

# Requirement Specification

Online Tutoring Web Application



**Team #3**

이름	학번
나종명	2014311352
박은찬	2017312838
임성규	2015310908
최아영	2015312791

# Table of Contents

1.	Preface .....	4
1.1	Intended Audience / Readership .....	4
1.2	Objective.....	4
2.	Introduction.....	5
2.1	Objective.....	5
2.2	System Overview .....	5
2.3	System Overview .....	7
2.4	Expected Effect .....	8
3.	Glossary.....	9
4.	User Requirements Definitions.....	10
4.1	Functional Requirements .....	10
4.2	Non-Functional Requirements .....	13
5.	System Architecture.....	15
5.1	Overall Architecture.....	15
5.2	Front-end Architecture.....	16
5.3	Back-end Architecture .....	16
5.4	Live-chatting Architecture .....	17
5.5	Screen-sharing Architecture .....	18
6.	System Requirements Specification .....	19
6.1	Functional Requirements .....	19
6.2	Non-Functional Requirements .....	37
7.	System Models .....	39
7.1	Objectives.....	39

7.2	Context Model.....	39
7.3	Process Model .....	39
7.4	Interactive Model .....	40
8.	System Evolution.....	54
8.1	Anticipated changes due to hardware evolution.....	55
8.2	Changing user needs.....	55
9.	Appendices.....	56
9.1	Hardware Requirements.....	56
9.2	Database Requirements.....	57
10.	Index.....	58
10.1	Table Index .....	58
10.2	Figure Index.....	59

# 1. Preface

## 1.1 Intended Audience / Readership

### A. Project Managers / System Developers

This document is mainly intended for people with knowledge in software systems, i.e. software engineers, designers, managers. System requirements are specifically mainly intended for developers for detailed specification for user requirements.

### B. Users

User requirements, intended for users and customers, explains the system requirements in natural language. It limits using professional jargon, and defines requirements easily with simplified explanation in natural language.

## 1.2 Objective

This document's objective is to provide overall guidance and specification of the requirements needed for the development of QAHub. The document includes the following:

- Explanation of the project goal (motivations, expected effect)
- User requirements definition
- System architecture and system requirements specification.
- System models and system evolution.

This document also serves as the foundation of our project, helping developers know the requirements for this project.

## 2. Introduction

### 2.1 Objective

이 부분에서는 QAHub를 개발의 동기가 된 학생들의 문제 해결에 대한 needs와 그에 대한 수요를 설명한다. 그리고 QAHub가 이 needs를 반영하여 어떻게 구현되었는지 대해 설명하고, 기존의 유사한 시스템들과는 어떻게 차별화되는지 설명한다.

### 2.2 System Overview

대학생과 과제는 서로 떨어질 수 없는 불가분의 관계에 있다. 과제는 학업 성취도를 평가하기 위한 수단이며, 학생들은 이를 수행함으로써 자신의 역량을 객관적인 기준으로 확인하고, 나아가 이를 발전시킬 계기를 얻게 된다.

문제는 여기에서부터 시작된다. 사람인 이상, 학생들은 여러가지 이유들로 인해 과제의 수행에 어려움을 겪게 되는 순간과 마주하게 된다. 이 어려움을 해결하기 위하여 학생들은 여러가지 수단을 통해 정보 획득을 모색한다. 특히 인터넷에서 획득할 수 있는 정보의 양은 막대하지만, 인터넷이 과제에 대한 궁극적인 해결책이었다면 이 문단이 작성될 이유도 없었을 것이다. 학생들이 마주하게 되는 난관은 크게 다음 2가지 정도로 나뉘어진다.

#### A. 관련된 정보 획득이 가능한가?

분명히 인터넷을 통해 방대한 양의 정보를 검색하여 획득할 수 있는 것은 사실이지만, 그 정보들 중에 우리가 원하는 정보가 있을지는 또 다른 문제이다. 많은 사람들이 다루지 않는 주제여서 관련된 정보 자체가 없을 수도 있고, 검색에 익숙하지 않아 잘못된 키워드들을 이용하여 정보 검색에 실패할 수도 있고, 획득에 성공한다 하여도 수많은 쓸모 없는 정보를 확인하는 과정에서 많은 시간이 소요될 수 있다.

#### B. 획득한 정보는 정확한 정보인가?

검색을 통해 정보를 얻었다고 할지라도, 아직 안심할 수 있는 단계에 도달하진 못했다. 획득한 정보의 작성자는 그 분야의 전문가일 수도 있지만, 막 그 분야에 입문한 초심자일 수도 있다. 설령 전문가가 작성한 정보라도, 그것이 100% 정확하다는 보장은 누구도 할 수 없다. 따라서 사람들은 필연적

으로 정보의 정확성을 검증하는 과정을 거치게 된다.

대부분의 과제는 수업에 사용되는 교재에 첨부된 연습 문제들을 해결하는 식으로 주어지기 때문에, 많은 학생들은 솔루션, 즉 교재의 해답지를 찾으려는 노력을 하게 된다. 이 과정에서 가장 많이 이용되는 플랫폼 중 하나는 Chegg이다.

← 체크	← chegg
<p>체크 팻 구하시는 분 연락주세요 저는 한 권이면 돼요 10/23 익명</p> <p>체크 같이 하실 분 있나요? 10/23 익명</p> <p>체크 2명 모집합니다. 체크하실분? 10/22 익명</p> <p>체크 한자리 남았는데 꺾사람있어?? 나 나오고 들어갈사람 10/19 익명</p> <p>체크 팻 더 모집하는 사람 있나요 10/18 익명</p> <p>체크솔루션 한권만살수있음? 넘비싼데.. 10/17 익명</p> <p>아 체크 결제해야되나 과제때에 시험공부들뚫아줄게 10/16 익명</p>	<p>chegg한문제만 알려주실분 키프리온 간단안거로 사려하겠습니다ㅠㅠ 제발좀 나눠주세요ㅠㅠ 10/20 익명</p> <p>chegg 꺼줄분 꼭주세요 10/15 익명</p> <p>chegg 쓰시는 분 탐승 가능할까요 사례 드립니다 10/14 익명</p> <p>cheggㅠㅠ체크 chegg 여러명어서 쓰고있는데 앱으로하든 컴퓨터로하든 동시접속... 10/12 익명</p> <p>chegg해제하는법 아는사람 ㅌㅌ 10/10 익명</p> <p>chegg 공유 동시접속만 걸리는거임 아니면 그냥 여러 아이디로 로그인하면 걸리... 10/02 익명</p> <p>chegg 하실분 세명에서 할 예정인데 꼭지나 댓글 주세요 10/02 익명</p>

Figure 1 커뮤니티 '에브리타임'에서의 Chegg에 대한 꾸준한 수요

Chegg는 교재 판매-대여로 사업을 시작했지만, 문제 솔루션 서비스를 제공하고 전문성 있는 튜터들을 섭외하여 1:1 튜터링을 중개하는 쪽으로 사업을 확대함으로써, 현재는 교육 분야의 일각을 차지한 회사이다. 앞서 말했던 대로 과제는 거의 교재에 적혀 있는 문제들을 해결하는 식이기 때문에, 학생들은 과제에 들이는 노력을 줄이기 위해 Chegg에서 해답지를 구매하거나, 질문을 업로드하고 답변을 기다리는 방식으로 서비스를 이용하게 된다.

Chegg를 이용하는 많은 학생들이 공통적으로 표하는 아쉬움이 몇 가지가 있는데, 하나는 한국어를 지원하지 않는다는 점에서 마주하는 언어의 장벽이고, 다른 하나는 혼자서 이용하기에는 부담되는 가격이다. 사이트나 애플리케이션이 한국어를 지원하지 않아 결제/환불에 어려움을 겪기도 하고, 혼자서 이용하기에는 부담되는 가격에 3~4명이 한 아이디를 돌려쓰는 방법을

택하기도 한다. 이런 문제점들과 앞서 언급한 두 가지 장애물들에 착안하여, 우리말로 된 질문자-답변자들의 웹 기반 플랫폼을 개발의 필요성을 느끼게 되었다.

### 2.3 System Overview

QAHub는 프로그래밍, 수학 등 여러 가지 분야의 문제 해결을 원하는 멘티와 이 문제들에 대해 해답을 제공할 수 있는 전문성을 가진 멘토를 중개해 주는 웹 서비스이다.

멘티들은 서비스를 이용하기 위하여 포인트를 결제하고, 포인트를 소모함으로써 질문을 등록한다. 답변 방법은 텍스트/일반 라이브 채팅/화면 공유를 포함한 라이브 채팅으로 나뉘며, 멘티는 원하는 답변 방법을 선택할 수 있다. 답변을 제공 받았다면, 멘티들은 이에 대한 평점을 매김으로써 그 답변이 원하는 답변이었는지, 답변이 정확했는지에 대한 평가를 할 수 있다.

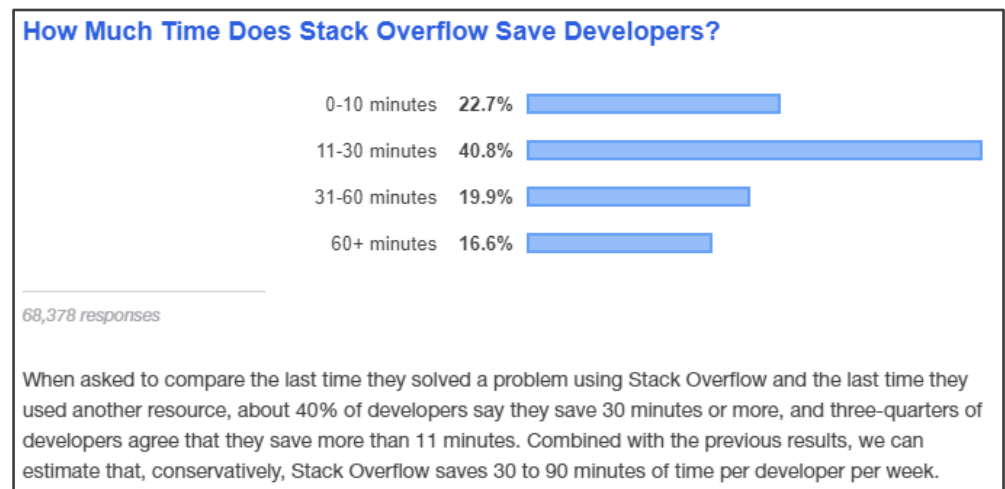
자신이 해답을 원하는 문제가 이미 등록되어 있고, 해답이 제공되었는지를 알아보기 위하여 멘티는 문제를 여러 가지 키워드들을 통하여 검색할 수 있고, 답변이 등록된 문제라면 일정량의 포인트를 소모함으로써 등록된 답변을 열람할 수 있다. 답변을 열람하기 전에, 평점을 확인함으로써 열람할지에 대한 여부를 고민할 수 있다.

멘토들은 관심 분야를 선택하여 해당 분야에서 답변을 기다리는 질문들의 리스트를 확인할 수 있으며, 최근 등록/조회수/답변 방법 등의 여러가지 필터들을 적용하여 질문들의 범위를 좁힐 수 있다. 답변을 제공하고 멘티가 평점을 등록한 이후에 멘토는 그 답변에 대한 포인트를 획득할 수 있다.

## 2.4 Expected Effect

### A. 멘티에게 보다 저렴한 가격으로 서비스 제공

QAHub를 이용하는 멘티들은 Chegg의 이용료보다 상대적으로 적은 금액을 지불함으로써 전문성 있는 멘토와의 1:1 멘토링을 통해 원하는 문제에 대한 해답을 얻을 수 있을 것이다. 또한, 정형화된 플랫폼을 이용함으로써 해답을 얻는 데 들이는 시간을 어느 정도 단축할 수 있게 될 것이다. 다음은 2019년 Stack Overflow의 설문조사 중 일부를 발췌한 것이다.



**Figure 2 StackOverflow를 사용한 시간 절약 효과 설문(2019)**

75% 이상의 응답자들이 11분 이상의 시간을 절약할 수 있었다고 답했다  
(응답자 68,378명)

QAHub가 성공적으로 교육 플랫폼의 입지를 다질 수 있게 된다면, QAHub를 이용하는 멘티들에게 소모한 시간 대비 정확한 답변을 제공할 수 있게 될 것이다.

### B. 멘토에게 성취감 및 전문성을 입증할 수 있는 증빙자료 제공

멘토들은 질문에 답변을 제공함으로써 포인트를 획득하여 자신의 회원 등급을 올리고, 포인트를 현금으로 전환하여 실제 수익을 기대할 수 있게 될 것이다. 덤으로, 멘티들을 도와줬다는 사실에서 오는 심리적인 만족감도 기대할 수 있게 될 것이다. 또한 답변의 채택율이나 회원 등급 등을 이용하여 구직 활동, 면접 등에서 자신의 전문성을 강조할 수 있는 계기를 마련할 수 있게 될 것이다.



### 3. Glossary

용어	뜻
라이브 채팅 (Live Chatting)	사용자들이 다른 사용자들이나 사이트의 관리자들과 실시간으로 메시지를 주고받을 수 있게 돕는 서비스
멘토링 (Mentoring)	풍부한 경험과 지혜를 겸비한 신뢰할 수 있는 사람이 1:1로 지도와 조언을 하는 것
멘토 (Mentor)	멘토링에서 조력자의 역할을 하는 사람
멘티 (Mentee)	멘토링에서 조력을 받는 사람
Authentication	등록을 마친 사용자가 시스템을 이용하기 위해 등록한 정보로 시스템에 자신의 신원을 인증하는 과정
Available Time	멘토가 주로 활동하며 실시간으로 답변을 제공할 수 있는 시간
Credit	본 시스템을 이용하기 위해 요구되어지는 포인트의 일종
DB	Database의 약자. 여러 사람이 공유하여 사용할 목적으로 체계화해 통합, 관리하는 데이터의 집합
Field	흥미를 가지고 있는, 또는 전문성을 가지고 있는 분야, 또는 과목
QAHub	본 문서에 설명된 서비스들을 제공하는 시스템의 이름
Registration	사용자가 시스템을 이용하기 위해 자신의 정보를 시스템에 등록하는 과정
Stack Overflow	프로그래밍에 관한 질문과 답변을 주고받는 개발자 커뮤니티
Verification	멘티가 멘토의 답변이 얼마나 유익했는지를 평가하는 행위

Table 1 Glossary

## 4. User Requirements Definitions

### 4.1 Functional Requirements

#### A. 유저 등록

- a. 신규 유저가 시스템에 등록을 할 수 있도록 한다.
  - i. ID 중복확인을 해야만 등록할 수 있어야 한다.
  - ii. 비밀번호는 최소 8자리 이상이며 소문자 알파벳, 대문자 알파벳, 그리고 숫자를 포함하여야 한다.
  - iii. 추가로 이름, 별명 (nickname), 그리고 이메일을 입력한다.
- b. 가입 시 멘토 혹은 멘티로 등록하도록 한다.
  - i. 이때, 멘토로 등록 시 본인의 필드와 해당 필드에서의 전문성을 입증할 수 있는 자료를 첨부한다. 또한, 실시간으로 활동하는 available time을 설정한다.
  - ii. 멘티는 관심있는 필드를 선택할 수 있도록 한다.

#### B. 유저 인증

- a. 유저는 등록한 ID 와 Password 로 시스템에 인증을 할 수 있어야 한다.
- b. ID 와 Password 를 잊어버렸을 시에 등록된 기본정보로 찾을 수 있어야 한다.
  - i. ID 는 이름과 이메일을 입력하면 웹사이트 상에서 아이디를 보여준다.
  - ii. Password 는 아이디, 이름, 이메일을 입력한 후 이메일로 임시 비밀번호를 발급하여 보내준다.
- c. 여러 기기에서 인증을 했을 경우 가장 최신 인증한 기기에서만 인증이 유효하도록 설정한다.

#### C. 유저 정보 열람 및 수정

- a. 유저는 인증이 된 상태에서 본인이 등록한 정보를 열람 및 수정 할 수

있어야 한다.

- i. 이때 본인의 비밀번호를 한번 더 확인한다.
  - ii. 단, 아이디와 type (멘토 혹은 멘티 여부) 는 변경이 불가능 하도록 설정한다.
- b. 유저는 본인이 등록한 문제 혹은 답변을 한 문제를 조회할 수 있어야 한다.

D. 유저 등급

- a. 멘티의 경우 검증된 답변이 달린 질문의 개수로 등급을 부여한다.
- i. 1 - 10 일반 멘티
  - ii. 11 - 50 열혈 멘티
  - iii. 51 - 100 베스트 멘티
  - iv. 100 + 플래티넘 멘티
- b. 멘토에 경우 검증된 답변의 개수로 등급을 부여한다.
- i. 1 - 10 일반 멘토
  - ii. 11 - 50 열혈 멘토
  - iii. 51 - 100 베스트 멘토
  - iv. 100 + 플래티넘 멘토
- c. 유저의 등급은 회원이 로그인 하였을 시 보이도록 한다.

E. 포인트 결제 / 환급

- a. 유저는 본인의 결제수단을 이용하여 서비스에서 이용하기 위한 포인트를 구매할 수 있게 한다.
- b. 포인트는 원하는 계좌로 환급받을 수 있도록 한다. 이때 수수료가 발생 하는 것을 안내한다.
- i. 수수료는 유저의 등급에 따라 부여한다.

- ii. 실제 환급 받는 금액과 수수료를 고지한다.

#### F. 문제 등록

- a. 멘티가 문제를 작성하고 보상 크레딧의 값을 정할 수 있는 기능을 포함한다.
- b. 필수 기입: 제목, 내용, 관련 분야, 원하는 답변 방법, 보상 포인트가 있다.
  - i. 원하는 답변 방법은 (1) 텍스트 설명, (2) 일반 라이브 채팅, (3) 화면 공유를 포함하는 라이브 채팅이 있다.
  - ii. 이 때 보상 포인트는 기본값은 선택 답변 방법에 따라 결정되며, 이보다 적은 값을 입력할 수 없다. 또한 100원 단위로 입력할 수 있고 자신의 보유 포인트를 초과할 수 없다.
  - iii. 답변 종류에 따른 보상 포인트 기본값
    - (1) 텍스트 설명: 500
    - (2) 일반 라이브 채팅: 2000
    - (3) 화면 공유를 포함하는 라이브 채팅: 3000
- c. 선택 기입: 태그

#### G. 문제 답변

- a. 멘토는 멘티가 등록한 문제에 대한 답변을 할 수 있다. 이때 멘티가 선택한 방법을 통해 답변을 한다.
  - i. 라이브 채팅 혹은 화면 공유 같은 경우 본인의 시간대에 맞는 질문이 있을 경우 분리되어 우선적으로 보여지며 목록에서 선택하여 예약할 수 있다. 이 경우, 예약된 시간대에 접속하여 채팅 혹은 화면 공유를 통해 답변할 수 있다.
  - ii. 텍스트 설명같은 경우 시간제약이 없지만 이미 답변이 있을 경우 답변을 등록할 수 없다.
- b. 질문을 등록한 멘티가 답변 내용을 검증한 후 멘토가 답변에 걸린 보상

을 획득할 수 있다.

- i. 이때, 멘티는 멘토에게 등록된 포인트보다 많은 포인트를 지급할 수 있다.
- ii. 답변이 된 후 3일 이내에 답변 내용을 검증하지 않을 경우 자동으로 등록된 포인트가 지급되도록 한다.
- iii. 답변에 문제가 있을 경우 운영진 측에서 확인하여 포인트 환급 등 적절한 조치를 취할 수 있어야 한다.

#### H. 문제 보기

- a. 관심 분야를 선택하여 해당 분야로 업로드된 문제 리스트를 확인할 수 있다. 이 때 등록순/조회수/선호도/답변 방법 등으로 필터링하거나 정렬할 수 있다.
- b. 리스트의 항목을 클릭하여 문제를 볼 수 있으며, 이미 답변이 된 문제인 경우 해당 문제의 답변 방법이 텍스트일 경우에만 일정 포인트를 차감하여 답변을 확인할 수 있다.
- c. 나의 문제 보기에서 본인이 등록한 문제 / 답변을 볼 수 있다.

### 4.2 Non-Functional Requirements

#### A. Usability

- a. Chrome 브라우저에서 정상적으로 동작하여야 한다.
- b. 멘토와 멘티가 문제를 확인하는데 어려움이 없도록 직관적인 UI 및 UX를 제공한다.

#### B. Performance

- a. 라이브 채팅 및 화면 공유가 정상적인 인터넷 환경에서 끊김없이 동작하여야 한다.
- b. 모든 서비스의 접속이 3초 이내로 되도록 한다.

#### C. Dependability

- a. 서버 failure 시 자동으로 복구하도록 한다.

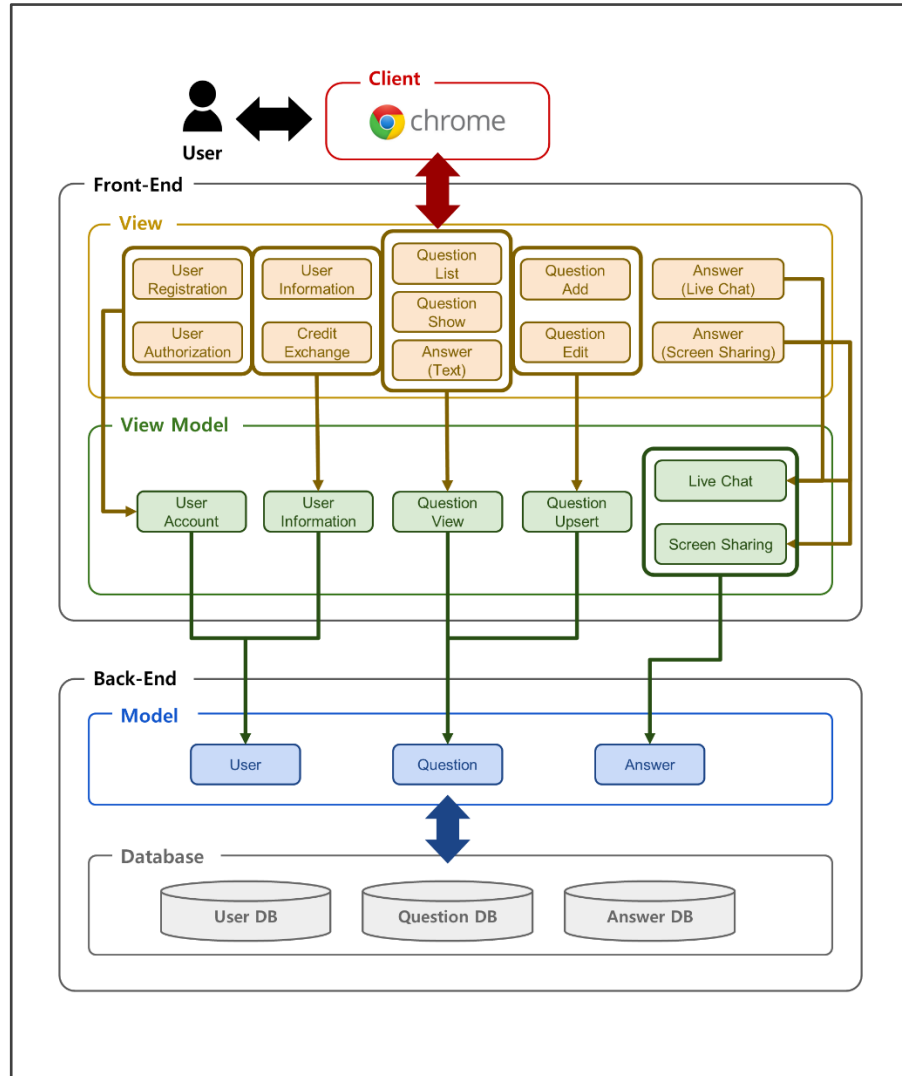
- b. 유저의 데이터가 정상적으로 저장되고 주기적으로 백업되어야 한다.
- c. 사용하는 라이브러리, 프레임워크, 혹은 브라우저의 업데이트가 발생하여도 정상적으로 작동하여야 한다.

D. Security

- a. 인증받지 않은 유저에 대해서 서비스의 이용이 차단되어야 한다.
- b. 포인트를 사용하지 않은 유저가 포인트 사용이 필요한 항목에 대해 접근이 불가해야 한다.
- c. 비밀번호가 암호화 되어 저장되어야 한다.
- d. 개인정보가 제 3자에게 노출되지 않아야 한다.

## 5. System Architecture

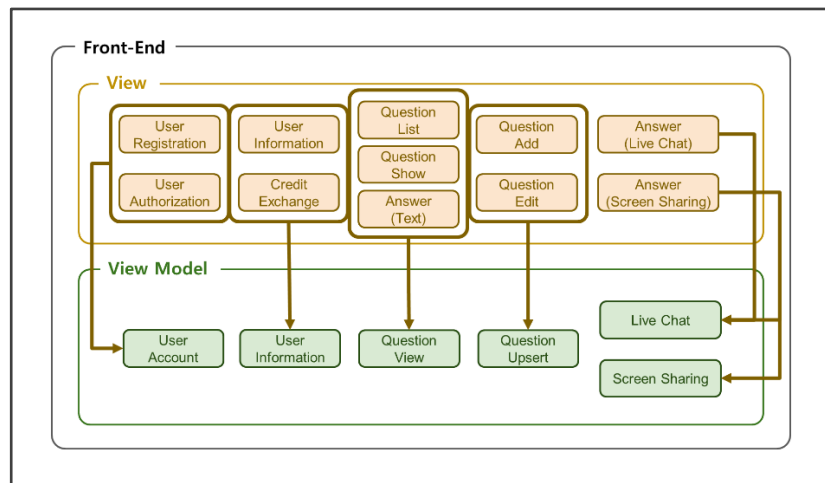
### 5.1 Overall Architecture



**Figure 3 Overall Architecture**

QAHub의 전체 아키텍처는 크게 Front-end, Back-end, Live-chatting System, Screen-sharing System으로 나뉘어져 있다.

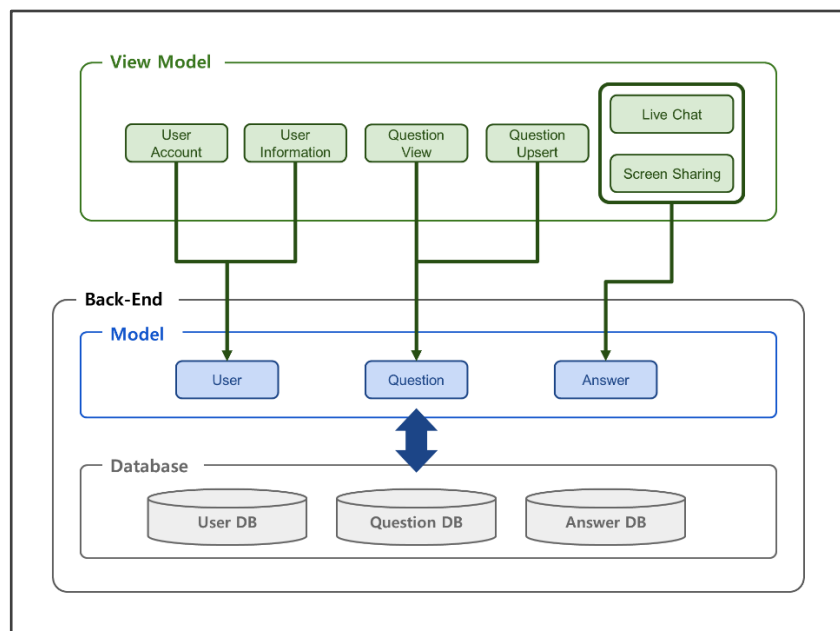
## 5.2 Front-end Architecture



**Figure 4 Front-end Architecture**

사용자와 직접적으로 상호작용하는 Front-end 아키텍처이다. 11개의 View와 6개의 View Model로 이루어져 있으며, vue.js를 활용할 것이다.

## 5.3 Back-end Architecture



**Figure 5 Back-end Architecture**

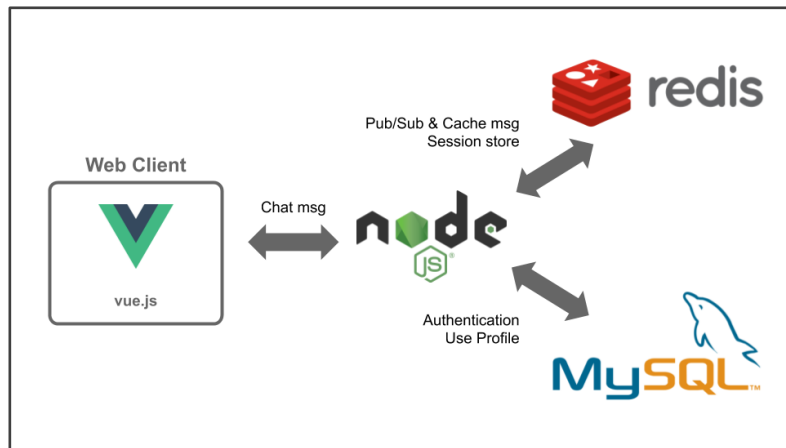
프론트에서 보내는 REST API 요청을 핸들링 하고 데이터베이스와 통신하는



Back-end 아키텍처이다. 모델은 크게 User, Question, Answer 로 나뉘며 각각 해당 데이터 베이스 엔티티와 연결이 되어 Query를 보내게 된다. Node.js 로 구현이 될 예정이다.

#### 5.4 Live-chatting Architecture

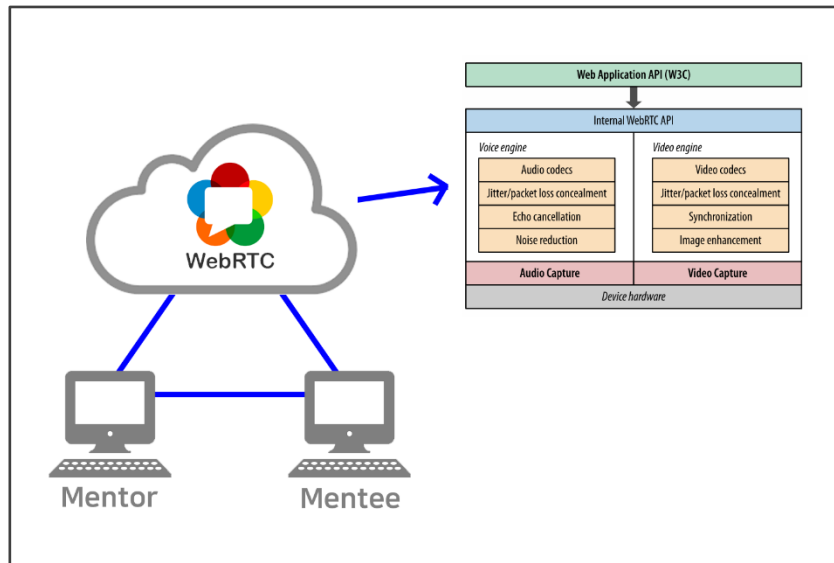
dd



**Figure 6 Live-chatting Architecture**

실시간 채팅을 위한 아키텍처로 vue.js와 socket.io를 이용하여 node.js 웹 서버와 통신하며 이브러리로 구성되어 있다. 웹 서버에서 요청한 유저에 대한 정보를 MySQL 데이터베이스에서 확인하고 redis를 이용해 채팅 내역 저장과 메시지 캐싱을 하는 구조이다.

## 5.5 Screen-sharing Architecture



**Figure 7 Screen-sharing Architecture**

화면 공유를 위한 아키텍처로 각 사용자가 브라우저에서 WebRTC API를 통해 통신하게 된다. 이때 WebRTC API에서는 유저의 디바이스로부터 오디오 및 화면을 캡처 하여 상대방에게 전달하게 된다.

## 6. System Requirements Specification

### 6.1 Functional Requirements

#### A. User Registration

Function	Register user to system		
Description	Registers the user to the system with corresponding user information. Required for accessing services provided by the system.		
Source	User input		
Inputs	<u>id</u> (String)	personal identification unique to each user	
	<u>password</u> (String)	user password	
	<u>name</u> (String)	name of the user	
	nickname (String)	nickname of the user, defaults to name if not given	
	<u>email</u> (String)	email address of the user, used for id / password recovery	
	<u>type</u> (String)	specify whether user is mentor or mentee	
	field(s) (String)	fields of interest (mentee) or fields of expertise (mentor)	
	available_time(s) (day of week, time)	(mentor only) available time for live help	
Outputs	return true on success, return false with error message on failure	Destination	User browser

<b>Action</b>	Search user table in the database when user checks if the ID exists in the system. Apply hashing to password. Combine user input and store in database. Redirect to login page on success. Display error message on error.
<b>Requirements</b>	<ul style="list-style-type: none"> <li>• ID must be unique (check if ID exists in the system when button is clicked)</li> <li>• All underlined fields are required</li> <li>• ID must consist of alphabet letters or numbers.</li> <li>• email must not be empty and written in specific format.</li> <li>• Password must be encrypted using hash function (SHA256)</li> <li>• Password must contain at least 8 characters including upper, lowercase letters and numbers.</li> <li>• Must make input for password confirmation and make sure they match.</li> </ul>
<b>Pre-condition</b>	User is not registered in the system and cannot use services provided by the system.
<b>Post-condition</b>	User is registered in the system and can have access to all the services provided by the system when authorized.
<b>Side effects</b>	None

**Table 2 Functional Requirement: User Registration**

## B. User Authentication

<b>Function</b>	Authenticate user		
<b>Description</b>	Authenticate user with user id and password (login). Required for accessing services provided by the system.		
<b>Source</b>	User input		
<b>Inputs</b>	<u>id</u> (String)	registered id	
	<u>password</u> (String)	corresponding user password	
<b>Outputs</b>	return true on success, return false on failure	<b>Destination</b>	User browser
<b>Action</b>	Search user table in the database with entered id. If id is exist, check whether the password in the database is equal to the hashed value of entered password. If it is, create user session and redirect to main page. If id doesn't exist or the passwords do not match, return false and alert that login was unsuccessful.		
<b>Requirements</b>	<ul style="list-style-type: none"> <li>inputs of id and password must only consist of alphabet letters or numbers</li> <li>id or password cannot be empty</li> <li>On failure, it should display "ID or password is incorrect"</li> <li>If a session with the id already exists, delete the one that exists and create new session (keep only the most recently logged in device logged in)</li> <li>All underlined fields are required</li> </ul>		
<b>Pre-condition</b>	User is not authorized to access the services provided by the system.		
<b>Post-condition</b>	User is authorized to access the services provided by the system.		
<b>Side effects</b>	None		

**Table 3 Functional Requirement: User Authentication**

C. ID Recovery

<b>Function</b>	Recover user id		
<b>Description</b>	Allow a user to recover his/her id when he/she forgets what it is		
<b>Source</b>	User input		
<b>Inputs</b>	<u>name</u> (String)	registered name	
	<u>email</u> (String)	registered email	
<b>Outputs</b>	registered ID on success, error message on failure	<b>Destination</b>	User browser
<b>Action</b>	Search user table in the database with the entered name and email. If the user exists, display the registered ID. If it does not exist, display failure message		
<b>Requirements</b>	<ul style="list-style-type: none"> <li>All underlined fields are required</li> </ul>		
<b>Pre-condition</b>	A registered user does not know his/her id		
<b>Post-condition</b>	The user recovers his/her forgotten id		
<b>Side effects</b>	If one's name and email is known, his/her id can be fetched		

**Table 4 Functional Requirement: ID Recovery**

#### D. Password Recovery

<b>Function</b>	Recover user password		
<b>Description</b>	Allow a user to recover his/her password when he/she forgets what it is		
<b>Source</b>	User input		
<b>Inputs</b>	<u>id</u> (String)	registered ID	
	<u>name</u> (String)	registered name	
	<u>email</u> (String)	registered email	
<b>Outputs</b>	return true on success, error message on failure	<b>Destination</b>	User browser
<b>Action</b>	Search user table in the database with the entered id, name and email. If they all match, send a temporary password to the email. The email should encourage the user to change the password after logging in with the temporary password. If any of the information do not match, display error message.		
<b>Requirements</b>	<ul style="list-style-type: none"> <li>• All underlined fields are required</li> <li>• Email address must be valid</li> </ul>		
<b>Pre-condition</b>	A registered user does not know his/her password and cannot login		
<b>Post-condition</b>	The user gets a temporary password for authentication.		
<b>Side effects</b>	If one's name and email is known and the email address is vulnerable, the user can be hacked.		

**Table 5 Functional Requirement: Password Recovery**

#### E. User Information Modification

<b>Function</b>	Modify user information		
<b>Description</b>	Allow user to view their registered information and modify them		
<b>Source</b>	User input		
<b>Inputs</b>	password (String)	user password	
	name (String)	name of the user	
	nickname (String)	nickname of the user, defaults to name if not given	
	email (String)	email address of the user, used for id / password recovery	
	field(s) (String)	fields of interest (mentee) or fields of expertise (mentor)	
	available_time(s) (day of week, time)	(mentor only) available time for live help	
<b>Outputs</b>	return true on success, return false with error message on failure	<b>Destination</b>	User browser
<b>Action</b>	When submitted, update user table in the database with the newly entered user information. If successful, alert successful message and refresh page. If unsuccessful, display error message.		
<b>Requirements</b>	<ul style="list-style-type: none"> <li>• Must be logged in</li> <li>• Must check user's password one more time before entering this page</li> <li>• Each input except for password must be pre-filled with the registered information when the user enters the page to modify their information (get information from user session)</li> <li>• User cannot change their id or type (mentor or mentee)</li> </ul>		



	<ul style="list-style-type: none"> <li>• Password must be encrypted using hash function (SHA256)</li> <li>• Password must contain at least 8 characters including upper, lowercase letters and numbers.</li> <li>• Must make input for password confirmation and make sure they match.</li> </ul>
<b>Pre-condition</b>	-
<b>Post-condition</b>	User information is updated
<b>Side effects</b>	None

**Table 6 Functional Requirement: User Information Modification**

## F. Purchase Credit

<b>Function</b>	Purchase credits		
<b>Description</b>	Allow user to purchase credits that can be used on the system		
<b>Source</b>	User input		
<b>Inputs</b>	<u>amount</u> (int)	amount of credits to purchase	
	<u>pay_method</u> (select)	method to pay for the credits	
<b>Outputs</b>	true on success, error message on failure	<b>Destination</b>	User browser
<b>Action</b>	When user submits the request to purchase credits, request transaction with the total price to the corresponding service. When transaction is confirmed and payment is completed, give corresponding credit to user in the database.		
<b>Requirements</b>	<ul style="list-style-type: none"> <li>• Must be logged in</li> <li>• Display the amount of credit to be purchased and the total price</li> <li>• When payment transaction fails, the credit should not be given to the user</li> <li>• Must confirm user of submitting their input</li> <li>• All underlined fields are required</li> </ul>		
<b>Pre-condition</b>	-		
<b>Post-condition</b>	purchased credits are given to user		
<b>Side effects</b>	If the selected payment method is not available (ex. bank server maintenance), it becomes impossible to purchase credit.		

**Table 7 Functional Requirement: Purchase Credit**

## G. Exchange Credit to Currency

<b>Function</b>	Exchange credits to currency		
<b>Description</b>	Allow users to exchange their credits back to currency		
<b>Source</b>	User input		
<b>Inputs</b>	<u>amount</u> (int)	amount of credits to exchange, a multiple of 1000	
	<u>bank</u> (String)	bank of the account used for exchange	
	<u>account_number</u> (number)	account number for receiving currency	
<b>Outputs</b>	true on success, error message on failure	<b>Destination</b>	User browser
<b>Action</b>	When user submits the request to exchange credits, request transaction to transfer the total calculated amount of currency to the entered account. When transaction is confirmed, subtract the amount of credit from the user in the database.		
<b>Requirements</b>	<ul style="list-style-type: none"> <li>• Must be logged in</li> <li>• Must display the amount of credits left and the total currency to be exchanged</li> <li>• When payment transaction fails, the credit should not be taken away from the user</li> <li>• Amount of credit must be a multiple of 1000</li> <li>• Amount of currency = amount of credit * (1 - commission_rate) where commission_rate = 0.15 for rank 1, 0.10 for rank 2, 0.05 for rank 3</li> <li>• Must display the commission rate and amount of currency taken by commission</li> <li>• amount to be exchanged should be less than or equal to the amount of credits the user has</li> <li>• Must confirm user of submitting their input</li> </ul>		

	<ul style="list-style-type: none"> <li>All underlined fields are required</li> </ul>
<b>Pre-condition</b>	user owns credits that can be used by the system
<b>Post-condition</b>	user's credits are exchanged to currency
<b>Side effects</b>	If the transaction fails (ex. bank server maintenance), the user cannot exchange credit to currency.

**Table 8 Functional Requirement: Exchange Credit to Currency**

## H. Question Registration

<b>Function</b>	Register question		
<b>Description</b>	Allow a mentee to register a question that he/she needs help with.		
<b>Source</b>	User input		
<b>Inputs</b>	<u>title</u> (String)	title of question	
	<u>content</u> (String)	content of the question	
	<u>reward</u> (int)	amount of credit given as reward to the mentor who answers this question	
	<u>type</u> (String)	type of answer requested (normal text, live chatting, screen sharing)	
	<u>field</u> (String)	the subject field of this question	
	available_time(s) (day of week, time)	(for live chatting or screen sharing only) time available to receive live help	
<b>Outputs</b>	return true on success, return false with error message on failure	<b>Destination</b>	User browser
<b>Action</b>	When submission is successful, store the data to database and redirect to page showing all the questions registered by the user (My questions page). If an error occurred, display error message.		
<b>Requirements</b>	<ul style="list-style-type: none"> <li>• Must be logged in</li> <li>• reward should be less than or equal to the amount of credit that user has</li> <li>• reward should be larger than the minimum reward for each type</li> <li>• available time must be given if type is live chatting or screen sharing</li> <li>• reward should be a multiple of 100</li> <li>• Must confirm user of submitting their input</li> </ul>		

	<ul style="list-style-type: none"> <li>All underlined fields are required</li> </ul>
Pre-condition	-
Post-condition	Question is registered for mentors to view / select for helping.
Side effects	None

**Table 9 Functional Requirement: Question Registration**

# I. Answer Question

<b>Function</b>	Answer Question		
<b>Description</b>	Allow a mentor to answer a registered question		
<b>Source</b>	User input		
<b>Inputs</b>	<u>qid</u> (int)	id of the question to answer	
	content (String)	(normal text answer only) content of the answer	
	arrangement (datetime)	(for live chatting or screen sharing only) arrangement time for helping the mentee	
<b>Outputs</b>	return true on success, return false with error message on failure	<b>Destination</b>	User browser
<b>Action</b>	If the question to answer asks for normal text answer, save the user's answer in the database with the corresponding qid, then refresh page. If the question asks for live chatting or screen sharing, store the selected arrangement time in the database. On failure, display error message.		
<b>Requirements</b>	<ul style="list-style-type: none"> <li>• Must be logged in as a mentor</li> <li>• The question must not already have an answer</li> <li>• Must confirm user of submitting their input</li> <li>• All underlined fields are required</li> </ul>		
<b>Pre-condition</b>	The question is not answered		
<b>Post-condition</b>	The question received an answer or got arranged to receive an answer		
<b>Side effects</b>	None		

**Table 10 Functional Requirement: Answer Question**

## J. Answer Verification

<b>Function</b>	Answer Verification		
<b>Description</b>	Allow a mentee to verify the answer given		
<b>Source</b>	User input		
<b>Inputs</b>	<u>aid</u> (int)	id of the answer	
	<u>star</u> (String)	rating (0 ~ 5 stars) of the answer	
	feedback (String)	feedback of the answer	
<b>Outputs</b>	return true on success, return false with error message on failure	<b>Destination</b>	User browser
<b>Action</b>	When submitted, store the star and feedback of the answer. If the star is above 0, the reward for the question gets sent to the mentor who answered the question. If the star is 0, it does not give the credit to the mentor. The credit will get refunded or given to the mentor after review of the administrative team.		
<b>Requirements</b>	<ul style="list-style-type: none"> <li>• Must be logged in as a mentee</li> <li>• The answer with the 'aid' should be the answer of a question registered by the user</li> <li>• For normal text answers, it can be verified any time after the answer is registered</li> <li>• For live chatting and screen sharing, the answer can be verified at least 30 minutes after the arrangement time.</li> <li>• The question must not already have an answer</li> <li>• Must confirm user of submitting their input</li> <li>• All underlined fields are required</li> </ul>		
<b>Pre-condition</b>	The answer is not verified by the mentee		
<b>Post-condition</b>	The answer is verified by the mentee and the reward gets awarded to the mentor who answered the question (or waits for review).		



Side effects	A mentee can maliciously give inappropriate star and feedback on an answer.
--------------	---

**Table 11 Functional Requirement: Answer Verification**

# K. View Question List

<b>Function</b>	View question list		
<b>Description</b>	Allow a user to view the list of questions registered in the system		
<b>Source</b>	User input		
<b>Input</b>	filter (select)	filter list of questions by order of registration, views, stars, type of answer	
	search (string)	keyword to search the list of questions	
	searchCateg (select)	select whether to search by name, content, or name + content	
<b>Outputs</b>	List of questions registered in the system on success, return false with error message on failure	<b>Destination</b>	User browser
<b>Action</b>	Display list of questions registered in the system. If a filter is selected, the list should be sorted according to the filter. If the user clicks a question, the user should be redirected to the corresponding question page. If the user enters a search string, it should search the list according to the search category given (default: name) and display the matching questions.		
<b>Requirements</b>	<ul style="list-style-type: none"> <li>• Must be logged in</li> <li>• The list should be paginated.</li> <li>• Display each question's title, registered date, views, stars, and type of answer</li> <li>• Display whether the question is answered or not</li> </ul>		
<b>Pre-condition</b>	-		
<b>Post-condition</b>	-		

Side effects	None
--------------	------

**Table 12 Functional Requirement: View Question List**

# L. View Question

<b>Function</b>	View question		
<b>Description</b>	Allow a user to view a question		
<b>Source</b>	User input		
<b>Input</b>	<u>qid</u> (int)	id of the question to view	
<b>Outputs</b>	Question page with answer (if it exists) on success, return false with error message on failure	<b>Destination</b>	User browser
<b>Action</b>	Display the title and content of the selected question with qid. If the question is answered using normal text and the user is the owner of the question, display the answer. If the question is answered using normal text but the user is not the owner of the question, allow user to view the answer by paying 20% of the reward of the question. If the user is the owner of the question and the question is not yet verified, display a button for the user to go to the verification page for that question.		
<b>Requirements</b>	<ul style="list-style-type: none"> <li>• Must be logged in</li> <li>• If a user pays 20% of the reward to view the answer, the user must own more than the amount that he/she needs to pay.</li> <li>• Display whether the question is answered or not</li> </ul>		
<b>Pre-condition</b>	-		
<b>Post-condition</b>	-		
<b>Side effects</b>	None		

**Table 13 Functional Requirement: View Question**

## 6.2 Non-Functional Requirements

### A. Usability

- a. Chrome 브라우저에서 정상적으로 동작하여야 한다.
- b. 멘토와 멘티가 문제를 확인하는데 어려움이 없도록 직관적인 UI 및 UX를 제공한다.
  - i. Bootstrap, Font-awesome 등 오픈소스를 사용하되 필요시 포토샵 등으로 제작하여 사용
  - ii. Form 에러 처리시 에러 메시지가 명확하게 돋보이도록 설계

### B. Performance

- a. 라이브 채팅 및 화면 공유가 정상적인 인터넷 환경에서 끊임없이 동작하여야 한다.
  - i. WebRTC 이용
  - ii. 시스템 접근 권한을 요청해야 할 수도 있다
- b. 모든 서비스의 접속이 3초 이내로 되도록 한다.
  - i. 동시 접속자가 많아도 서비스가 느려지지 않도록 효율적인 설계.
  - ii. 화면 공유 (스트리밍) 고려하여 서버 아키텍처 설계

### C. Dependability

- a. 서버 failure 시 자동으로 복구하도록 한다.
  - i. 서버 재부팅시 필요 서비스가 다시 켜지도록 쉘 스크립트 등록
- b. 유저의 데이터가 정상적으로 저장되고 주기적으로 백업되어야 한다.
  - i. 매일 Mysql dump file 생성
  - ii. DB Operation 로그 저장
- c. 사용하는 라이브러리, 프레임워크, 혹은 브라우저의 업데이트가 발생하여도 정상적으로 작동하여야 한다.

- i. Back-end나 Front-end에서 업데이트 시 문제가 생길 수 있는 모듈인 경우 package.json 에 버전 fix
- ii. Chrome 최신 버전에서 동작하도록 확인

#### D. Security

- a. 인증받지 않은 유저에 대해서 서비스의 이용이 차단되어야 한다.
  - i. Session-based Authentication 사용
  - ii. URL 넘기기로 서비스의 조회가 되지 않도록 라우팅 설정
  - iii. 가장 최신에 로그인한 device 의 세션 사용
- b. 포인트를 사용하지 않은 유저가 포인트 사용이 필요한 항목에 대해 접근이 불가해야 한다.
  - i. JS 난독화 작업 후 배포
  - ii. 포인트 필요한 작업인 경우 필히 데이터베이스에서 권한 확인
- c. 비밀번호가 암호화 되어 저장되어야 한다.
  - i. SHA256 (혹은 더 복잡한 encryption) 사용
- d. 원치않는 개인정보가 제 3자에게 노출되지 않아야 한다.
  - i. HTTPS 통신 사용

## 7. System Models

### 7.1 Objectives

System Modeling은 추상적인 모델을 도식화하여 각 시스템 이해당사자들의 시스템의 이해를 돕는 것을 목적으로 한다. Context Model, Process Model, Interactive Model을 기술하여 각 구성요소 간의 상호작용을 나타냈다.

### 7.2 Context Model

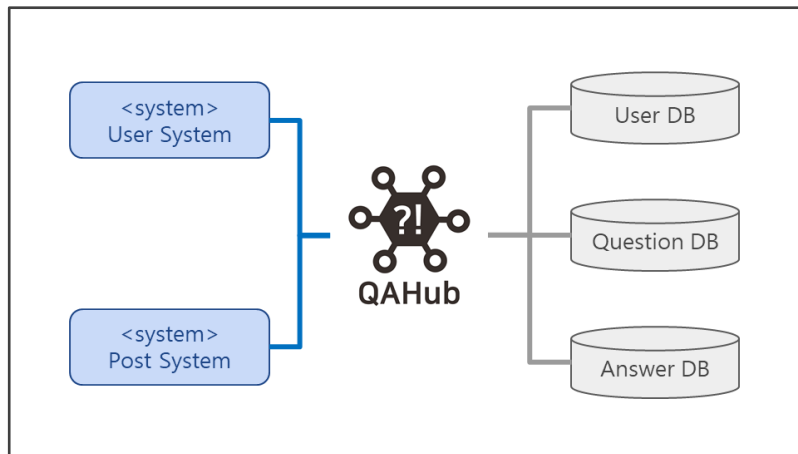


Figure 8 Context Model

### 7.3 Process Model

#### A. Mentor

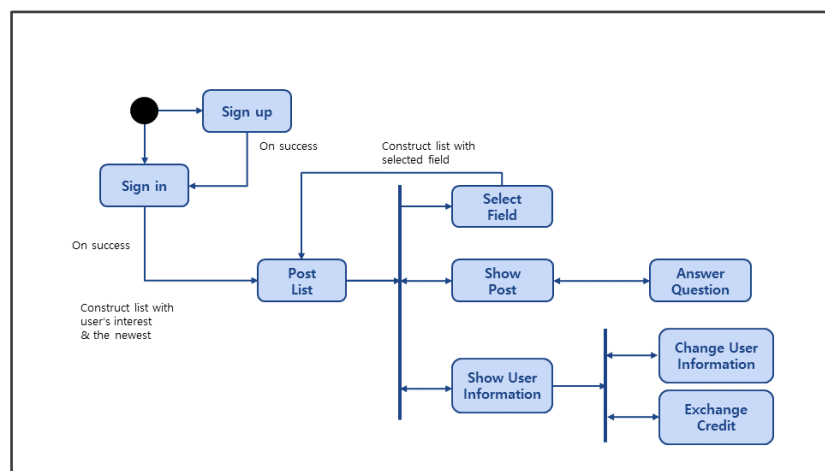
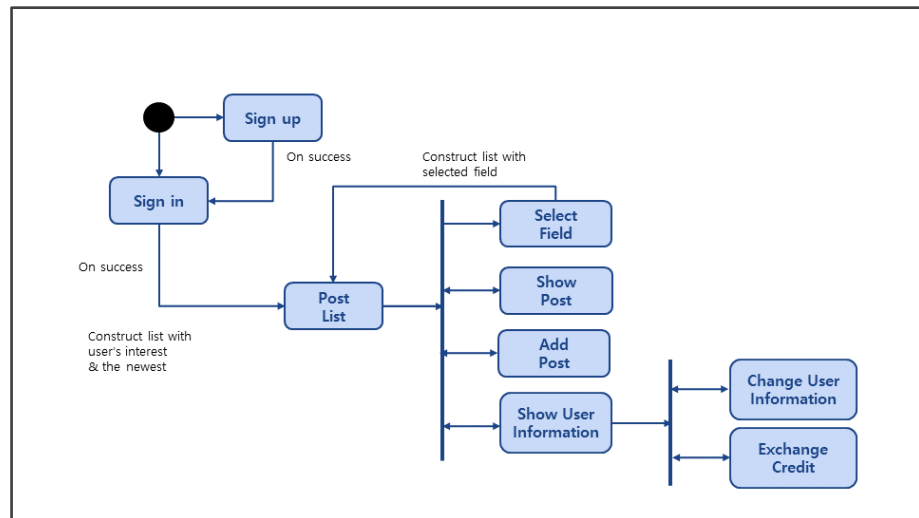


Figure 9 Process Model: Mentor

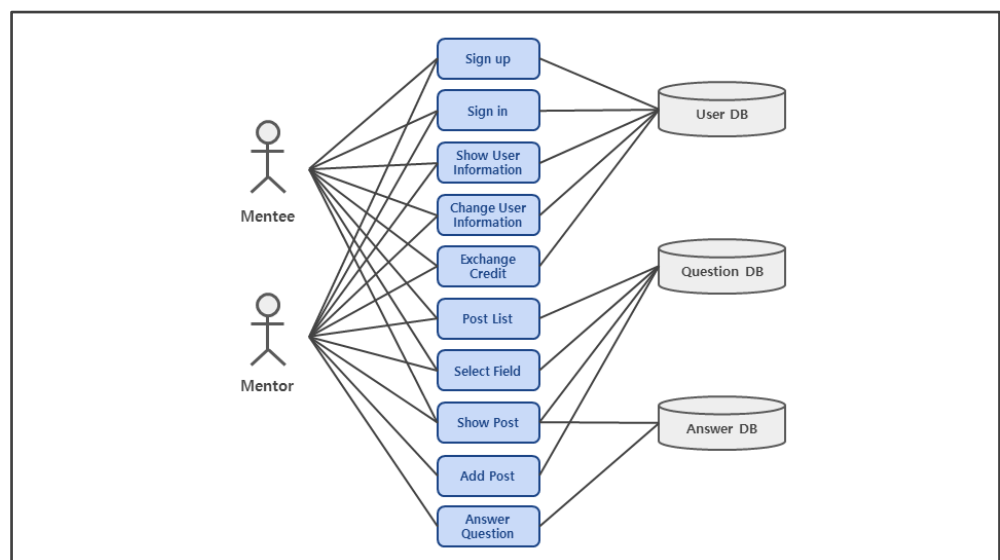
## B. Mentee



**Figure 10 Process Model: Mentee**

## 7.4 Interactive Model

### A. Use Case Diagram



**Figure 11 Use Case Diagram**



B. Tabular Description

a. Sign Up

Use Case	가입하기
Actor	멘토, 멘티
Description	회원 가입에 필요한 정보들을 form 으로 입력 받아, 데이터베이스에 새로운 사용자 계정 정보를 저장한다.
Stimulus	사용자가 form 을 채워 제출 버튼을 클릭한다.
Response	먼저 form 에 입력한 정보가 올바른 형식인지 확인한 후, 맞다면 입력한 정보를 전송한다. 입력한 정보를 바탕으로 DB 에 새로운 유저 생성을 시도하며, 그 결과를 반환한다. 가입 과정이 성공하였을 경우 로그인 페이지를, 실패하였을 경우 가입 form 을 다시 표시한다.

Table 14 Tabular Description: Sign Up

b. Sign In

Use Case	로그인하기
Actor	멘토, 멘티
Description	아이디 및 비밀번호를 입력받아 해당 사용자에게 권한을 부여한다.
Stimulus	사용자가 이메일과 비밀번호를 입력하고 로그인 버튼을 클릭한다.
Response	입력한 정보를 DB 에서 찾은 정보와 대조한다. 일치하는 정보가 있으면 해당 사용자에게 권한을 부여하고, 로그인 결과를 반환한다.

Table 15 Tabular Description: Sign In

c. Show User Information

<b>Use Case</b>	사용자 정보 보기
<b>Actor</b>	멘토, 멘티
<b>Description</b>	사용자가 자신이 입력한 개인 정보를 확인할 수 있다.
<b>Stimulus</b>	메인 화면에서 개인 정보 버튼을 누른다.
<b>Response</b>	DB 에서 해당 사용자를 찾아 개인 정보를 반환한다.

**Table 16 Tabular Description: Show User Information**

d. Change User Information

<b>Use Case</b>	크레딧 환전
<b>Actor</b>	멘토, 멘티
<b>Description</b>	사용자가 자신의 크레딧을 환전할 수 있다.
<b>Stimulus</b>	사용자가 개인 정보 화면에서 크레딧 환전 버튼을 클릭하고 원하는 환전 방법 및 값을 입력하여 확인 버튼을 클릭한다.
<b>Response</b>	입력한 정보로 크레딧을 환전하고, DB 에서 해당 사용자의 크레딧 값을 업데이트한다.

**Table 17 Tabular Description: Change User Information**

e. Exchange Credit

<b>Use Case</b>	크레딧 환전
<b>Actor</b>	멘토, 멘티
<b>Description</b>	사용자가 자신의 크레딧을 환전할 수 있다.
<b>Stimulus</b>	사용자가 개인 정보 화면에서 크레딧 환전 버튼을 클릭하고 원하는 환전 방법 및 값을 입력하여 확인 버튼을 클릭한다.
<b>Response</b>	입력한 정보로 크레딧을 환전하고, DB 에서 해당 사용자의 크레딧 값을 업데이트한다.

**Table 18 Tabular Description: Exchange Credit**

f. Post List

<b>Use Case</b>	게시물 목록
<b>Actor</b>	멘토, 멘티
<b>Description</b>	사용자가 최신 문제들을 확인할 수 있다.
<b>Stimulus</b>	사용자가 메인 화면에 진입한다.
<b>Response</b>	게시물 DB 에서 관심 분야 또는 선택 분야의 최신 문제들의 제목으로 목록을 구성하여 메인 화면에 표시한다.

**Table 19 Tabular Description: Post List**

g. Select Field

<b>Use Case</b>	게시물 분야 선택
<b>Actor</b>	멘토, 멘티
<b>Description</b>	사용자가 원하는 분야의 최신 문제들을 볼 수 있다.
<b>Stimulus</b>	사용자가 메인 화면에서 원하는 분야를 선택한다.
<b>Response</b>	게시물 DB 에서 선택한 분야의 최신 문제들의 제목 목록을 구성하여 메인 화면에 표시한다.

**Table 20 Tabular Description: Select Field**

h. Show Post

<b>Use Case</b>	게시물 보기
<b>Actor</b>	멘토, 멘티
<b>Description</b>	사용자가 선택한 게시물을 볼 수 있다.
<b>Stimulus</b>	사용자가 게시물 목록 또는 개인 정보 등에서 원하는 게시물을 선택한다.
<b>Response</b>	선택한 게시물의 정보를 DB 에서 받아와 화면에 표시한다.

**Table 21 Tabular Description: Show Post**

i. Add Post

Use Case	게시물 작성
Actor	멘티
Description	사용자가 게시물을 작성할 수 있다.
Stimulus	사용자가 메인 화면에서 게시물 작성 버튼을 클릭하여, form 을 채우고 제출 버튼을 클릭한다.
Response	입력받은 form 을 바탕으로 DB 에 게시물 정보를 저장하고, 생성한 게시물을 화면에 표시한다.

**Table 22 Tabular Description: Add Post**

j. Answer Question

Use Case	답변 작성
Actor	멘토
Description	사용자가 게시물에 답변을 할 수 있다.
Stimulus	게시물에 답변이 없을 때 사용자가 게시물 화면의 답변 버튼을 클릭하고, 사용자가 답변을 작성하여 확인 버튼을 누른다.
Response	입력받은 form 을 바탕으로 DB 에 답변 정보를 저장하고, 답변이 포함된 게시물을 다시 표시한다.

**Table 23 Tabular Description: Answer Question**

## C. Sequence Diagram

### a. Sign Up

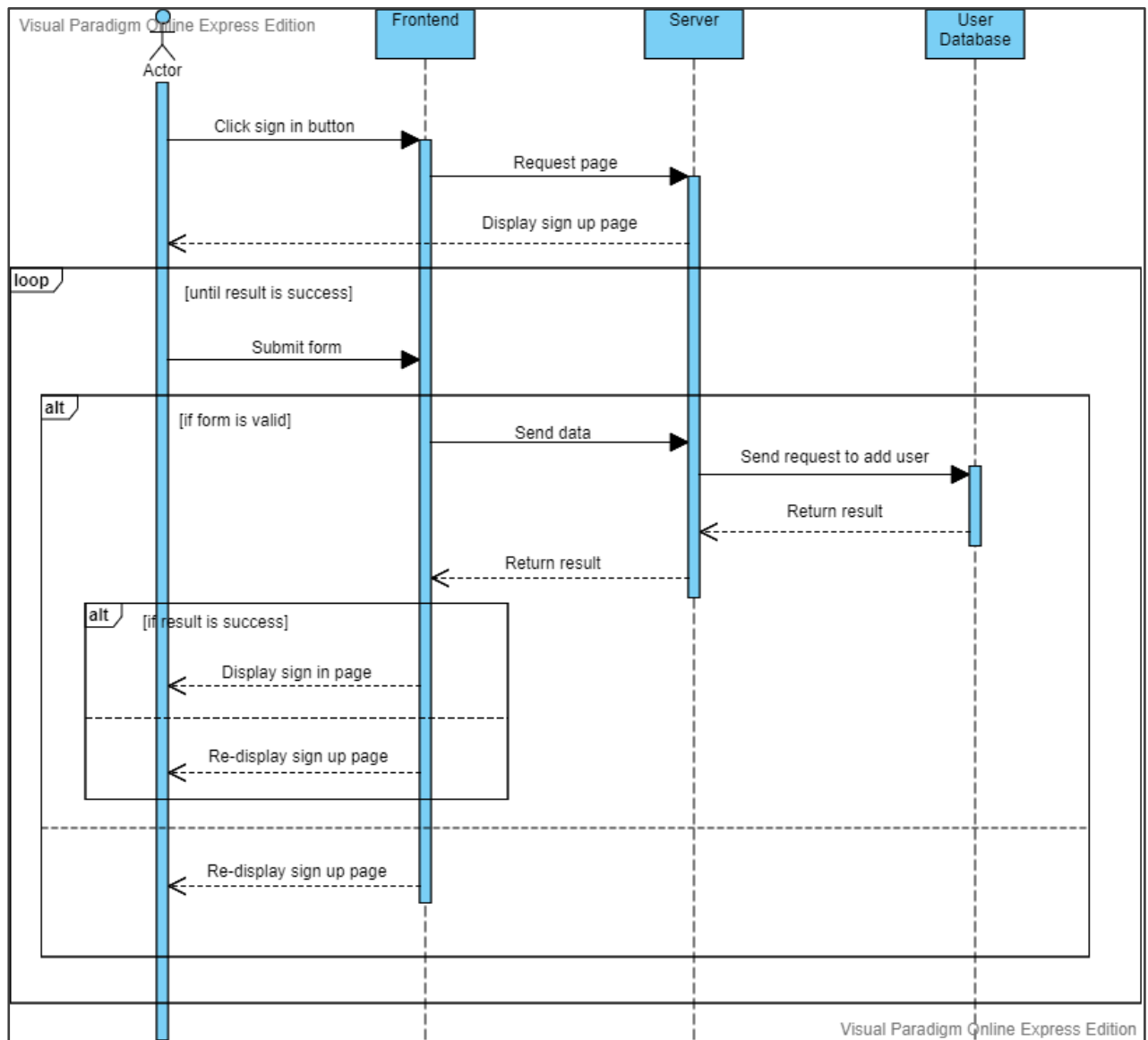


Figure 12 Sequence Diagram: Sign Up

### b. Sign In

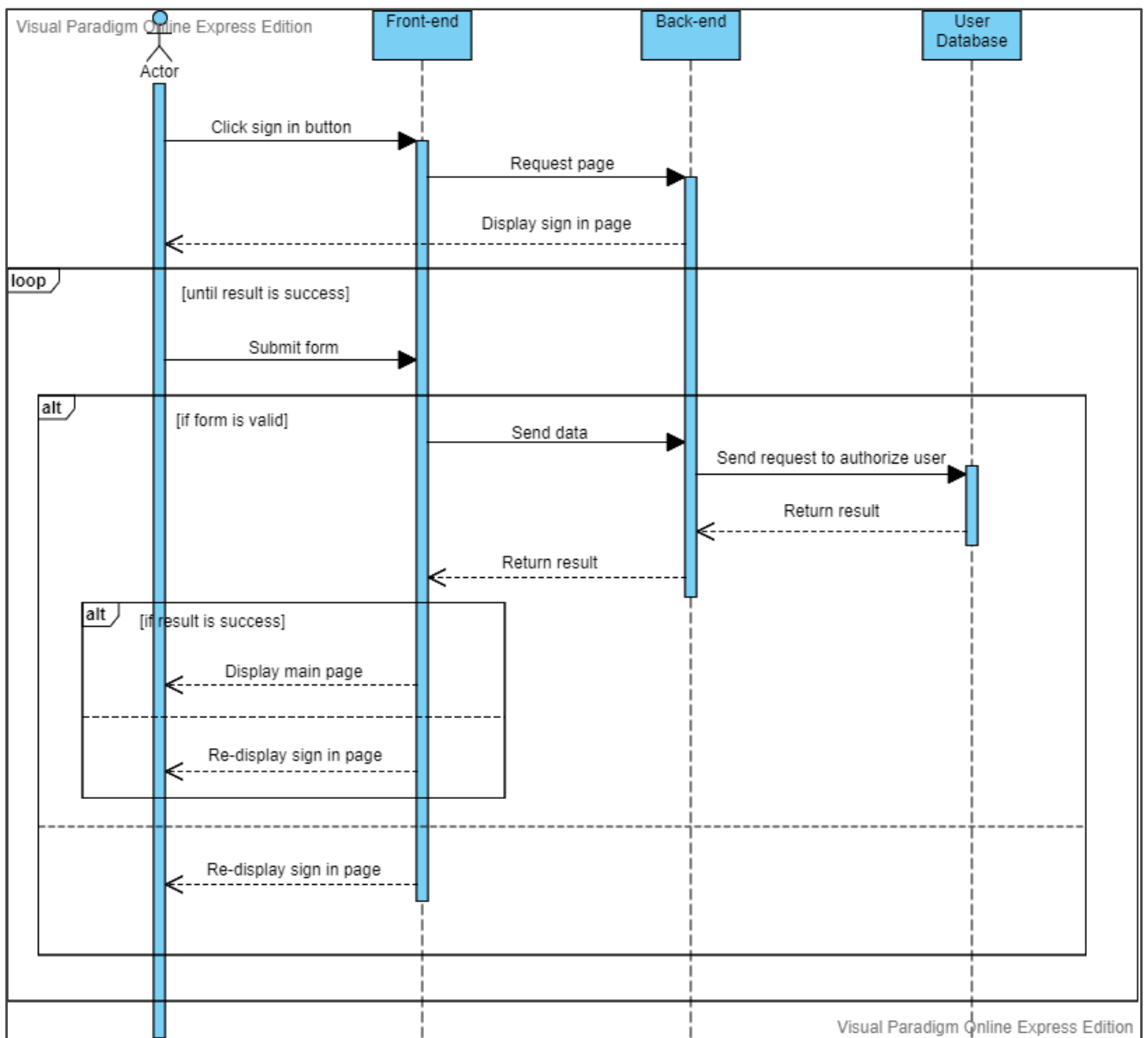


Figure 13 Sequence Diagram: Sign In

c. Show User Information

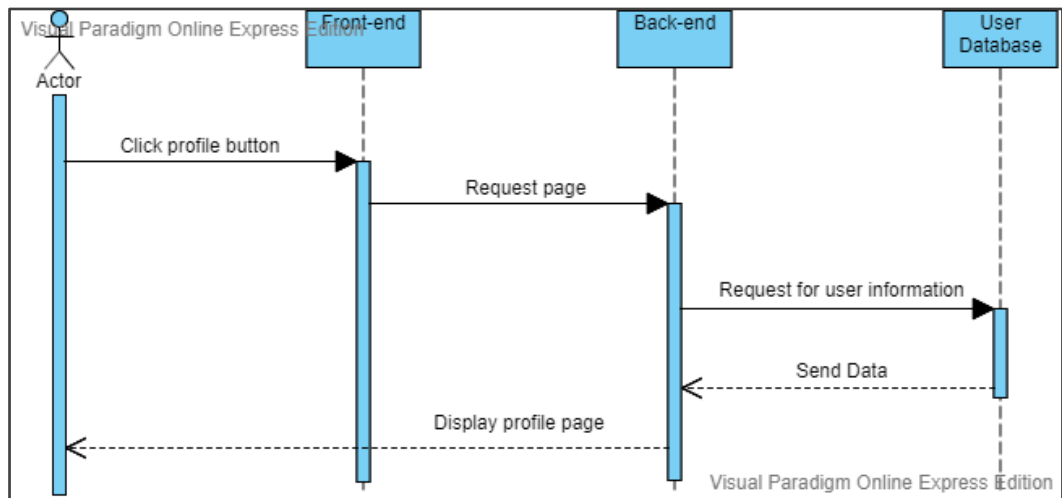


Figure 14 Sequence Diagram: Show User Information



#### d. Change User Information

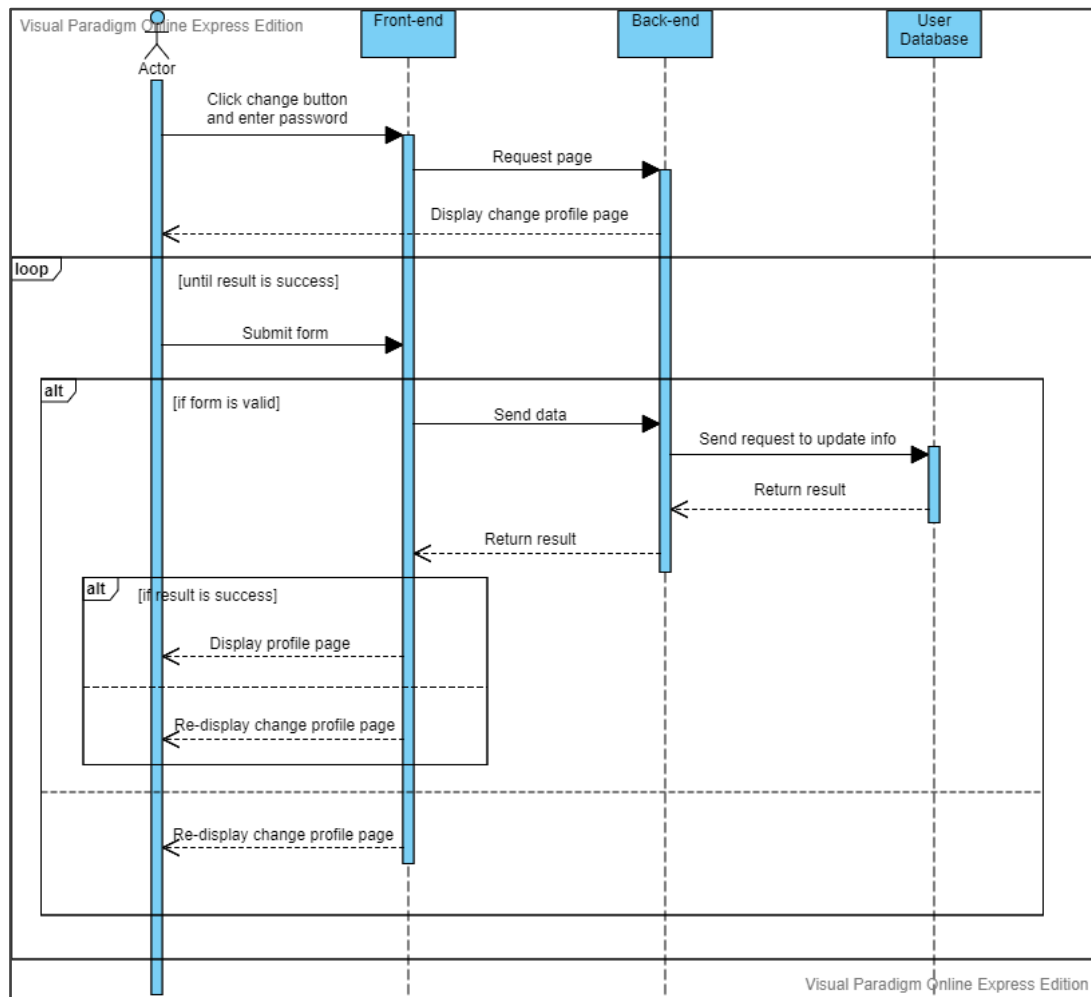


Figure 15 Sequence Diagram: Change User Information

e. Exchange Credit

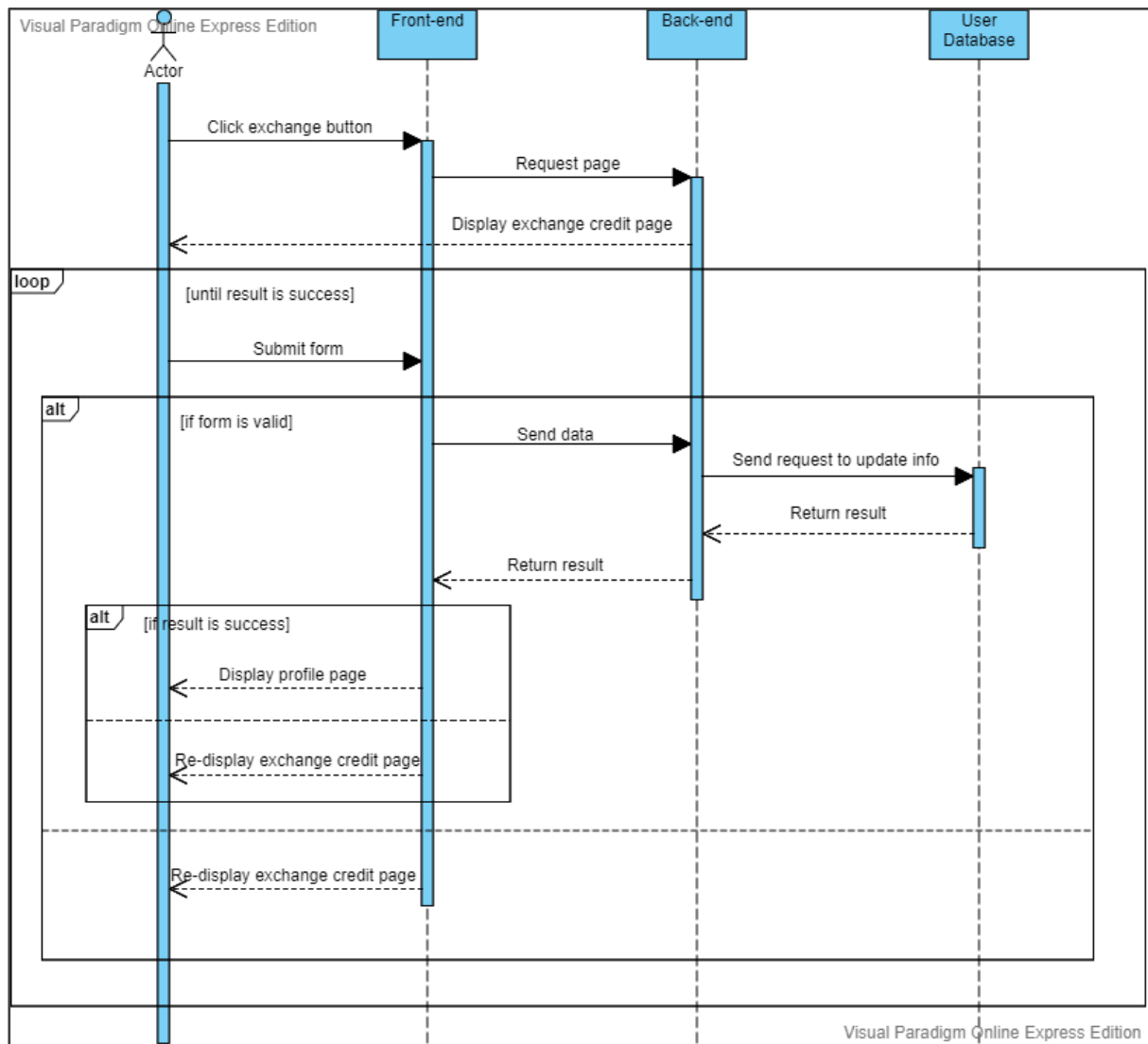


Figure 16 Sequence Diagram: Exchange Credit

f. Post List

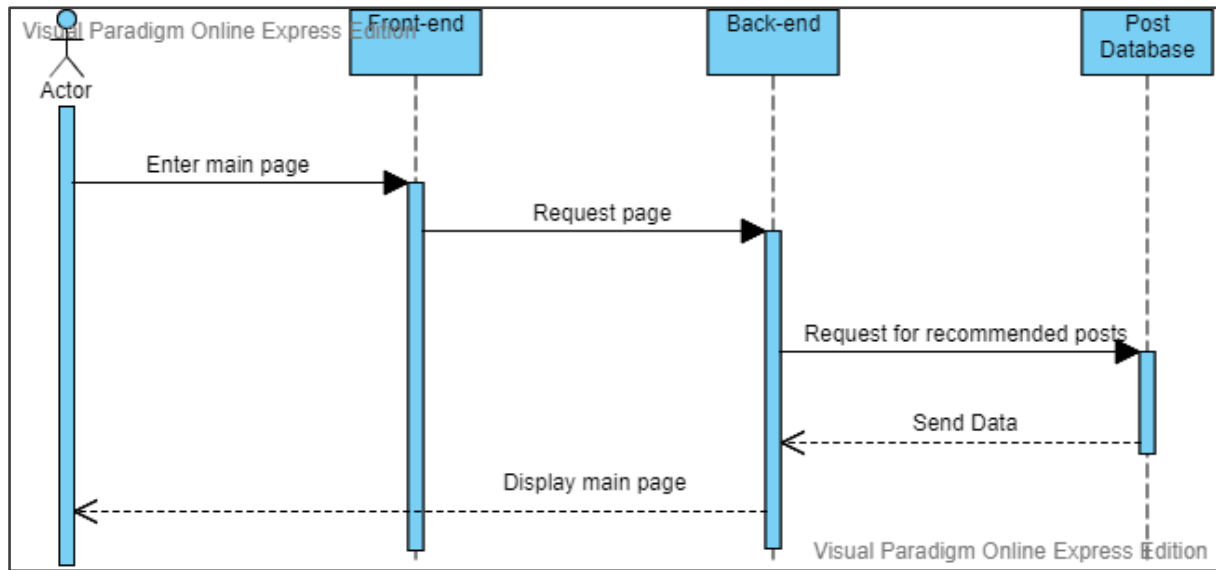


Figure 17 Sequence Diagram: Post List

g. Select Field

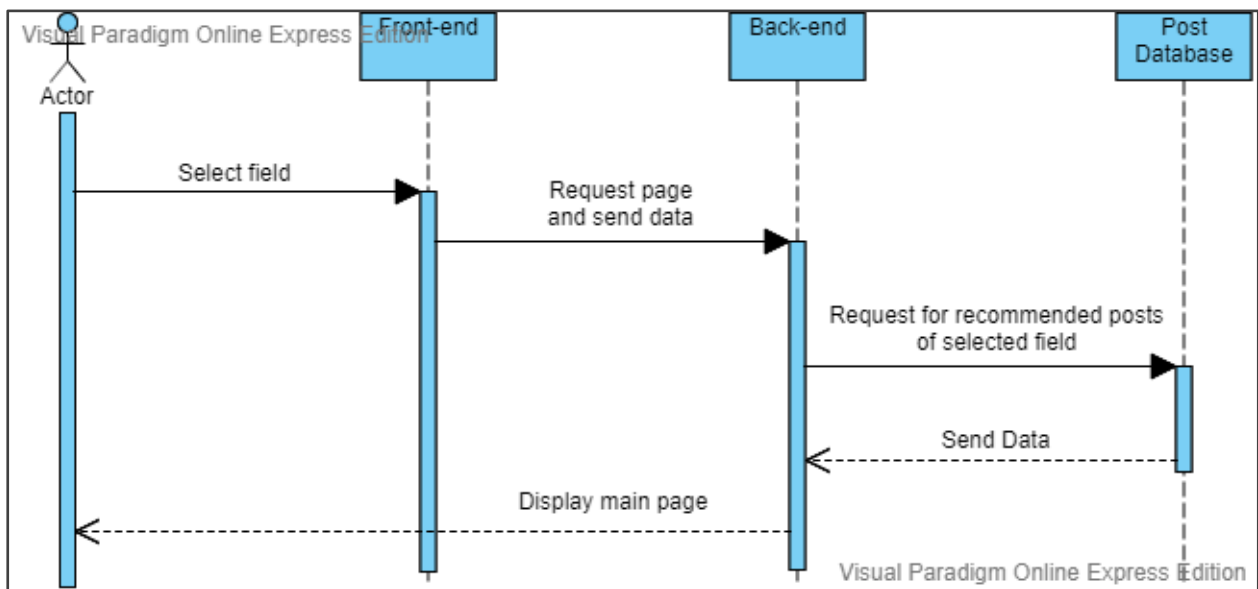


Figure 18 Sequence Diagram: Select Field

### h. Show Post

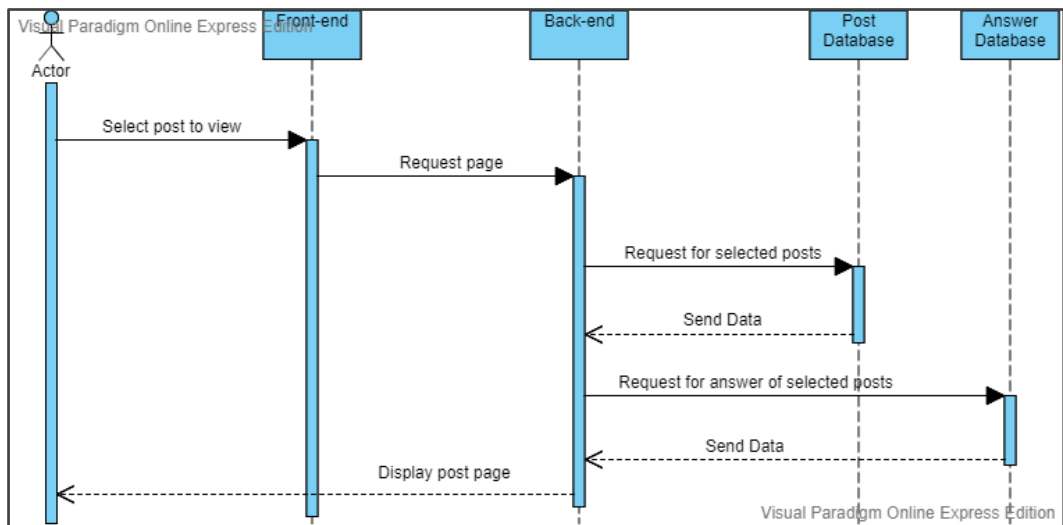


Figure 19 Sequence Diagram: Show Post

### i. Add Post

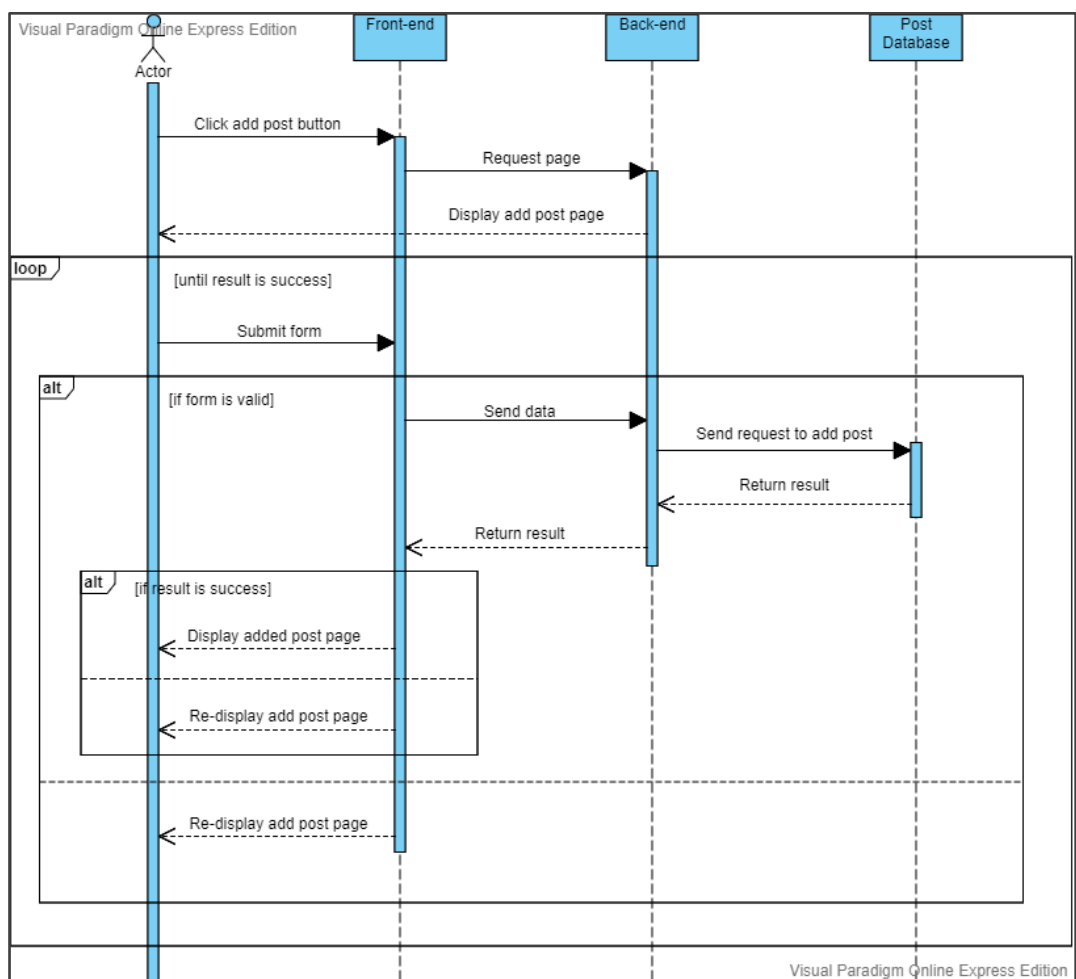
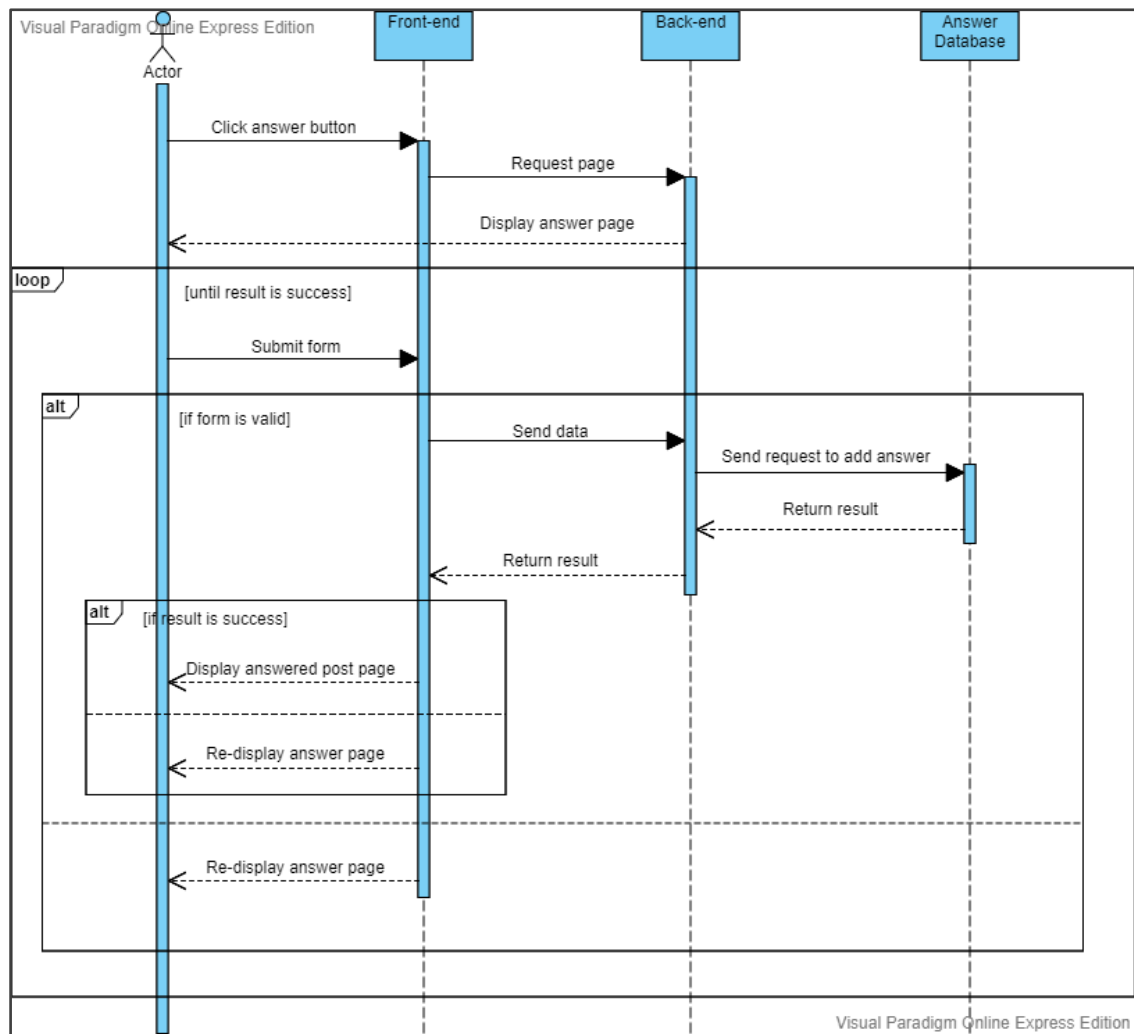


Figure 20 Sequence Diagram: Add Post

## j. Answer Question



**Figure 21 Sequence Diagram: Answer Question**

## 7.5 Structural Models

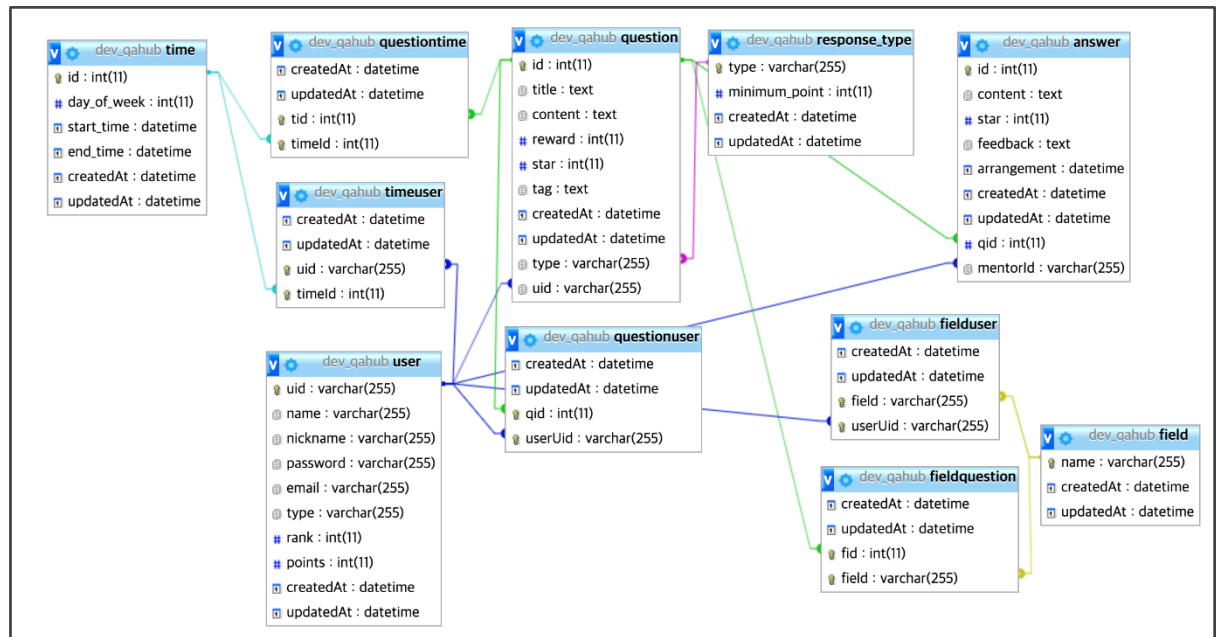


Figure 22 Database UML

## 8. System Evolution

### 8.1 Anticipated changes due to hardware evolution

- Exchanging credit way because of problem that devices do not support banking tech and customer has only that device.
- Optimization each page structure for variable device.
- Automatic backup system for unavoidable trouble in hardware.
- Security about password cracking or system infection from utilizing hardware which is infected.
- System optimization for low performance hardware or high performance hardware.
- Page size control for small device.
- Optimization for device software upgrade.

### 8.2 Changing user needs

- user needs about changing their personal information
- user needs about changing their login information
- user needs about deleting or uploading their activity in site
- user needs about looking posting list which only they want (like word searching function)
- user needs about more category(more various field) then what website provide now
- They might not satisfy our site mentor and they can want more mentor
- They cannot want sharing their posts with other users for various reason
- They cannot want that other users write a comment

## 9. Appendices

### 9.1 Hardware Requirements

#### A. User (Desktop / Laptop)

구분	최소사양	권장사양
운영체제	Windows 7, Windows 8, 또는 Windows 10 정품, MacOS version 10.8 or later	
CPU	CPU 2.0+GHz	CPU 3.0+ GHz
RAM	2GB	4GB or 8GB
네트워크	0.5 Mb/s upstream, 1 Mb/s downstream, wired or wireless	2 Mb/s upstream, 4 Mb/s downstream, wired
입력장치	녹음 가능 입력 장치	헤드셋 마이크
출력장치	소리 재생 가능 출력 장치	

**Table 24 Hardware Requirements: User**

#### B. Server

구분	사양
운영체제	Ubuntu or CentOS latest version
CPU	CPU 3.0+GHz (유동적으로 변경 가능)
RAM	RAM 16GB+
네트워크	0.5 Gb/s upstream, 1 Gb/s downstream, wired
HDD	2 TB (유동적으로 변경 가능)

**Table 25 Hardware Requirements: Server**

\* 어떠한 이유로 종료되었을 시 자동으로 재부팅을 시도하는 기능 포함



## 9.2 Database Requirements

### A. User

사용자 정보와 포인트를 저장하는 Entity

### B. Question

시스템에 등록되는 질문들을 저장하는 Entity

### C. Answer

등록된 질문의 답변들을 저장하는 Entity

### D. Field

멘티의 관심 분야, 질문 분야, 멘토의 전문 분야 등을 저장하는 Entity

### E. Response Type

답변 형식과 각 형식에 최소 포인트를 저장하는 Entity

### F. Time

멘토의 available time 과 멘토-멘티 라이브 채팅 혹은 화면공유를 위한 arrangement 등에 사용되는 시간을 저장하는 Entity

### G. Chatlog

실시간 채팅을 위한 Entity

## 10. Index

### 10.1 Table Index

Table 1 Glossary .....	9
Table 2 Functional Requirement: User Registration.....	20
Table 3 Functional Requirement: User Authentication .....	22
Table 4 Functional Requirement: ID Recovery .....	22
Table 5 Functional Requirement: Password Recovery.....	23
Table 6 Functional Requirement: User Information Modification .....	25
Table 7 Functional Requirement: Purchase Credit.....	26
Table 8 Functional Requirement: Exchange Credit to Currency.....	28
Table 9 Functional Requirement: Question Registration .....	30
Table 10 Functional Requirement: Answer Question .....	31
Table 11 Functional Requirement: Answer Verification .....	33
Table 12 Functional Requirement: View Question List.....	35
Table 13 Functional Requirement: View Question.....	36
Table 14 Tabular Description: Sign Up.....	41
Table 15 Tabular Description: Sign In.....	41
Table 16 Tabular Description: Show User Information .....	42
Table 17 Tabular Description: Change User Information.....	42
Table 18 Tabular Description: Exchange Credit.....	43
Table 19 Tabular Description: Post List .....	43
Table 20 Tabular Description: Select Field.....	44
Table 21 Tabular Description: Show Post.....	44
Table 22 Tabular Description: Add Post.....	45

Table 23 Tabular Description: Answer Question.....	45
Table 24 Hardware Requirements: User .....	56
Table 25 Hardware Requirements: Server.....	56

## 10.2 Figure Index

Figure 1 커뮤니티 '에브리타임'에서의 Chegg에 대한 꾸준한 수요 .....	6
Figure 2 StackOverflow를 사용한 시간 절약 효과 설문(2019).....	8
Figure 3 Overall Architecture.....	15
Figure 4 Front-end Architecture.....	16
Figure 5 Back-end Architecture .....	16
Figure 6 Live-chatting Architecture .....	17
Figure 7 Screen-sharing Architecture .....	18
Figure 8 Context Model.....	39
Figure 9 Process Model: Mentor .....	39
Figure 10 Process Model: Mentee .....	40
Figure 11 Use Case Diagram.....	40
Figure 12 Sequence Diagram: Sign Up .....	46
Figure 13 Sequence Diagram: Sign In .....	47
Figure 14 Sequence Diagram: Show User Information.....	48
Figure 15 Sequence Diagram: Change User Information .....	49
Figure 16 Sequence Diagram: Exchange Credit .....	50
Figure 17 Sequence Diagram: Post List.....	51
Figure 18 Sequence Diagram: Select Field.....	51
Figure 19 Sequence Diagram: Show Post .....	52

Figure 20 Sequence Diagram: Add Post.....	52
Figure 21 Sequence Diagram: Answer Question .....	53
Figure 22 Database UML.....	54