

그래서 우리는 코딩 학습 사이트 '코드테일'을 제안한다. 코드테일은 학습지와 이야기를 활용한 코딩 학습 사이트로, 다양한 주제와 난이도의 학습지를 제공하고, 문제를 풀이하고 피드백 받을 수 있는 플랫폼이다. 코드테일을 통해 코딩을 배우면, 강의나 문제만으로는 자칫 흥미가 떨어질 수 있는 교육과정을 학습지와 이야기 형식을 통해, 흥미롭게 배울 수 있다. 이와 같은 방식으로 코드테일은 코딩에 대한 흥미와 동기를 높여주고, 실시간 피드백을 제공한다. 코드테일은 코딩을 배우는 모든 유저들에게 유용하고 즐거운 학습 경험을 선사할 것이다.

코드테일을 이용하면 학생들은 웹 사이트에서 여러 가지 언어를 쉽게 배울 수 있는 사용자 맞춤 난이도의 학습지를 제공받는다. 이를 통해 배운 알고리즘을 풀어볼 수 있는 연습 문제를 마찬가지로 제공합니다. 또한, 여기서 그치지 않고 효율적 풀이법도 제시하여, 코딩 역량을 높이는 데 도움을 줍니다. 레벨테스트를 통하여 본인의 역량 향상에 따른 맞춤 학습을 제공한다. 여기에 교강사도, 퀴즈를 올리며 수강자에게 필요한 정보를 제공할 수 있다.

* 1. **PRODUCT PERSPECTIVE**

**Table 3. Table of Product Functions**

|  |  |  |
| --- | --- | --- |
| **Function Requirement No.** | **Name of Function** | **Function Description** |
| F1 | 사용자 모드 선택 | 사용자가 원하는 모드를 선택할 수 있게 한다. |
| F2 | 사용자 정보 수정 | 사용자가 입력했던 정보를 수정할 수 있도록 한다. |
| F3 | 사용자 등록 | 사용자가 본인의 정보를 DB에 등록한다. |
| F4 | 스토리 텔러 선택/변경 | 사용자가 스토리 텔러를 선택하고 후에 다시 변경할 수 있다. |
| F5 | 학습지 문제 추가/삭제 | 사용자(선생님)가 학습지 제작 시 생성된 문제를 추가 및 삭제할 수 있다. |
| F6 | 학습지 문제 수정 | 사용자(사용자)가 생성된 문제를 변경할 수 있다. |
| F7 | 커리큘럼 SKIP 기능 | 사용자가 지정된 커리큘럼에서 특정 과정을 SKIP할 수 있다. |
| F8 | 코드 에디터 제공 | 코드 에디터를 제공한다. |
| F9 | 코드 복사 | 코드 에디터에 작성한 코드를 복사할 수 있다. |
| F10 | 파일 불러오기 | 시스템 DB에 저장된 코드 파일을 불러와 사용할 수 있다. |
| F11 | 코드 되돌리기 | 초기(skeleton code)의 코드로 되돌릴 수 있다. |
| F12 | 코드 중간 저장 | 사용자가 임의로 코드를 저장할 수 있다. |
| F13 | 테스트케이스 검증/실행 | 테스트케이스를 검증하고 실행한다. |
| F14 | 테스트케이스 채점 | 테스트케이스에 대한 내용 및 채점 결과를 제공한다. |
| F15 | 사용자 코드 채점 | 코드의 기능 및 효율율에 대한 채점 기능을 제공한다. |
| F16 | 코드 Diff | 정답 코드와 학생 코드 사이의 Diff 기능을 제공한다. |
| F17 | 코드 설명 | 정답 코드에 대한 설명을 자동으로 제공한다. |
| F18 | 레벨 테스트 제공 | 레벨 테스트를 제공하고 채점한다. |
| F19 | 사용자 수준 정의 | 레벨테스트에 따라 사용자 수준을 정의한다. |
| F20 | 언어 선택 | 사용자가 학습에 사용할 언어를 선택한다. |
| F21 | 피드백 제공 | 사용자의 코드에 대해, 가능한 리팩토링 사항, 오류 정보를 제공한다. |

**2.2.1 LOGIN PAGE**

시스템 접속 시, 처음 화면은 '로그인 페이지'가 나온다.

상단에는 시스템의 이름과 시스템의 대표 캐릭터 이미지가 삽입되어 있다.

로그인을 통하여, 메인 페이지로 진입합니다. 여기서 STUDENT와 INSTRUCTOR는 계정에 따라 나뉜다.

**2.2.2 Register PAGE**

사용자가 시스템에 사용자 정보를 등록하는 페이지이다. Submit 버튼을 통하여 사용자 정보를 DB에 저장한다.

**2.2.3 Mode Select PAGE**

로그인 버튼을 누를 시, 'mode select 페이지'가 나온다. 사용자가 학생모드 혹은 선생님모드를 선택할 수 있다.

**2.2.4 Language Select PAGE**

사용자(학생 및 선생님)가 Register 후, 학습할 언어를 선택하는 페이지이다. 학습할 언어를 선택하면 DB에 해당 언어를 사용자 정보에 등록하고, 언어 학습 단계로 넘어간다.

**2.2.5 Learning PAGE**

사용자(학생)이 언어를 학습할 수 있는 페이지이다. AI 모델이 DB에서 부터 가져온 커리큘럼과 진행도를 기반으로, 스토리텔링 기반의 학습지를 생성하여 학생이 이를 학습한다. Progress바가 표시되고, 학생의 진행도에 따라 올라가게 된다.

**2.2.6 Coding PAGE**

사용자(학생)이 문제를 받고, 코딩하는 페이지이다. 코드를 작성 후 Run 버튼을 통하여, 코드를 실행해 볼 수 있다. 또한 결과를 확인해 볼 수 있다. 사용자가 충분히 문제를 해결했다고 생각하면, 제출 버튼을 통해 코드를 제출할 수 있다.

**2.2.7 Grading PAGE**

사용자가 코드를 제출 후, 결과를 보는 페이지이다. Test Case 데이터를 DB로 부터 불러와, ChatGPT API를 통해 채점을 하고 결과를 보여준다. 또한 API를 통하여, 리팩토링 가능한 사항이나, 고쳐야 할 오류를 알려주게 된다.

**2.2.9 Level Test PAGE**

사용자(학생)이 수준별 학습을 위해 수준을 진단하는 페이지이다. AI 모델이 레벨테스트를 진행하고, 결과에 따라 학생의 수준을 DB에 저장하게 된다.

**2.2.10 Teacher PAGE**

사용자(선생님)이 AI 모델을 통해, 학습용 문제를 만들 수 있는 페이지이다. 사용자(선생님)이 Question 조건을 입력한 후 Question Generate 버튼을 눌러 AI 모델을 통해 만들어진 문제를 등록할 수 있다. 선생님모드일 경우에만, 해당 페이지에 접근할 수 있다.

**2.3. USER CLASSES AND CHARACTERISTICS**

**2.3.1. SYSTEM ADMINISTRATOR**

‘SYSTEM ADMINISTRATOR’은 시스템 관리자로, 이 시스템을 운영 및 관리, 유지 보수를 총괄하는 사람으로서 학생 또는 교강사 등의 사용자로부터의 요구사항을 처리하는 사람을 의미한다. 시스템 관리자는 백엔드와 React를 바탕으로 한 프론트엔드 웹 서비스를 개발하는데 능숙해야 한다. 또한 REST API를 사용한 백엔드와 프론트엔드 사이의 통신에 대한 전반적인 이해도 가지고 있어야 한다. 이러한 지식과 활용능력을 통해 사용자들이 겪는 문제를 해결하기 위한 빠른 판단과 대처 능력을 가지고 있어야 하는 것도 시스템 관리자의 필요 역량이다.

**2.3.2. STUDENT**

‘STUDENT’는 학생으로, 이 시스템을 이용하여 코딩에 대한 개인의 지식과 역량을 키우기 위해 공부하고자 하는 사람을 의미한다. 학생은 웹 브라우저를 통해 이 시스템에 접속하여 기본적인 GUI 및 코드 에디터를 활용할 수 있어야 한다.

**2.3.3. INSTRUCTOR**

‘INSTRUCTOR'은 교강사로, 이 시스템을 이용하여 자신이 수업했던 내용에 대한 Quiz/질문지를 생성하고자 하는 사람을 의미한다. 교강사는 웹 브라우저를 통해 이 시스템에 접속하여 기본적인 GUI를 다룰 줄 알아야 하며, 본인이 수업했던 내용에 대한 지식과 강의 자료를 가지고 있어야 한다. 원하는 Quiz/질문지를 생성하기 위해서 그 질문에 대한 방향성을 확실히 해야 하므로 꽤 정확하고 구체적인 강의 자료를 가지고 있어야 한다.

**2.4. OPERATING ENVIRONMENT**

* Window 7
* Window 10
* Window 11
* Mac OS

**2.5. DESIGN AND IMPLEMENTATION CONSTRAINTS**

본 아래의 제약 사항들을 준수하여 개발, 운영, 사용되어야 한다.

* 가능한 오픈소스를 사용할 것.
* 사용자가 쉽게 사용가능 하도록 구성할 것.
* 운영체제 상 다른 시스템에 악영향을 주지 않게 설계할 것.
* 사용자의 정보에 악영향을 줄 수 있는 구조를 피할 것.
* 버전 제어 시스템을 사용하여 협업할 것.
* 사용자가 추가 설치사항 없이 사용가능 하게 구성할 것.
* 차후 유지보수를 고려한 설계

**2.6. ASSUMPTIONS AND DEPENDENCIES**

본 문서의 시스템은 웹 브라우저에서 사용되는 것은 기본 전제로 한다. 이에 따라 인터넷 망이 안정적으로 연결되어 있어야 시스템을 활용할 수 있으므로, 인터넷 망이 안정적이라는 상태를 가정한다.

또한 한국어로 시스템을 제공하므로 유저가 한국어를 사용하는 데 어려움이 없음을 가정한다. 또한 제시된 운영체제를 만족하는 환경에서 정상적으로 작동됨을 가정한다. 그 외 환경에서는 시스템 활용이 보장되지 않는다.

# **3. EXTERNAL INTERFACE REQUIREMENTS**

## **3.1. USER INTERFACE**

**Table 4. User Interface of Main Page**

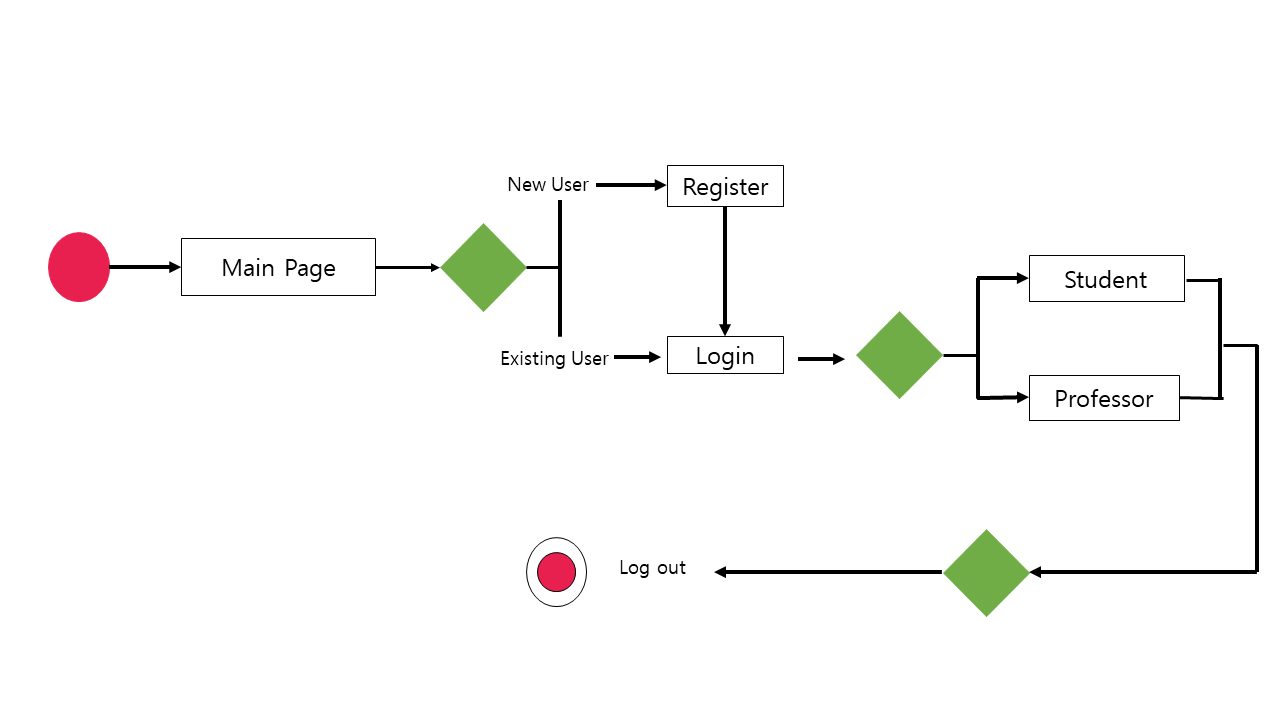
|  |  |
| --- | --- |
| Name | Main Page |
| Description | 사용자가 웹사이트에서 가장 먼저 만나는 초기 페이지로 학생/교수 모드 선택과 로그인/회원가입을 진행할 수 있는 페이지이다. |
| Input/Output | 유저/서버 |
| Screen format  / Composition | 1. 메인페이지는 크게 Header와 Context, footer로 나누어져 있다. 2. Header에는 좌측부터 페이지 로고, 로그인,회원가입으로 구성되어 있다. 3. Context는 학생/교수를 선택할 수 있는 Mode Select Button이 존재한다. 4. Footer에는 그 외의 정보(팀원, 참고 자료)가 존재한다. |
| Stimulus | 1. 로고 버튼을 눌러 초기 홈페이지 상태로 돌아간다. 2. 로그인 버튼을 눌러 로그인 페이지로 넘어간다. 3. 회원가입 버튼을 눌러 회원가입 페이지로 넘어간다. 4. 학생 모드를 선택해 학생 페이지로 넘어간다. 5. 교수 모드를 선택해 교수 페이지로 넘어간다. |
| Response | 1. 학생/교수 모드를 선택했을 때 로그인이 되어 있지 않은 상태이면 로그인 페이지로 넘어간다. |

**Table 5. User Interface of Main Page - Login**

| Name | Login Page |
| --- | --- |
| Description | 사용자가 메인페이지에서 로그인 버튼을 눌렀을 때 나타나는 창으로 아이디와 비밀번호를 입력받아 로그인을 할 수 있는 페이지이다. |
| Input/Output | 유저/서버 |
| Screen format  / Composition | 1. 위부터 로고 - 아이디 입력창 - 비밀번호 입력창 - 로그인 버튼 - 회원가입 버튼으로 이루어져 있다. |
| Stimulus | 1. 회원가입 버튼을 누를 시 회원 가입 페이지로 넘어간다. |
| Response | 1. 입력한 아이디와 비밀번호가 일치하지 않거나 신규 사용자일 경우 “아이디와 비밀번호가 일치하지 않거나 신규 사용자 입니다”라는 문구가 출력된다. 2. 아이디와 비밀번호가 서버 DB에 존재하는 경우 로그인이 성공적으로 진행되며 메인페이지로 돌아가게 된다. |

**Table 6. User Interface of Main Page - Register**

|  |  |
| --- | --- |
| Name | 회원가입 페이지 |
| Description | 사용자가 웹사이트에 회원가입을 하기 위한 정보를 입력하는 페이지이다. |
| Input/Output | 유저/서버 |
| Screen format  / Composition | 1. 위에서부터 로고 - 아이디 입력창 - 비밀번호 입력창 - 비밀번호 재확인 창 - 본인확인 이메일 창 - 가입하기 버튼으로 이루어져 있다. |
| Stimulus | 1. 가입하기 버튼을 눌러 가입을 진행할 수 있다. |
| Response | 1. 비밀번호와 비밀번호 재확인에 입력된 문자가 다를 경우 “비밀번호를 재확인 해 주세요”라는 문구가 출력된다. 2. 입력된 정보가 성공적으로 DB에 전달된 경우 “성공적으로 가입이 되었습니다”라는 문구가 출력된다. |



**Figure 1. Main Page Diagram**

**Table 7. User Interface of Student Level Test Page**

|  |  |
| --- | --- |
| Name | 학생 레벨 테스트 페이지 |
| Description | 사용자가 학생 모드에서 나오는 초기 페이지이며, 학생의 수준을 평가할 문제들을 나타낸다 |
| Input/Output | 유저/서버 |
| Screen format  / Composition | 1. 페이지는 header, context와 footer로 나눠져있다 2. header에는 해당 학생의 이름과 로그아웃 버튼으로 구성되어 있다 3. context에는 ai가 형성한 레벨 테스트를 진행하는 Start Test 버튼이 있다 4. footer에는 그 외의 정보(팀원, 참고 자료)가 존재한다 |
| Stimulus | 1. 로고 버튼을 눌러 초기 홈페이지 상태로 돌아간다 2. 로그아웃 버튼을 눌러 로그아웃을 한다 3. Start Test 버튼을 눌러 레벨 테스트를 시작한다. |
| Response | 1. 만약 level test를 완료하였을 때 start test를 누를 경우 “레벨 테스트 진행 완료하였습니다” 문구가 뜬다. |

**Table 8. User Interface of Student Code Test Page**

|  |  |
| --- | --- |
| Name | 학생 코드 테스트 페이지 |
| Description | 사용자가 학생 모드에서 특정 코드 테스트를 선택하였을 때 나오는 페이지이다 |
| Input/Output | 유저/서버 |
| Screen format  / Composition | 1. 페이지는 header, context와 footer로 나눠져있다 2. header에는 해당 학생의 이름과 로그아웃 버튼으로 구성되어 있다 3. context에는 선택한 코드 테스트의 문제가 나오며, 아래에는 코드 작성이 가능한 ide가 있다. 학생이 선택한 언어를 바탕으로 코드 작성이 진행되며, ide밑에는 run 버튼이랑 submit버튼, review 버튼이 있으며, grading 결과도 있다. 그 밑으로는 next 버튼이 있다. 4. footer에는 그 외의 정보(팀원, 참고 자료)가 존재한다 |
| Stimulus | 1. 로고 버튼을 눌러 초기 홈페이지 상태로 돌아간다 2. 로그아웃 버튼을 눌러 로그아웃을 한다 3. Next 버튼을 선택해 다음 진행 상황으로 넘어간다. 4. Run 버튼을 선택해 코드를 run하며, 컴파일 결과를 보여준다. 5. Submit 버튼을 누를 경우 기존의 test case 와 비교하여 grading을 진행한다. 6. Review 버튼을 누를 경우 해당 코드의 오류나 refactoring 결과를 보여준다 |
| Response | 1. 만약 grading 이전에 review 버튼을 누를 경우, “코드를 제출하세요”라는 문구가 뜬다 2. 만약 작성된 코드가 없는 상황에서 run 버튼을 누를 경우, “코드를 작성하세요”라는 문구가 뜬다. 3. 만약 작성된 코드가 없는 상황에서 버튼을 누를 경우, “코드를 작성하세요”라는 문구가 뜬다. |

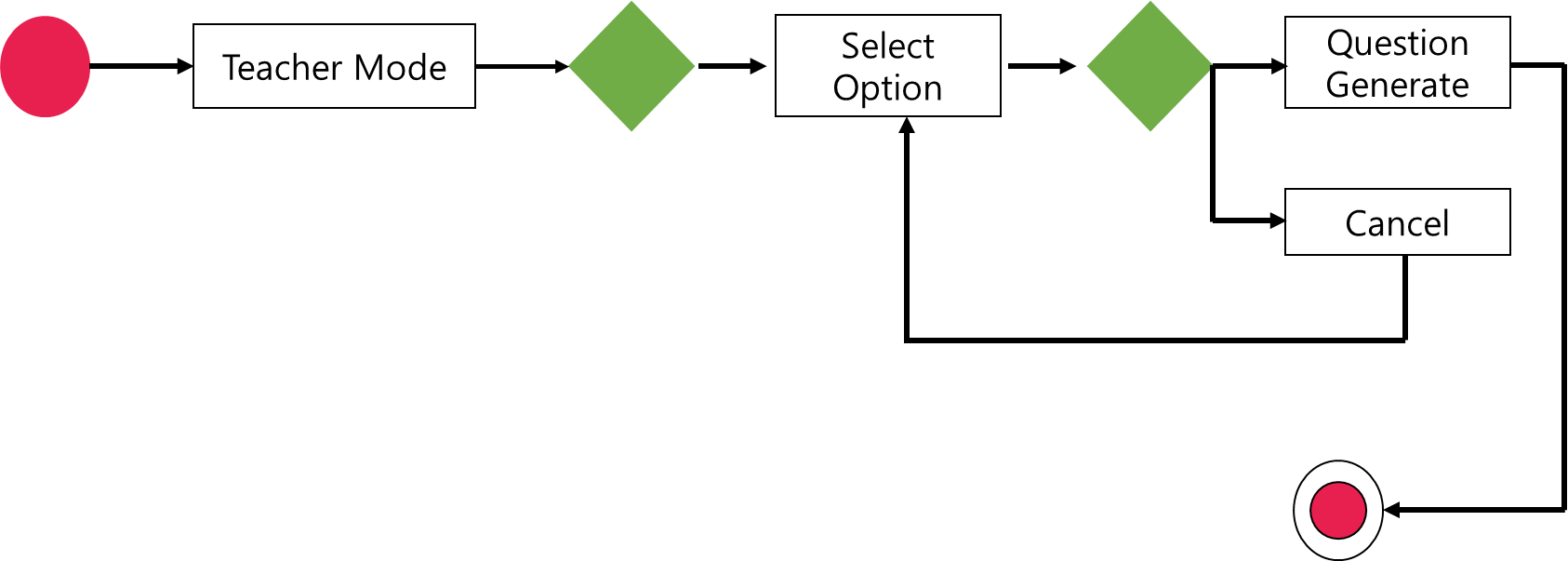
**Table 9. User Interface of Student Curriculum Page**

|  |  |
| --- | --- |
| Name | 학생 커리큘럼 페이지 |
| Description | 사용자가 학생 모드로 선택하였을 때 시스템에 접근하면 볼 수 있는 초기 페이지이다. |
| Input/Output | 유저/서버 |
| Screen format  / Composition | 1. 페이지는 header, context와 footer로 나눠져있다 2. header에는 해당 학생의 이름과 로그아웃 버튼으로 구성되어 있다 3. context에는 해당 학생의 커리큘럼이 있으며, table의 형식으로 되어있다. 진도가 진행되는 것을 옆의 check 표시를 통해 확인할 수 있다. 또한, 전체 진행도를 bar 형태로 나타내어 학생의 진행 상황을 보여준다 4. footer에는 그 외의 정보(팀원, 참고 자료)가 존재한다 |
| Stimulus | 1. 로고 버튼을 눌러 초기 홈페이지 상태로 돌아간다 2. 로그아웃 버튼을 눌러 로그아웃을 한다 3. 커리큘럼의 내용을 선택해 해당 code test 페이지로 넘어간다 |
| Response | 1. 만약 완료된 부분을 선택할 시 “완료하였습니다” 문구가 뜬다. |

****

**Table 10. User Interface of Teacher Mode Page**

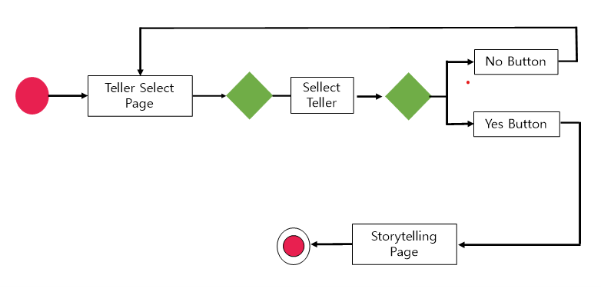
|  |  |
| --- | --- |
| Name | 선생님모드 페이지 |
| Description | 사용자가 선생님 모드로 선택하였을 때 시스템에 접근하면 볼 수 있는 초기 페이지이다. |
| Input/Output | 유저/서버 |
| Screen format  / Composition | 1. 페이지는 header, context와 footer로 나눠져있다 2. header에는 해당 학생의 이름과 로그아웃 버튼으로 구성되어 있다 3. context에는 생성할 question의 조건들을 기입하는 덱스트바가 있다. 밑에는 qeustion generate 버튼과 cancel 버튼이 있다 4. footer에는 그 외의 정보(팀원, 참고 자료)가 존재한다 |
| Stimulus | 1. 로고 버튼을 눌러 초기 홈페이지 상태로 돌아간다 2. 로그아웃 버튼을 눌러 로그아웃을 한다 3. question generate 버튼을 선택할 시 조건에 맞는 문제가 출력된다 4. cancel 버튼을 선택할 시 생성된 문제가 삭제된다 |
| Response | 1. 문제가 없는데 cancel버튼을 선택할 시 “취소할 문제가 없습니다”라는 문구가 출력된다 |

****

**Table 11. User Interface of Teller Select Page**

|  |  |
| --- | --- |
| Name | Teller Select Page |
| Description | 각 Teller의 이미지와 그 Teller들마다 가지고 있는 스토리텔링 예시가 적혀 있는 페이지로, 사용자가 원하는 캐릭터를 클릭하면 사용자의 Teller로 지정된다. |
| Input/Output | 유저/서버 |
| Screen format/ Composition | 1. 화면에 바둑판 모양으로 여러 Teller 이미지가 표시되고, 이미지 우측에는 해당 Teller의 스토리텔링 예시가 표시된다.Teller 이미지 우측에 적힌 스토리텔링은 사용자가 앞으로 배우게 되는 커리큘럼의 내용이 적혀 있어 사용자의 Teller 선택을 돕는다.2. Teller와 스토리텔링 예시가 적혀 있는 버튼을 클릭하면 팝업창이 생기며 '이 Teller를 선택하시겠습니까?' 라는 문구와 함께 아래 Yes 버튼과 No 버튼이 나타난다. |
| Stimulus | 1. 여러 Teller들 중 한 Teller을 선택하여 버튼을 누른다.2. 팝업창의 Yes 버튼과 No 버튼 중 원하는 버튼을 누른다. |
| Response | 1. 원하는 Teller을 클릭하면 팝업창이 생기며 '이 Teller를 선택하시겠습니까?'라는 문구와 함께 Yes/No 버튼이 나타난다.2. Yes 버튼을 누르면 해당 정보를 데이터 베이스에 저장하고 'Storytelling Page'로 넘어간다. No 버튼을 누르면 팝업창이 사라지며 'Teller Select Page'로 돌아간다. |

## 



## **Table 12. User Interface of Storytelling Page**

|  |  |
| --- | --- |
| Name | Storytelling Page |
| Description | 사용자(학생)가 커리큘럼에 따라 학습지들을 풀어볼 수 있는 페이지로, 자신의 현재 진행 상황에 맞는 학습지를 클릭하여 학습을 진행할 수 있다. 이전 단계의 학습이 완료되지 않은 상태라면 다음 단계의 학습을 진행할 수 없다. |
| Input/Output | 유저/서버 |
| Screen format/ Composition | 1. 화면 좌측 상단에는 현재 사용자가 선택한 Teller 이미지가 표시되고, 우측 상단에는 현재 학습 진행률을 보여주는 progress bar가 표시된다.2. 화면 중앙에는 징검다리 모양으로 공책 모양의 버튼들이 있고, 버튼 아래에 어떤 코딩 학습인지 주제(Stack, Queue 등)가 적혀있다. 징검다리는 학습에 알맞은 커리큘럼 순서로 구성되며, 이전 징검다리를 완료하지 않으면 다음 징검다리로 건너갈 수 없다.3. 공책 모양의 버튼에 마우스 포인터를 올려놓으면 팝업창이 나타난다. 팝업창의 상단부에는 주제가 표시되며, 하단에는 주제에서 배우게 될 목차들이 리스트 형태로 표시된다.4. 공책 모양의 버튼을 누르면 해당 주제에 해당하는 학습지가 나타난다. |
| Stimulus | 1. 공책 모양의 버튼에 마우스 포인터를 올려놓는다.2. 공책 모양의 버튼을 누른다. |
| Response | 1. 상단에는 주제가, 하단에는 주제에서 배우게 될 목차들이 적혀있는 팝업창이 나타난다. 마우스 포인터를 다른 곳으로 이동하게 되면 이 팝업창은 사라진다.2. 공책 모양의 버튼에 해당하는 주제의 학습지가 나타난다. |

## 

## 

## **3.2. HARDWARE INTERFACE**

**Table 13. Hardware Interface for User**

|  |  |
| --- | --- |
| Name | 사용자와 시스템 간의 상호작용을 위한 하드웨어 |
| Description | 사용자가 하드웨어를 통해 웹사이트에게 보내는 입/출력값 |
| Input/Output | 유저(하드웨어)/웹사이트(컴퓨터 또는 시스템) |
| Composition | 1. 키보드 2. 마우스 3. 모니터 4. 스피커 |
| Stimulus | 1. 키보드를 통한 입력 2. 마우스 클릭을 통한 입력 3. 모니터에 보이는 출력값 4. 스피커로 나오는 음성 출력값 |

## **3.3. SOFTWARE INTERFACES**

**Table 14. Software Interface of DB**

|  |  |
| --- | --- |
| Name | SQLite |
| Description | 사용자 정보와 AI 모델에서 생성된 정보들을 입출력하기 위한 데이터 베이스를 관리/소통하기 위한 데이터 베이스 매니지먼트 시스템 |
| Input/Output | 서버/데이터베이스  AI Model/데이터베이스 |

**Table 15. Software Interface of Front-end**

|  |  |
| --- | --- |
| Name | React |
| Description | 유저 인터페이스 생성을 위한 자바스크립트 라이브러리로 시각화와 상호작용을 위한 역할을 한다. 유저 인터페이스 요구사항을 정의하는데 쓰이며 버튼과 메뉴의 위치, 색상과 글꼴, 인터페이스의 레이아웃, 디자인 및 동작에 관여한다. 리액트는 또한 다양한 사용자 인터페이스 상태를 정의하고 각각의 전환 방법을 정의하는 데 사용될 수 있다. 예를 들어 사용자가 로그인 버튼을 클릭하면 인터페이스는 요청된 데이터가 가져올 때까지 로딩 상태로 전환되고 만약 회원 정보가 데이터베이스와 일치하지 않는다면 “아이디와 비밀번호가 일치하지 않거나 신규 사용자입니다”라는 문구를 출력할 수 있다. |
| Input/Output | 서버/데이터베이스  유저/서버 |

**Table 16. Software Interface of Back-end**

|  |  |
| --- | --- |
| Name | Django |
| Description | Django는 일반적으로 웹 애플리케이션을 구축하는 데 사용되는 웹 프레임워크로 사용자 요청을 처리하고 적절한 응답을 생성하는 소프트웨어 시스템의 서버 측 구성 요소를 정의한다. Django는 시스템의 서버 측 구성 요소의 인터페이스 요구사항을 정의하고 서버가 클라이언트에 노출할 RESTful API와 데이터 모델과 데이터베이스 스키마를 포함한다. 또한 웹사이트의 비즈니스 로직를 위한 데이터 처리 및 사용자 입력에 대한 시스템의 워크플로우, 알고리즘 및 규칙을 정의한다. |
| Input/Output | 서버/데이터베이스  유저/서버 |

**3.4. COMMUNICATION INTERFACES**

**Table 17. Communication Interfaces for Web Platform**

|  |  |
| --- | --- |
| Name | 웹 플랫폼을 위한 커뮤니케이션 인터페이스 |
| Description | 1. 플랫폼은 클라이언트 측과 서버 측 구성 요소 간의 통신에 RESTful API를 사용하여 구축된다. 2. 플랫폼은 모든 API 요청 및 응답에 HTTP 프로토콜을 사용해야 한다. 3. 플랫폼은 다음과 같은 HTTP 메서드를 지원해야 한다: GET, POST, PUT 및 DELETE. 4. 플랫폼은 모든 API 데이터 페이로드에 대해 JSON 형식을 사용해야 한다. 5. 플랫폼은 모든 API 요청 및 응답에 대한 적절한 오류 처리를 포함해야 한다. 6. 플랫폼은 표준 암호화 알고리즘을 사용하여 모든 중요 데이터를 암호화해야 한다. |
| Input/Output | 서버/데이터베이스  유저/서버 |

# 

# **4. SYSTEM REQUIREMENTS**

## **4.1. FUNCTIONAL REQUIREMENTS**

### **4.1.1. USE CASE**

**Table 18**. **Functional requirements abstraction**

|  |  |  |
| --- | --- | --- |
| Functional Requirement No. | Name of Function | Function Description |
| F1 | Curriculum | AI 모델이 개인화된 커리큘럼을 생성한다. |
| F2 | Progress | 학생이 학습 진행도를 확인한다. |
| F3 | Storytelling | AI 모델이 학생에게 스토리텔링 기반의 학습지를 생성한다. |
| F4 | Question Generation | 선생님이 학습용 문제를 생성한다. |
| F5 | Code Edit & Run | 학생이 코드를 작성(Edit)하고 실행(Run)한다. |
| F6 | Code Submit | 학생이 작성한 코드를 제출한다. |
| F7 | Code Grading | 학생이 제출한 코드를 실행 후 데이터베이스에 있는 Test Case Data를 받아서 이를 ChatGPT API를 통해 채점해주는 기능이다. |
| F8 | Code Review | 학생이 입력한 코드를 Grading후, 코드에서 refactoring할 점이나 오류가 있을 시 ChatGPT API를 활용하여 알려주는 단계이다. |
| F9 | Level Test | 학생이 AI 모델이 만들어준, 수준을 평가할 수 있는 레벨테스트를 진행하는 단계이다. 시험 결과에 따라 학생의 수준을 판단한다. |
| F10 | Teller Select | 학생이 AI 모델이 생성해준 각 캐릭터의 특성에 따른 스토리텔링 교육자료 예시를 보고, 캐릭터를 고르는 단계이다. |
| F11 | Language Select | 사용자(학생 및 선생님)가 register이후 학습시 활용할 언어를 고르는 단계이다 |
| F12 | Login | 사용자가 로그인화면에서 사용자의 id와 password를 입력 후 로그인을 하는 단계이다. |
| F13 | Register | 사용자(학생 및 선생님)이 시스템에 사용자 정보를 등록하는 단계이다. |
| F15 | Mode | 사용자(학생 및 선생님)이 로그인 이후 학생모드 혹은 선생님 모드로 진입하는 것을 의미한다. 선택에 따라 데이터베이스에서 사용자의 데이터를 불러온다. |

**Table 19. Use Case of Curriculum**

|  |  |
| --- | --- |
| Name | Curriculum |
| Actors | 학생, AI 모델, 데이터베이스 |
| Description | AI 모델이 개인화된 커리큘럼을 생성한다. |
| Data | Curriculum |
| Stimulus | 레벨 테스트 후 AI 모델로 정보가 전달될 때 |
| Response | AI 모델이 데이터베이스로부터 학생에게 개인화된 커리큘럼을 생성하여, 데이터베이스 저장 및 학생에 전달한다. |
| Comments | 학생의 사용자 정보를 기반으로 개별화된 커리큘럼을 생성 후, 커리큘럼 정보를 데이터베이스에 저장한다. |

**Table 20. Use Case of Progress**

|  |  |
| --- | --- |
| Name | Progress |
| Actors | 학생, AI 모델, 데이터베이스 |
| Description | 학생이 학습 진행도를 확인한다. |
| Data | Progress |
| Stimulus | 학생이 학습 시작시 & 학습 진행시 |
| Response | 데이터베이스로부터 학생의 커리큘럼을 가져와 학습 진행도를 알려준다. |
| Comments | 퍼센트화해서 학생의 학습 진행도를 알려준다. |

**Table 21. Use Case of Storytelling**

|  |  |
| --- | --- |
| Name | Storytelling |
| Actors | 학생, AI 모델, 데이터베이스 |
| Description | AI 모델이 학생에게 스토리텔링 기반의 학습지를 생성한다. |
| Data | Input(User Information) |
| Stimulus | 학습 진행도 취득시 |
| Response | AI 모델이 데이터베이스에서 취득한 커리큘럼 및 진행도 기반으로, KoAlphaca 스토리텔링 학습지를 생성한다. |
| Comments | 선택한 Teller 기반의 스토리텔링을 형성한다. |

**Table 22. Use Case of Question Generation**

|  |  |
| --- | --- |
| Name | Question Generation |
| Actors | 선생님, AI 모델, 데이터베이스 |
| Description | 선생님이 학습용 문제를 생성한다. |
| Data | Input(Question query) |
| Stimulus | 선생님이 교육자 탭에서 Question 조건을 입력한 후 Question Generate 버튼을 누른다. |
| Response | AI 모델에 Question 모델이 도착하여, Question 생성 후 데이터베이스에 저장된다. |
| Comments | Query handling이후 ChatGPT AI가 가능한 문제를 보여준다. 마음에 들지 않을시 Cancel을 누르고 돌아갈 시 ClearText가 되어있어야 한다. |

**Table 23. Use Case of Code Edit & Run**

|  |  |
| --- | --- |
| Name | Code Edit & Run |
| Actors | 학생, AI 모델 |
| Description | 학생이 코드를 작성(Edit)하고 실행(Run)한다. |
| Data | Input(Code, input of code) |
| Stimulus | 사용자가 Code를 작성하고, Run Button을 누를 시 사용자의 input을 받는다. |
| Response | 코드 작성시 작성한 코드가 코드 편집 탭에 나타난다. 코드 실행시 AI 모델이 input data를 사용한 컴파일 결과를 표시해준다. |
| Comments | 사용자의 input이 필요 없는 문제일 경우 null을 input으로 받을 시, 이를 handling 해줘야 한다. |

**Table 24. Use Case of Code Submit**

|  |  |
| --- | --- |
| Name | Code Submit |
| Actors | 학생, AI |
| Description | 학생이 작성한 코드를 제출한다. |
| Data | Input(Submitted Code) |
| Stimulus | 학생이 Submit 버튼 클릭시 |
| Response | 코드 작성 탭에 있는 코드를 Grading 탭에 전달한다. |
| Comments | 언어 조건(C, Java, Python 등)도 함께 전달한다. 제출 후에도 코드는 텍스트로 남아있어야 한다. |

**Table 25. Use Case of Code Grading**

|  |  |
| --- | --- |
| Name | Code Grading |
| Actors | 학생, 데이터베이스, AI 모델 |
| Description | 학생이 제출한 코드를 실행 후 데이터베이스에 있는 Test Case Data를 받아서 이를 ChatGPT API를 통해 채점해주는 기능이다. |
| Data | Input(Submitted Code), Input from Database(Question’s Test Case) |
| Stimulus | 사용자가 submit 버튼을 누른다. |
| Response | 해당 데이터를 ChatGPT API를 통해 넘겨서 채점 결과를 받아온다. |
| Comments | 문제 데이터와 테스트케이스 데이터가 ChatGPT API에 잘 input data로 들어가야 하고, 결과물을 잘 parsing해야 한다. |

**Table 26. Use Case of Code Review**

|  |  |
| --- | --- |
| Name | Code Review |
| Actors | 학생, AI 모델 |
| Description | 학생이 입력한 코드를 Grading후, 코드에서 refactoring할 점이나 오류가 있을 시 ChatGPT API를 활용하여 알려주는 단계이다. |
| Data | Input(Submitted Code) |
| Stimulus | 사용자가 제출한 코드가 채점을 마치고, Review Code버튼을 누른다. |
| Response | ChatGPT API로 코드 데이터를 보내 결과물을 받고, parsing 후 사.용자에게 보여준다. |
| Comments | Parsing이 제대로 진행되어야 한다. 그리고, 채점 후에만 사용할 수 있도록 해야한다. |

**Table 27. Use Case of Level Test**

|  |  |
| --- | --- |
| Name | Level Test |
| Actors | 학생, 데이터베이스, AI 모델 |
| Description | 학생이 AI 모델이 만들어준, 수준을 평가할 수 있는 레벨테스트를 진행하는 단계이다. 시험 결과에 따라 학생의 수준을 판단한다. |
| Data | Input(Test Choice) |
| Stimulus | 학생이 모델이 생성해준 언어별 문제를 푼다. 다 풀 시 Submit button을 누른다. |
| Response | 채점 후 결과를 보여주고, 사용자의 레벨을 데이터베이스에 저장한다. 이후 커리큘럼을 보여주며 학습화면으로 들어간다. |
| Comments | 문제는 난이도별로 골고루 내야 한다. |

**Table 28. Use Case of Teller Select**

|  |  |
| --- | --- |
| Name | Teller Select |
| Actors | 학생, 데이터베이스, AI 모델 |
| Description | 학생이 AI 모델이 생성해준 각 캐릭터의 특성에 따른 스토리텔링 교육자료 예시를 보고, 캐릭터를 고르는 단계이다. |
| Data | Input(Character Choice) |
| Stimulus | 학생이 캐릭터 버튼을 클릭 후, Yes/No를 누른다. |
| Response | Yes를 누르면 해당 정보를 데이터베이스에 저장한다. No를 누를 시, 캐릭터 선택화면으로 돌아간다. |
| Comments | 캐릭터별로 스토리텔링 교육자료 예시를 만들어두고, 사용자에게 보여줄 수 있어야 한다. |

**Table 29. Use Case of Language Select**

|  |  |
| --- | --- |
| Name | Language Select |
| Actors | 학생, 선생님, 데이터베이스 |
| Description | 사용자(학생 및 선생님)가 register이후 학습시 활용할 언어를 고르는 단계이다 |
| Data | Input(language selection) |
| Stimulus | 학습시 활용할 언어가 쓰여진 버튼을 클릭한다. |
| Response | 데이터베이스에 언어 정보를 저장한 뒤, 언어 학습 단계로 넘어간다 |
| Comments | 언어 정보가 데이터베이스에 각 사용자 별로 잘 저장되어야 한다. |

**Table 30. Use Case of Login**

|  |  |
| --- | --- |
| Name | Login |
| Actors | 학생, 선생님, 데이터베이스 |
| Description | 사용자가 로그인화면에서 사용자의 id와 password를 입력 후 로그인을 하는 단계이다. |
| Data | input(id, password) |
| Stimulus | 사용자가 id, password를 입력한다. |
| Response | Login Button을 누른 뒤 로그인 성공여부를 알려주고, 성공시 mode select로 넘어간다. |
| Comments | 로그인 실패시 에러메시지를 띄우고, cleartext를 해줘야 한다. |

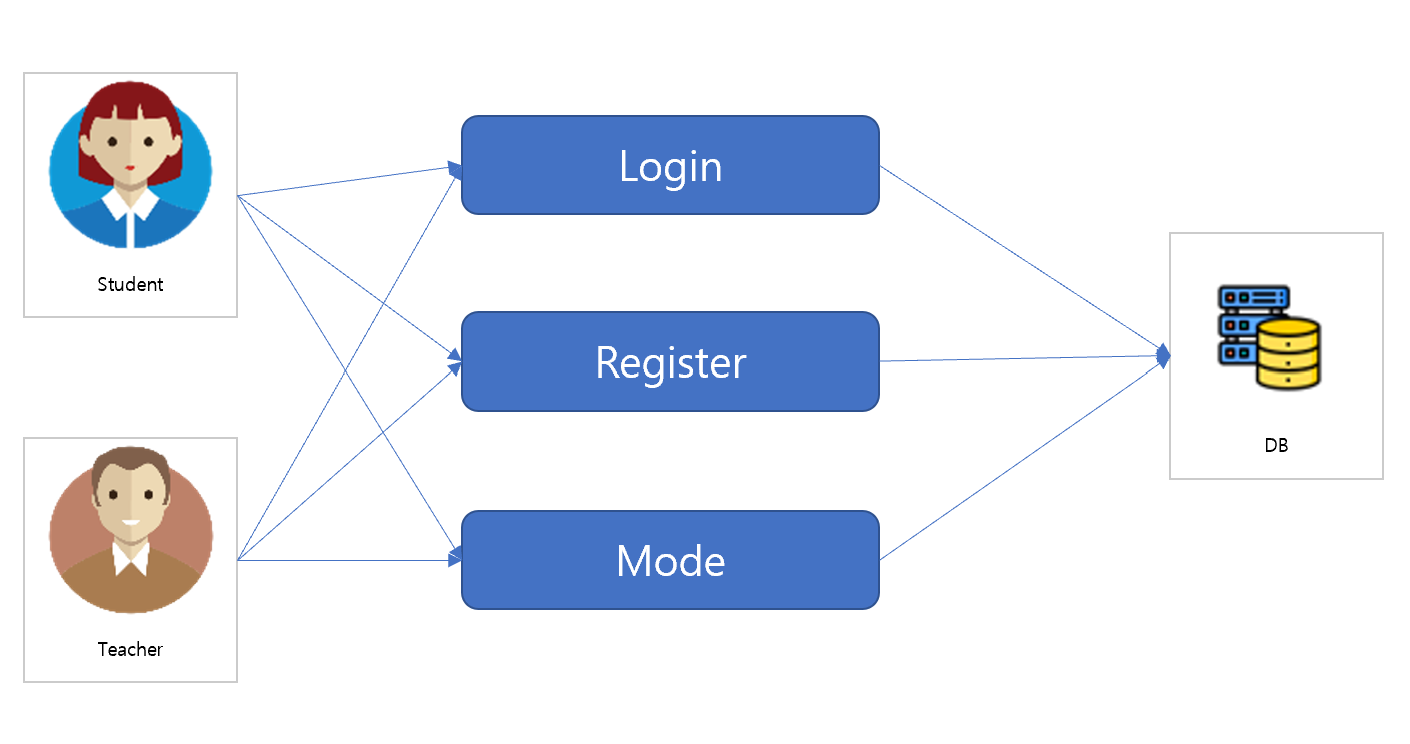
**Table 31. Use Case of Register**

|  |  |
| --- | --- |
| Name | Register |
| Actors | 학생, 선생님, 데이터베이스 |
| Description | 사용자(학생 및 선생님)이 시스템에 사용자 정보를 등록하는 단계이다. |
| Data | Input(User Information) |
| Stimulus | 사용자가 화면에서 사용자의 데이터를 입력한다. |
| Response | submit을 누르면 데이터베이스에 정보가 등.록된다 |
| Comments | 사용자 정보가 빠짐없이 잘 데이터베이스에 각각의 user별로 등록되어 있어야 한다. |

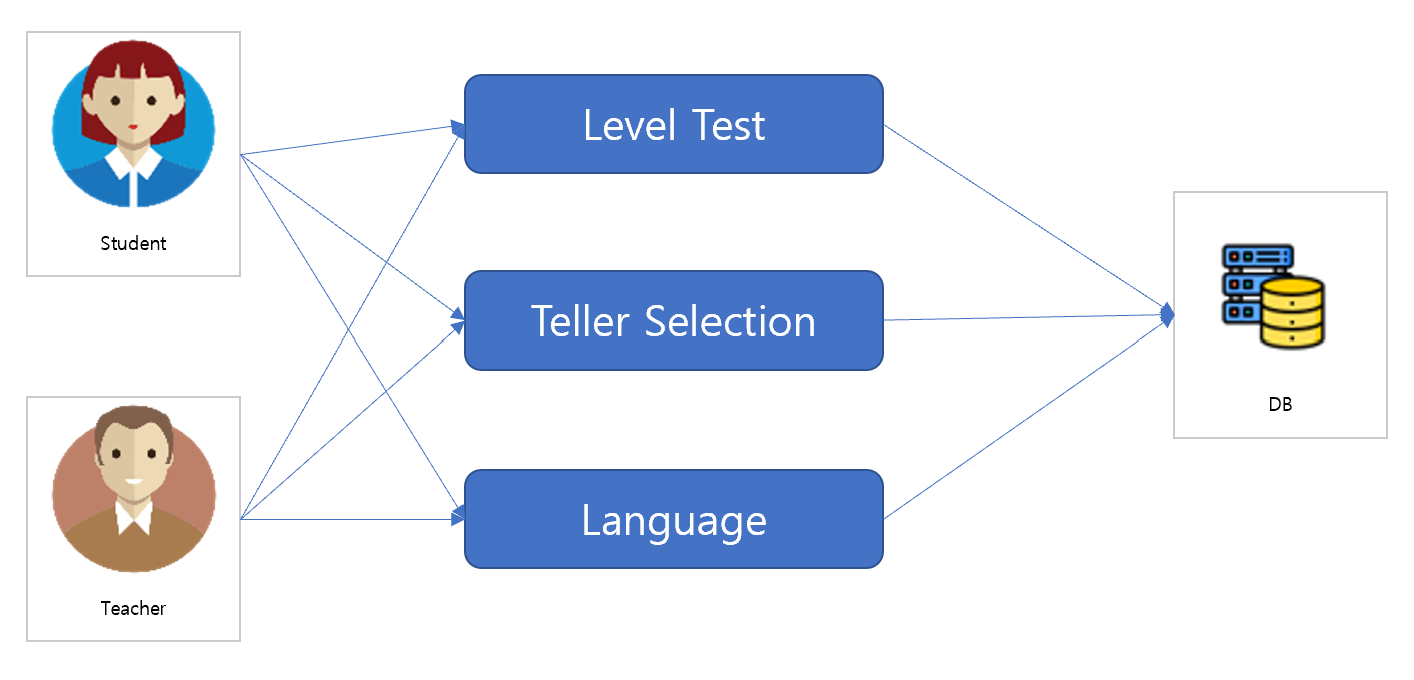
**Table 32. Use Case of Mode Select**

|  |  |
| --- | --- |
| Name | Mode Select |
| Actors | 학생, 선생님, 데이터베이스 |
| Description | 사용자(학생 및 선생님)이 로그인 이후 학생모드 혹은 선생님 모드로 진입하는 것을 의미한다. 선택에 따라 데이터베이스에서 사용자의 데이터를 불러온다. |
| Data | Input(User Selection) |
| Stimulus | 사용자가 모드선택에서 학생 혹은 선생님 버튼을 누른다. |
| Response | 사용자의 데이터를 로드한 뒤 해당 모드로 진입한다. |
| Comments | 사용자가 여러 명 로그인해도 각각의 데이터가 잘 불러와져야 한다. |

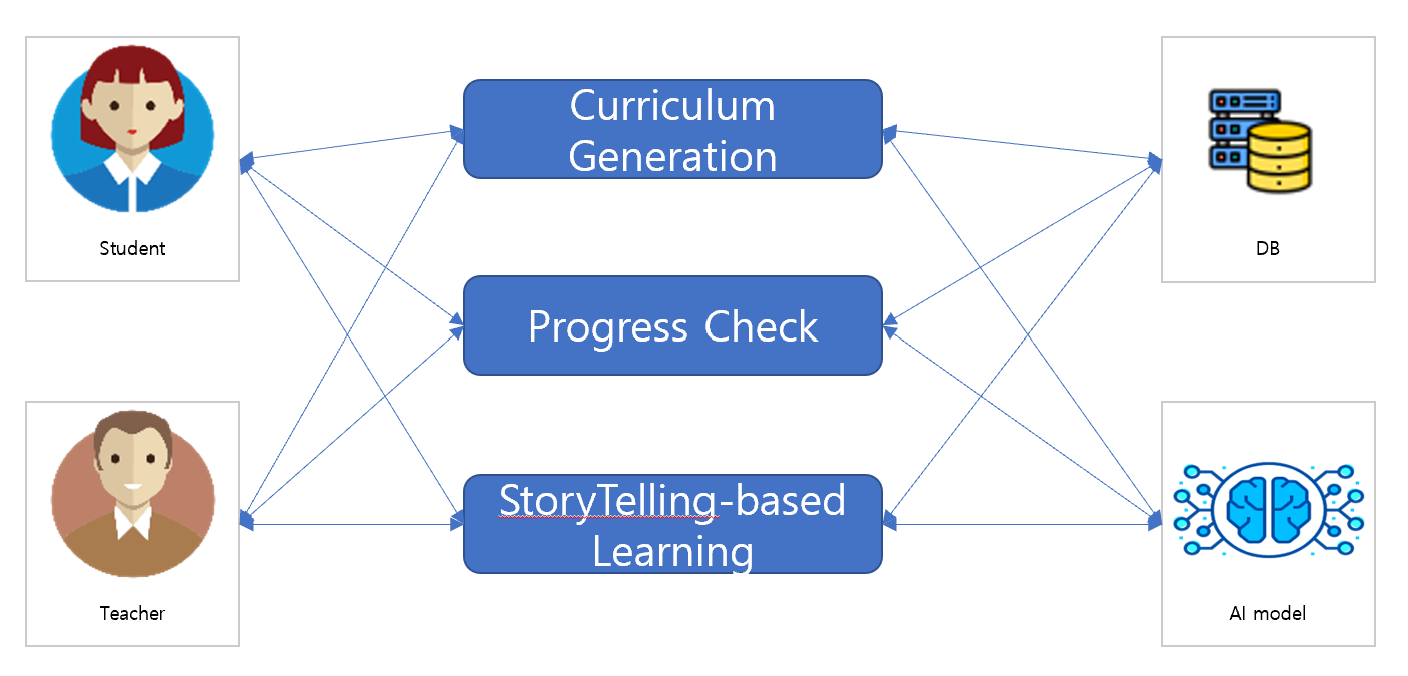
### **4.1.2. USE CASE DIAGRAM**

****

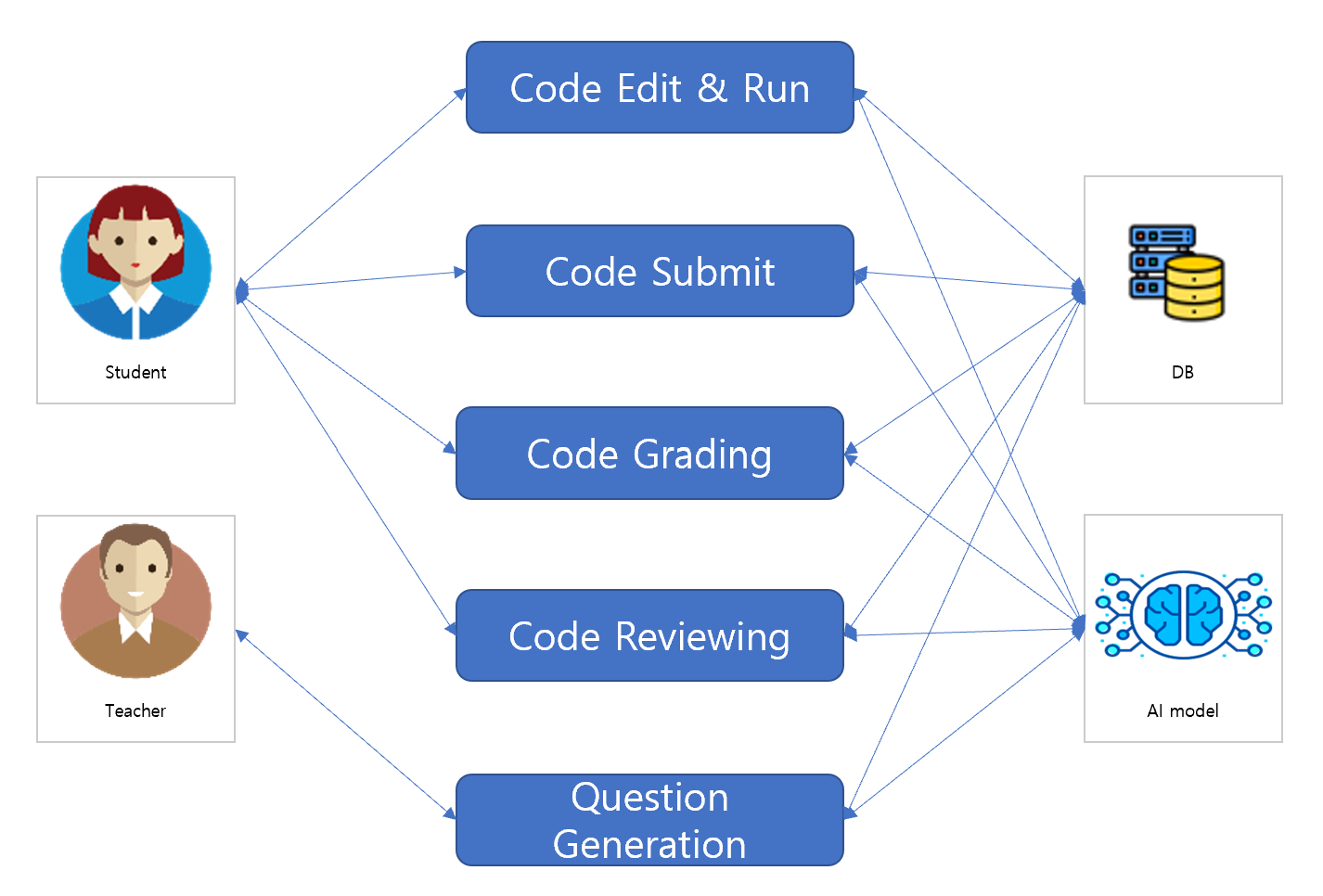
**Figure 2. Use Case Diagram - System**

****

**Figure 3. Use Case Diagram - Survey**

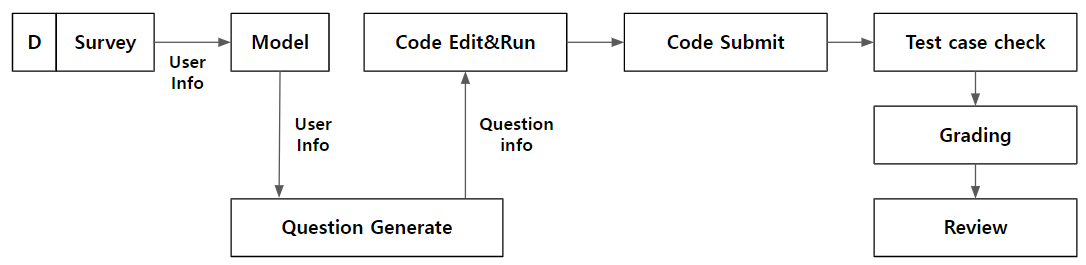
****

**Figure 4. Use Case Diagram - Learning**

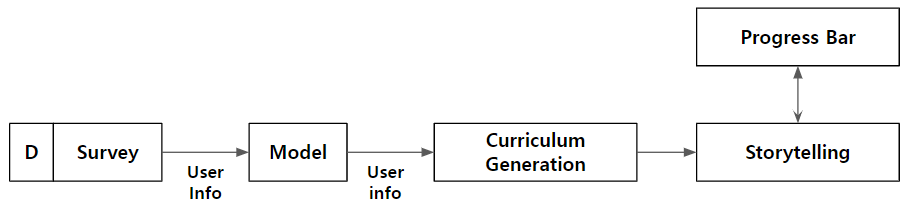
****

**Figure 5. Use Case Diagram - Question Solving**

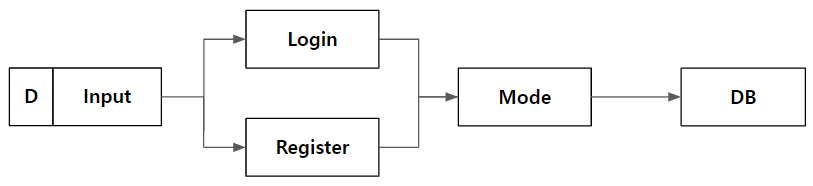
**4.1.3. DATA FLOW DIAGRAM**

****

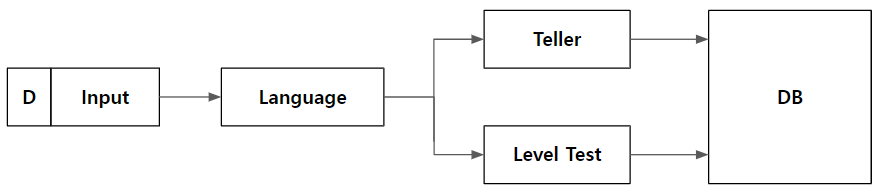
**Figure 6. Data Flow Diagram - Question Solving**

****

**Figure 7. Data Flow Diagram - Learning**

****

**Figure 8. Data Flow Diagram - System**

****

**Figure 9. Data Flow Diagram - Survey**

## **4.2. NON FUNCTIONAL REQUIREMENTS**

본 시스템이 제공하는 서비스 및 제약 조건을 서술한다.

### **4.2.1. PRODUCT REQUIREMENT**

소프트웨어의 작동과 제한을 명시하는 요구사항이다.

**A. USABILITY REQUIREMENT**

시스템은 코딩을 공부하는 학생의 입력된 정보에 맞는 커리큘럼을 제공해야 하며 스토리 텔러 기능을 통해 학생이 교육을 따라올 수 있도록 도와야 한다. (이후 사용에서의 지연시간)

**B. PERFORMANCE REQUIREMENT**

* 사용자와 AI 모델 간 주어진 정보에 따라 상호작용하는 과정.
* 테스트케이스를 검증하는 과정은 (3초) 이내로 완료되어야 한다.
* 코드를 채점하고 결과를 보여주는 과정은 (5초) 이내로 완료되어야 한다.
* 사용자 입력정보와 DB 간 연결

**C. DEPENDABILITY REQUIREMENT**

사용자가 입력 정보 및 그에 따라 AI 모델이 생성하는 정보는 DB에 저장되어야 하며 이는 사용자가 가시적으로 확인할 수 있어야 한다. Survey에 따른 사용자의 선택 사항에 맞게 알맞은 커리큘럼이 생성되어야 한다. 제출한 코드가 정확하게 채점되고 올바른 Test case를 통해 검증되어야 한다.

### **4.2.2. ORGANIZATIONAL REQUIREMENT**

고객 및 개발자 조직의 정책 및 절차의 결과로 발생하는 광범위한 요구사항이다.

**A. ENVIRONMENTAL REQUIREMENT**

사용자의 편리성과 개발환경을 고려하여 Window 및 Mac 등의 OS를 갖춘 PC로 접속할 것을 권장한다.

**B. DEVELOPMENT REQUIREMENT**

Front-End의 경우 React로 제작하며 Figma로 디자인을 관리한다. Back-End는 Django로 제작하며 AI 모델과 Firebase로 관리되는 DB를 포함한다다.

**C. OPERATIONAL REQUIREMENT**

해당 시스템은 코딩 교육을 하고자 하는 학생 또는 교육자의 입력 정보로 AI 모델이 생성하는 문제 해결 등을 통해 교육이 진행된다. 사용자는 학생 또는 교육자 모드를 선택하여 본 시스템을 사용할 수 있다.

학생은 survey를 통해 스토리 텔러와 사용하고자 하는 프로그래밍 언어를 선택할 수 있다. 교육자는 관련 카테고리를 통해 교육 목적의 학습지를 생성할 수 있다.

### **4.2.3. EXTERNAL REQUIREMENT**

시스템 및 개발 프로세스의 외부 요인에서 파생된 모든 요구사항을 다룬다. 여기에는 시스템의 규제 요구 사항이 포함된다.

**A. ETHICAL REQUIREMENT**

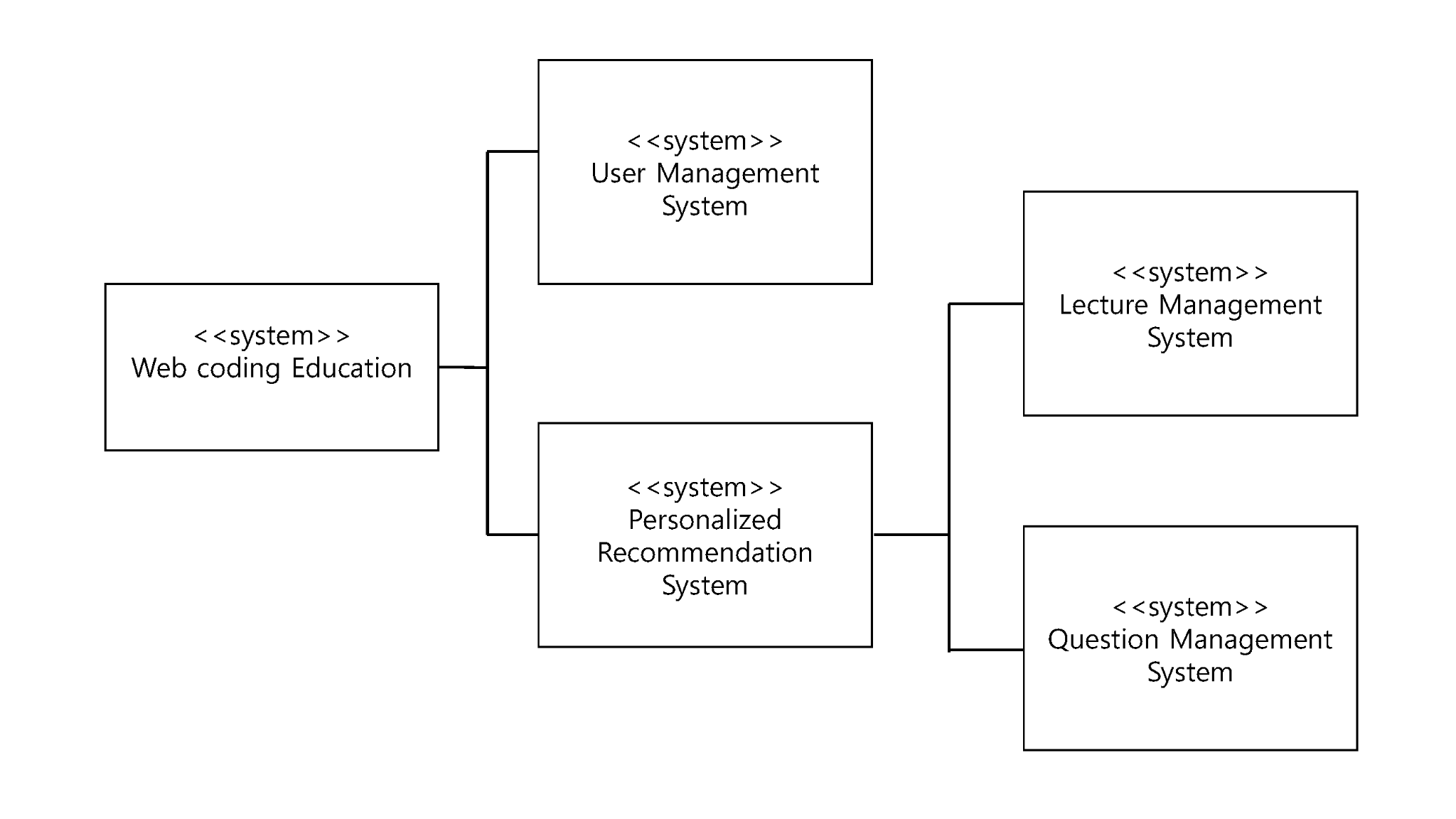
문제와 관련된 정보 및 사용하는 AI 모델 등의 출처를 정확히 명시하고 시스템에 사용되는 API의 라이센스를 검토해야 한다.

**B. SAFETY/SECURITY REQUIREMENT**

사용자가 작성한 코드 및 사용자의 입력 정보는 엄격하게 보호해야 한다.

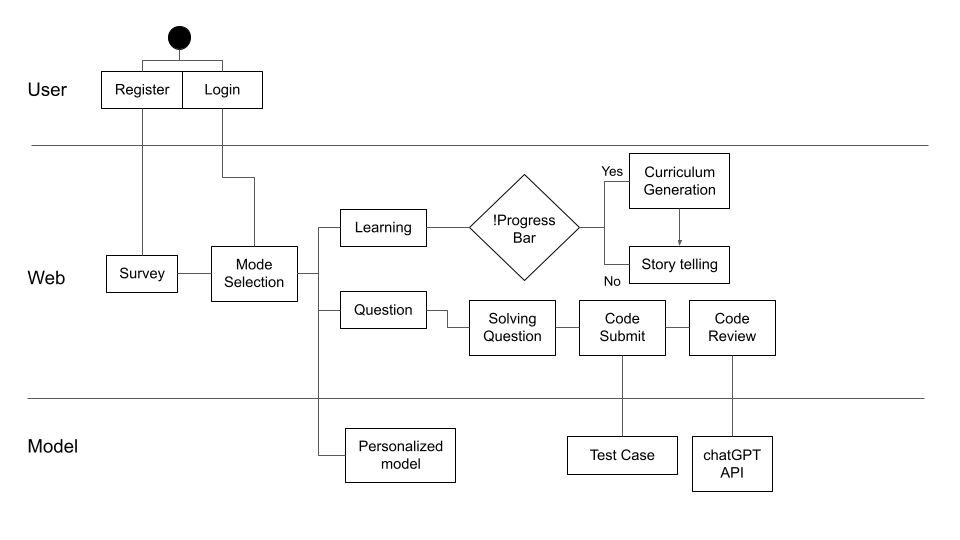
## **4.3. ORGANIZING SYSTEM FLOW**

**4.3.1 Context Model**

****

**Figure 10. Context Model**

**4.3.2 Process Model**

****

**Figure 11. Process Model**

**4.3.3 Interaction Model**

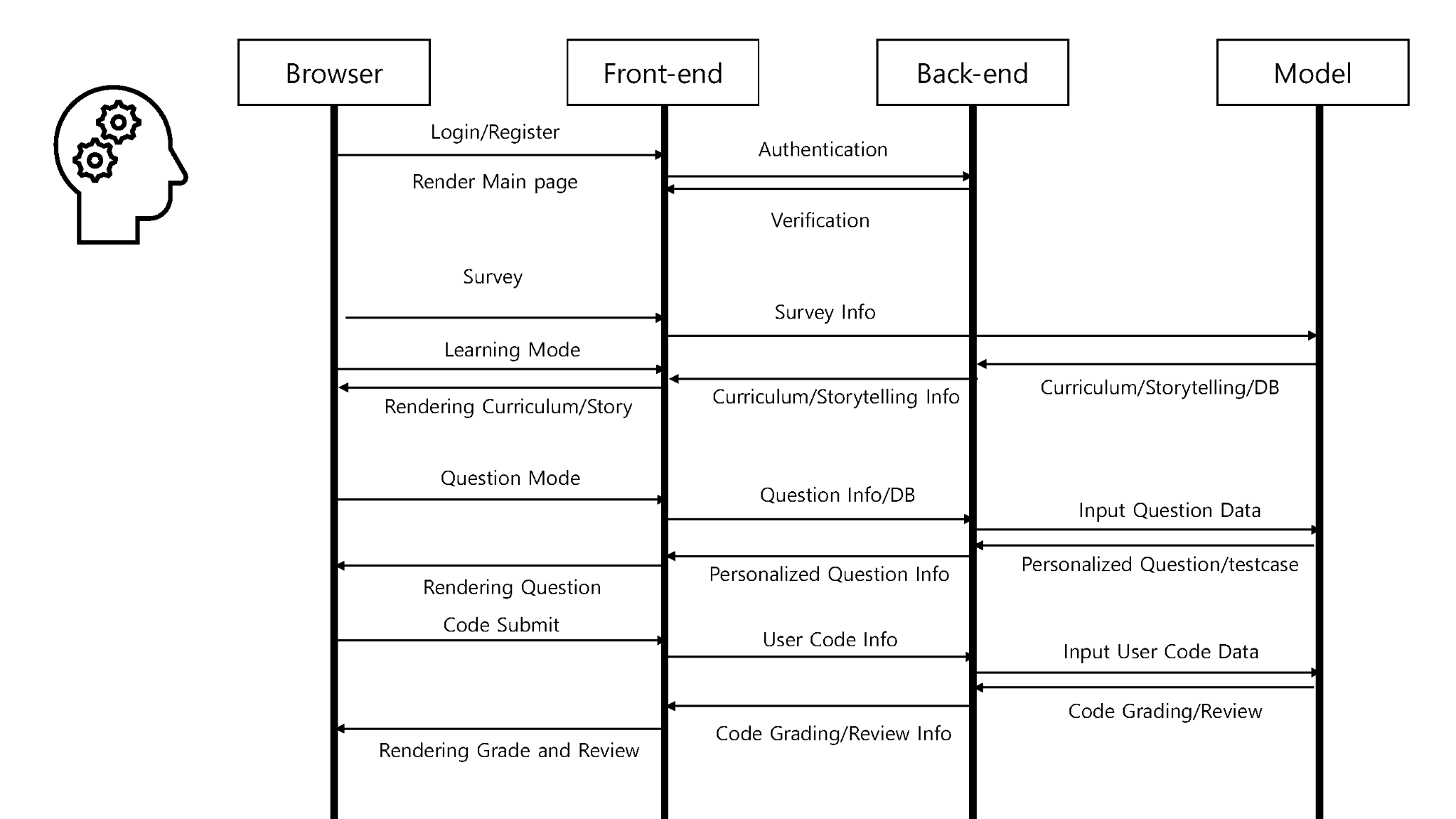
1. 1.2. Use Case Diagram 참고

**4.3.4 Behavior Model**

1. **Data Flow Diagram**

4.1.3. Data Flow Diagram 참고

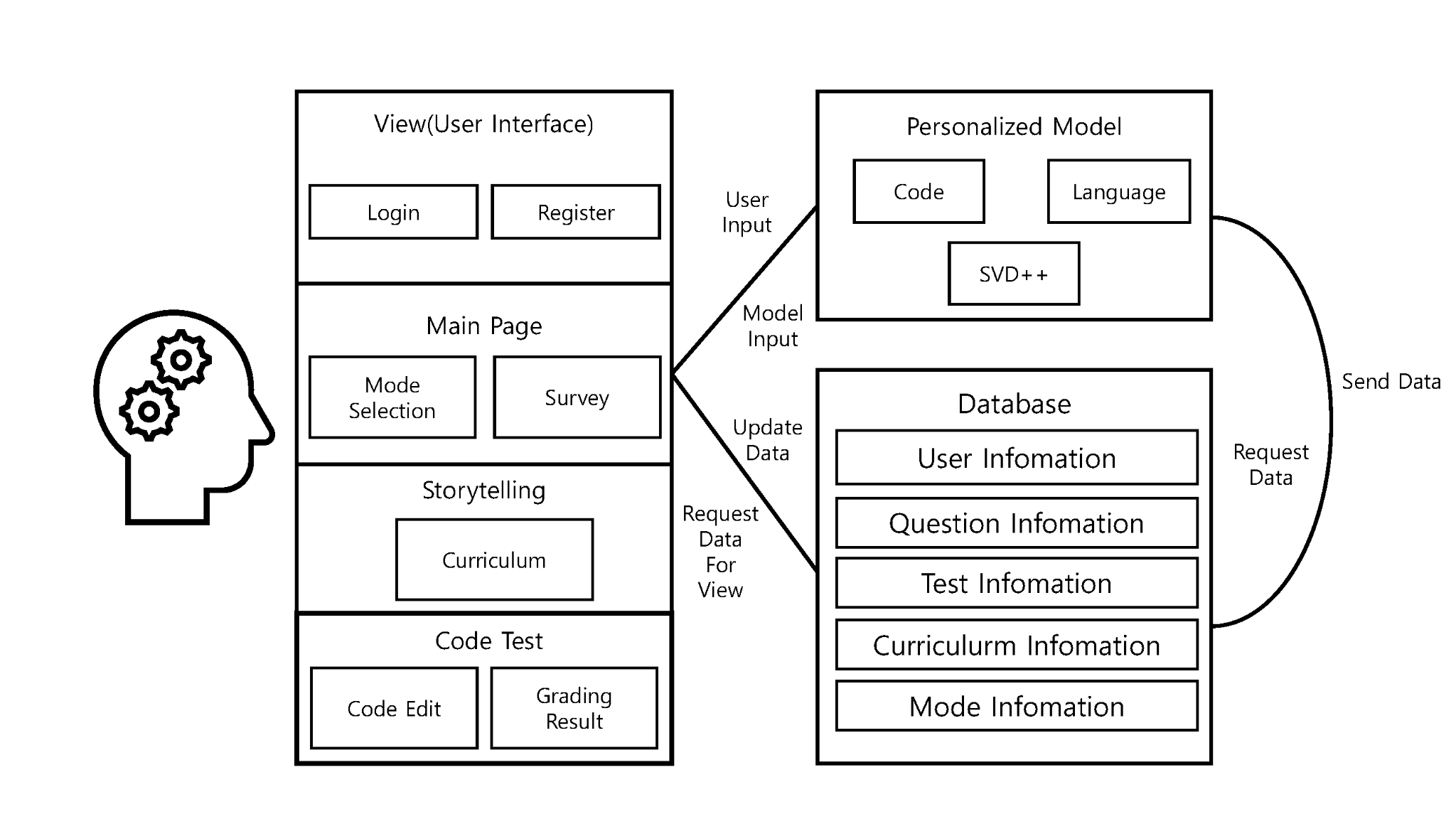
**B. Sequence Diagram**

****

**Figure 12. Sequence Model**

**4.4 System Architecture**

이 섹션에서는 예상되는 시스템 아키텍처에 대한 높은 수준의 개요를 제시한다.

**Figure 13. System Architecture**