##### Project Document

System Model (Sequence Diagram) Document

제 야간자율학습 조

조원 : 고태완, 김지혜, 최수연

지도교수: 이영석 (서명)

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Rev# | Date | Affected Section | Author |
| 1 | 2020/05/14 |  | 고태완, 김지혜, 최수연 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1.1. Objective 4](#_Toc41677081)

[2. Use Case Diagram 5](#_Toc41677082)

[3. Sequence Diagram 6](#_Toc41677083)

[3.1. 회원가입 6](#_Toc41677084)

[3.2. 로그인 8](#_Toc41677085)

[3.3. 레벨 테스트 9](#_Toc41677086)

[3.4. 강의 수강 11](#_Toc41677087)

[3.5. 코딩 실습 12](#_Toc41677088)

[3.6. Shell 실습 13](#_Toc41677089)

[3.7. Q&A 조회 14](#_Toc41677090)

[3.8. Q&A 질문 작성 15](#_Toc41677091)

[3.9. Q&A 답변 작성 16](#_Toc41677092)

[3.10. 강좌 업로드 17](#_Toc41677093)

[3.11. 수강 중 강좌 목록 조회 20](#_Toc41677094)

[3.12. 사용자의 Q&A 내역 조회 21](#_Toc41677095)

Introduction

## Objective

이 문서는 SW 자율학습 강의 Docker+웹 서비스 개발 시스템의 시스템 모델(시퀀스 다이어그램)에 대한 내용을 기술하고 있다. 요구사항 명세 단계에서 작성한 유스케이스 다이어그램을 기반으로 각 유스케이스의 상세한 내부 동작 흐름을 시퀀스 다이어그램으로 모델링한다.

# Use Case Diagram

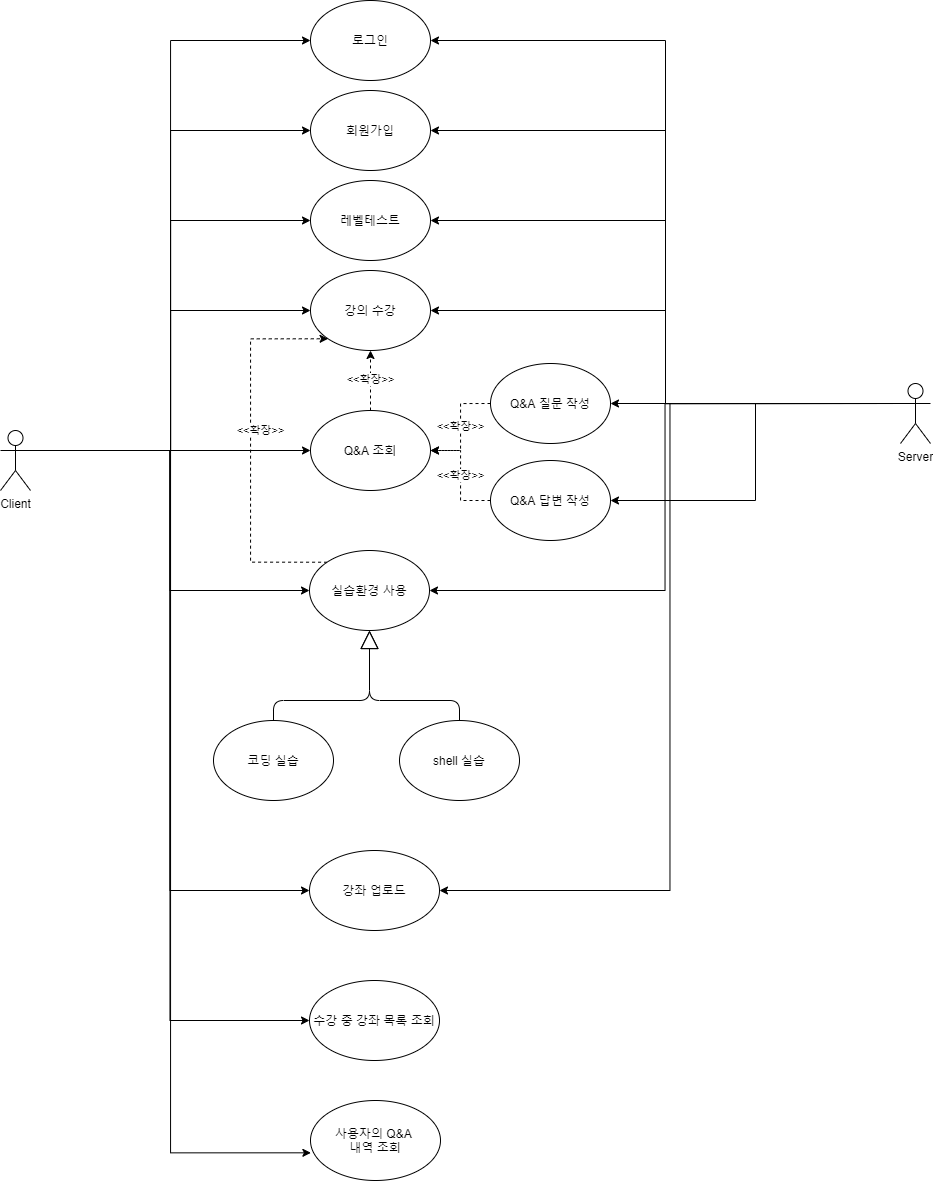
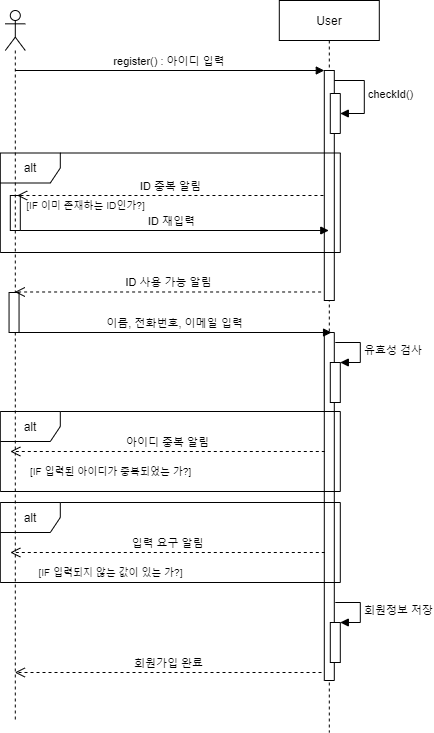


Figure 1 – SW 자율학습 웹 서비스 Use Case Diagram

# Sequence Diagram

## 회원가입

User의 요청에 따라 사용자 계정을 생성한다.



1. 사용자는 User 클래스에게 register() 실행을 요청하고, 아이디를 입력한다.

2. User 클래스는 checkId()로 id 중복체크를 한다.

2.1. 이미 존재하는 id일 경우 사용자에게 중복을 알린다

2.2. 사용자는 id를 재입력한다.

3. User 클래스는 사용가능한 id일 경우 사용자에게 이를 알린다.

4. 사용자는 이름, 전화번호, 이메일을 입력한다.

5. User 클래스는 유효성 검사를 한다.

5.1. 아이디가 중복일 경우 User는 사용자에게 아이디가 중복되어서 회원가입이 불가능함을 알린다.

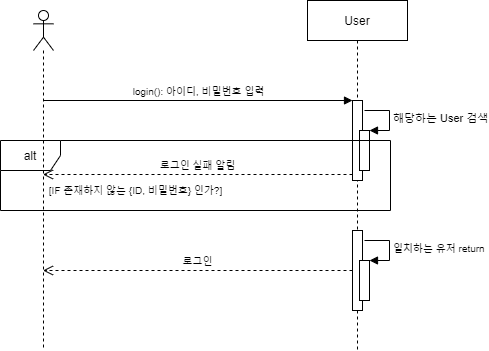
5.2. 입력되지 않은 값이 있을 경우 User는 사용자에게 이를 알린다.

6. User는 사용자가 입력한 값들이 모두 정상적이고 id가 사용 가능한 상태이면 User의 회원 정보를 저장한다.

7. User는 사용자에게 회원가입 완료를 알린다.

## 로그인

사전에 회원가입을 통해 요청된 User의 정보에 따라 인증된 사용자임을 구분한다.



1. 사용자는 로그인을 위해 ID, 비밀번호를 입력한다.

2. User는 해당 ID와 비밀번호와 일치하는 User 데이터를 탐색한다.

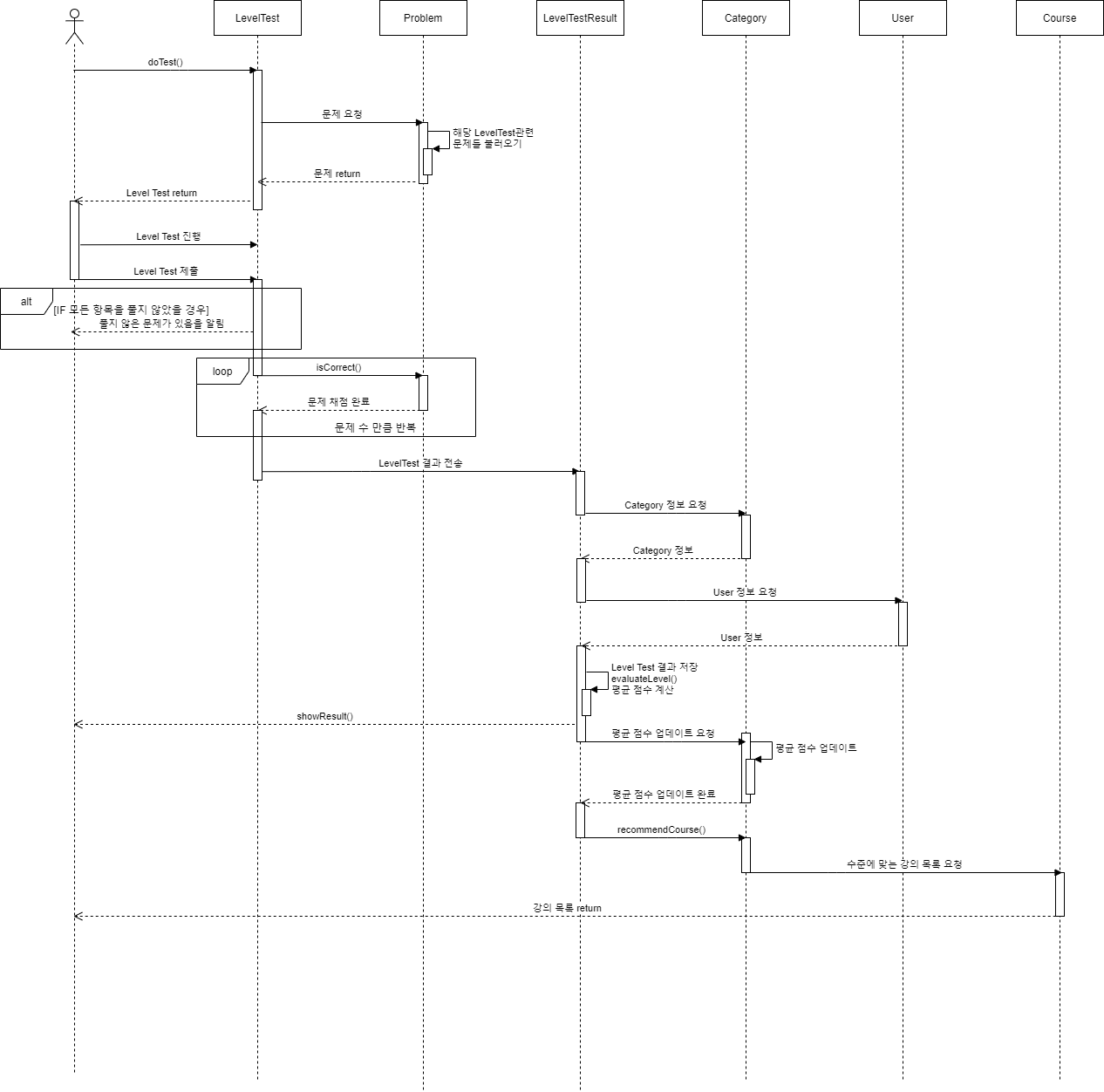
2.1. 존재하지 않는 {ID, 비밀번호}일 경우 로그인 실패를 알린다.

3. 만약 일치하는 유저를 찾았을 경우 Return 한다.

4. 사용자에게 탐색한 User에 대한 데이터 및 인증 토큰을 보낸다.

## 레벨 테스트

특정 카테고리에서 사용자의 수준을 측정할 수 있다.



1. 사용자는 LevelTest에게 doTest()를 요청하여 레벨 테스트를 볼 것을 요청한다.

1.1. LevelTest는 Problem에게 문제들을 요청한다.

1.2.1 Problem은 해당 레벨 테스트 관련 문제들을 불러온다.

1.2.2 Problem은 불러온 문제들을 LevelTest에게 제공한다.

1.3. LevelTest는 만들어진 레벨 테스트를 사용자에게 제공한다.

1.4. 사용자는 레벨 테스트를 진행한다.

1.5. 사용자는 레벨 테스트를 제출한다.

1.5.1. 풀지 않는 문제가 있을 경우 이를 알린다.

1.6. 제출이 되었을 경우 LevelTest는 Problem에게 isCorrect()로 문제 하나 당 해당 문제가 맞았는 지 채점할 것을 요청한다.

1.7. Problem은 문제 하나의 채점을 완료한다.

1.8. 1.6~1.7.을 문제 개수 만큼 반복한다.

1.9.1. LevelTest에서 레벨 테스트 결과를 LevelTestResult에 전달한다.

1.9.2. LevelTest는 Category에게 SW 과목에 대한 카테고리 정보를 요청한다.

1.9.3. Category는 카테고리 정보를 LevelTest에게 제공한다.

1.9.4. LevelTestResult는 User에게 User 정보를 요청한다.

1.9.5. User는 LevelTestResult에게 User 정보를 제공한다.

1.9.6. LevelTestResult는 레벨 테스트의 결과를 저장하고, evaluateLevel()로 사용자의 수준을 계산하고 해당 카테고리에 대한 평균 점수를 다시 계산한다.

1.9.7. LevelTestResult는 showResult()로 사용자에게 레벨 테스트 결과를 보여준다.

1.9.8. LevelTestResult는 Category에게 해당 카테고리에 대한 평균 점수를 업데이트 할 것을 요청한다.

1.9.9. Category는 해당 카테고리의 평균 점수를 업데이트한다.

1.9.10. Category는 평균 점수 업데이트를 완료한다.

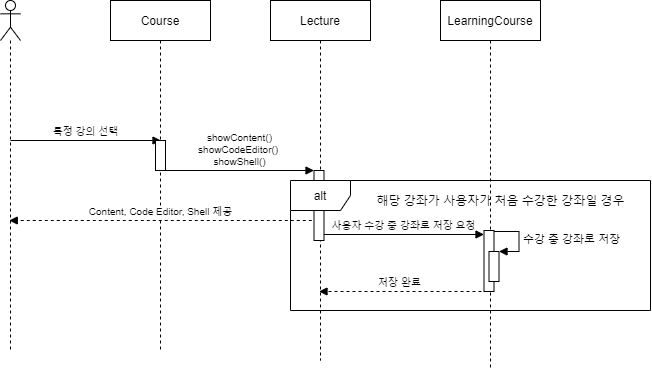
1.9.11.1. LevelTestResult는 Category에게 recommendCourse()를 실행하여 유저에게 강좌들을 추천해 줄 것을 요청한다.

1.9.11.2. Category는 수준에 맞는 강의 목록을 Course에게 요청한다.

2. Course는 사용자에게 수준에 맞는 강의 목록을 제공한다.

## 강의 수강

사용자가 강의를 수강하여 학습을 하는 과정을 나타낸다.



1. 사용자는 특정 강의를 선택한다.

1.1. Course는 Lecture에게 해당 강의의 showContent(), showCodeEditor(), showShell()을 실행하여 강의 내용과 코드 에디터, 쉘을 보여줄 것을 요청한다.

2. Lecture는 사용자에게 해당 content, code Editor, Shell을 제공한다.

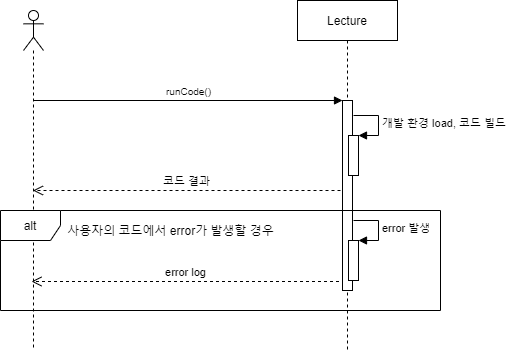
2.1. 해당 강좌가 사용자가 처음 수강한 강좌일 경우 Lecture는 LearningCourse에게 사용자의 수강 중 강좌로 저장해줄 것을 요청한다.

2.2. LearningCourse는 해당 강좌를 수강 중 강좌로 저장한다.

2.3. LearningCourse는 저장을 완료한다.

## 코딩 실습

사용자가 코드를 직접 구현하면서 실습을 진행하는 과정을 나타낸다.



1. 사용자는 Lecture에게 runCode()를 실행하여 자신이 짠 코드를 빌드해줄 것을 요청한다.

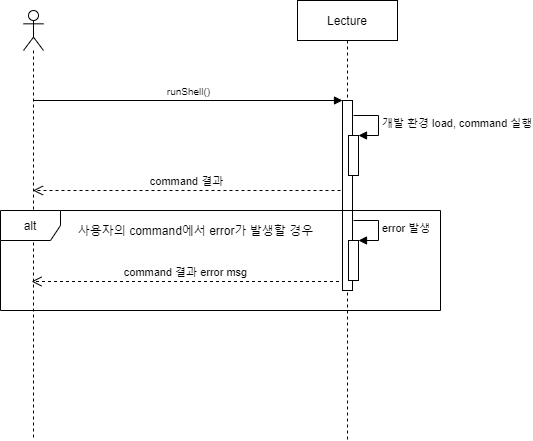
2. Lecture는 해당 강의에 대한 개발 환경을 load하고 사용자의 코드를 빌드한다.

3. Lecture는 사용자에게 코드의 결과를 보여준다.

3.1. 사용자의 코드에서 error가 발생할 경우 Lecture는 사용자에게 error log를 보여준다.

## Shell 실습

사용자가 shell을 사용하여 실습을 진행하는 과정을 나타낸다.



1. 사용자는 Lecture에게 runShell()를 실행하여 자신이 짠 코드를 빌드해줄 것을 요청한다.

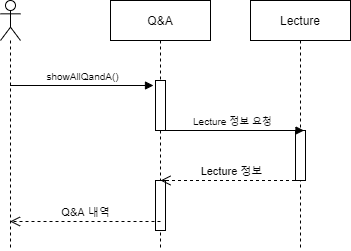
2. Lecture는 해당 강의에 대한 개발 환경을 load하고 사용자의 command를 개발 환경에서 실행한다.

3. Lecture는 사용자에게 command의 결과를 보여준다.

3.1. 사용자의 command에서 error가 발생할 경우 Lecture는 사용자에게 error log를 보여준다.

## Q&A 조회

특정 강의에 대한 Q&A를 조회하는 과정을 나타낸다.



1. 사용자는 Q&A에게 showAllQandA()를 실행하여 특정 강의의 모든 Q&A 내용을 보여줄 것을 요청한다.

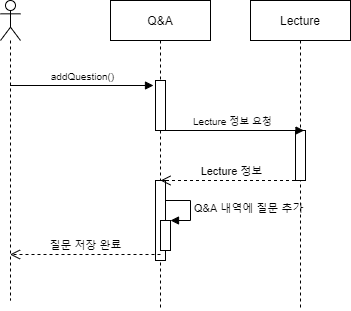
1.1. Q&A는 Lecture에게 해당 강의에 대한 정보를 제공해줄 것을 요청한다.

1.2. Lecture는 Q&A에게 해당 강의에 대한 정보를 제공한다.

2. Q&A는 해당 강의의 Q&A 내역을 사용자에게 제공한다.

## Q&A 질문 작성

사용자가 특정 강의에 대한 Q&A의 질문을 작성하는 과정을 나타낸다.



1. 사용자는 질문을 작성하고 Q&A에게 addQuestion()을 실행하여 질문 등록을 요청한다.

1.1. Q&A는 Lecture에게 Lecture 정보를 요청한다.

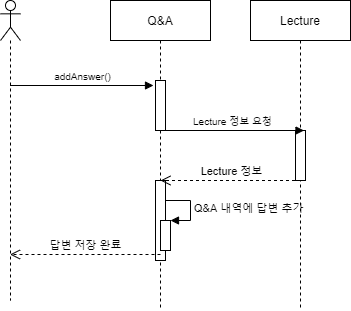
1.2. Lecture는 Q&A에게 Lecture 정보를 제공한다.

2. Q&A는 사용자가 작성한 질문을 Q&A 내역에 추가한다.

3. Q&A는 질문 저장을 완료한다.

## Q&A 답변 작성

사용자가 특정 강의의 Q&A의 질문에 대한 답변을 작성하는 과정을 나타낸다.



1. 사용자는 답변을 작성하고 Q&A에게 addAnswer()을 실행하여 질문에 대한 답변 등록을 요청한다.

1.1. Q&A는 Lecture에게 해당 강의 정보를 요청한다.

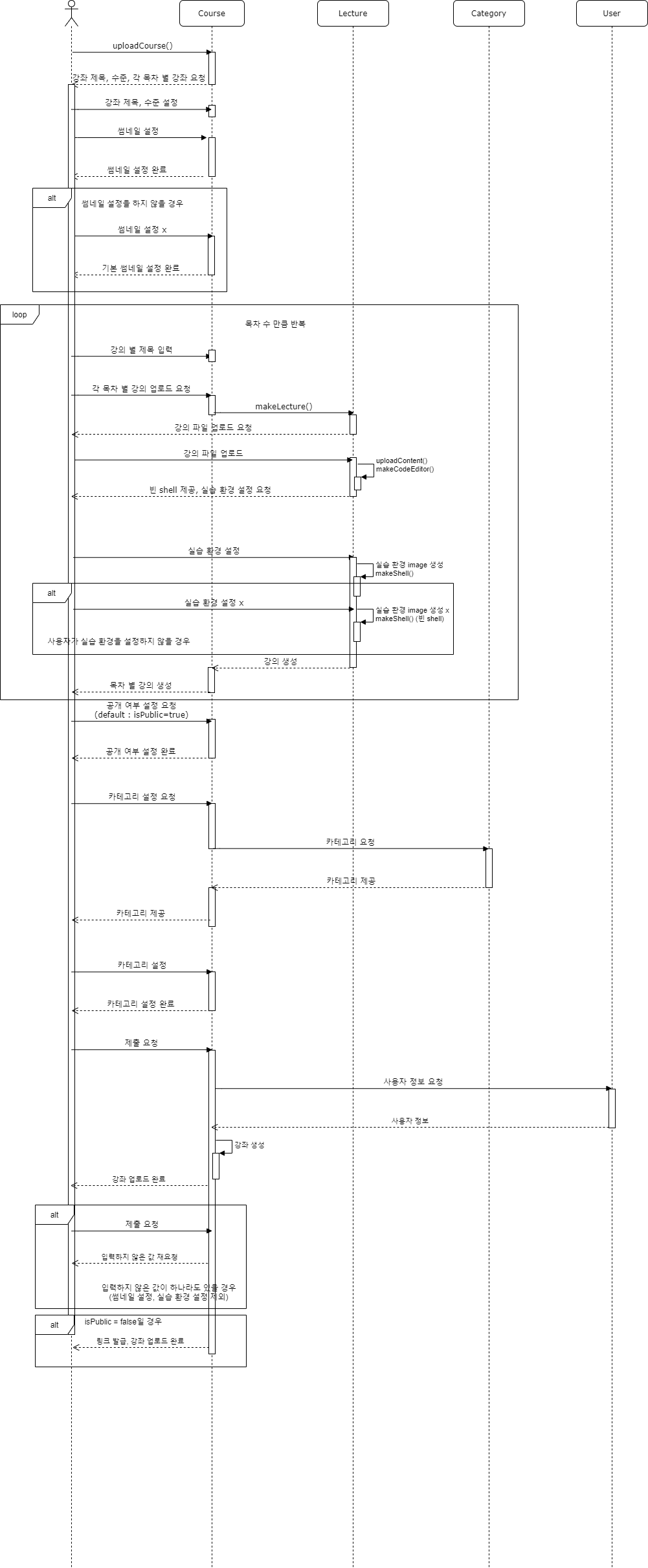
1.2. Lecture는 Q&A에게 해당 강의 정보를 제공한다.

2. Q&A는 사용자가 작성한 답변을 Q&A 내역에 추가한다.

3. Q&A는 답변 저장을 완료한다.

## 강좌 업로드

사용자가 강좌 업로드를 하는 과정을 나타낸다.



1. 사용자는 마이페이지에서 '강좌 업로드'를 선택하여 Course에게 uploadCourse()를 요청한다.

2. Course는 사용자에게 강좌 제목, 수준, 각 목차 별 강좌에 대해 요청한다.

3. 사용자는 강좌 제목과 강좌의 수준을 입력한다.

4. 사용자는 썸네일을 설정한다.

4.1 썸네일이 성공적으로 업로드 되면 사용자는 이를 확인할 수 있다.

4.2 사용자가 썸네일을 설정하지 않은 경우 기본 썸네일이 제공된다.

5. 사용자는 강좌의 강의마다 제목을 입력한다.

5.1 사용자는 각 목차 별 강의에 대해 업로드를 요청한다.

5.2 Course는 강의를 생성하기 위해 Lecture의 makeLecture()를 요청한다.

5.3 Lecture는 사용자에게 강의 파일 업로드를 요청한다.

5.4 사용자는 강의 파일을 업로드한다.

5.5 Lecture는 uploadContent()를 호출하여 강의 파일을 저장하고 makeCodeEditor()를 호출하여 강의에 대한 실습 환경을 생성한다.

5.6 Lecture는 사용자에게 빈 shell을 제공하며 실습 환경 설정을 요청한다.

5.7 사용자는 실습 환경을 설정한다.

5.8 Lecture는 실습 환경을 바탕으로 image를 생성하고 makeShell()를 통해 실습 환경을 생성한다.

5.8.1 사용자가 실습 환경을 별도로 설정하지 않는 경우 image를 생성하지 않으며 makeShell()를 통해 빈 shell 상태로 실습 환경을 생성한다.

5.9 Lecture는 하나의 강의 생성을 완료한다.

5.10 사용자는 목차별 강의가 생성되었음을 확인한다. 사용자가 원하는 만큼 5~5.10 과정을 반복한다.

6. 사용자는 강좌에 대한 공개 여부를 설정한다.

7. 사용자는 강좌가 전체 공개로 설정되었음을 확인한다.

8. 사용자는 카테고리를 요청한다.

9. Course는 Category에게 전체 과목 종류를 요청한다.

10. Category는 Course에게 전체 과목 종류를 제공한다.

11. 사용자에게 SW과목 카테고리(목록)를 제공한다.

12. 사용자는 강좌가 해당하는 SW과목의 카테고리를 설정한다.

13. 사용자는 강좌의 카테고리가 설정 완료 되었음을 확인한다.

14. 사용자는 강좌를 업로드 하기위해 제출을 요청한다.

14.1. Course는 User에게 사용자의 정보를 요청한다.

14.2. User는 Course에게 사용자 정보를 전달한다.

14.3 Course는 강좌를 생성하여 업로드를 완료한다.

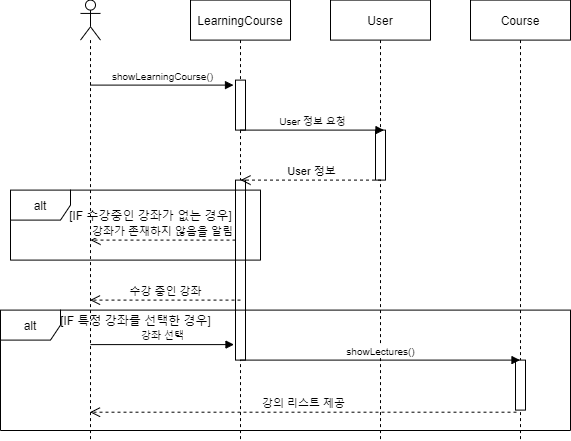
14.3.1 입력하지 않은 값이 존재하는 경우 이에 대한 입력값을 재요청한다.

14.3.2 강좌가 공개 범위가 전체가 아닌 경우 링크를 추가적으로 발급하며 강좌 업로드를 완료한다.

14.4 사용자는 강좌가 업로드 완료되었음을 확인한다.

## 수강 중 강좌 목록 조회

사용자 자신의 수강 중인 강좌 목록 조회를 하는 과정을 나타낸다.



1. 사용자는 마이페이지에서 '내 강좌'를 선택한 뒤 '수강 중인 강좌'를 선택하여 LearningCourse에게 showLearningCourse()를 요청한다.

1.1. LearningCourse는 User에게 사용자의 정보를 불러올 것을 요청한다.

1.2. User는 LearningCourse에게 사용자의 정보를 제공한다.

2. 사용자의 수강 중 강좌 목록을 제공한다.

2.1 사용자의 수강중인 강좌가 존재하지 않는 경우 이를 알린다.

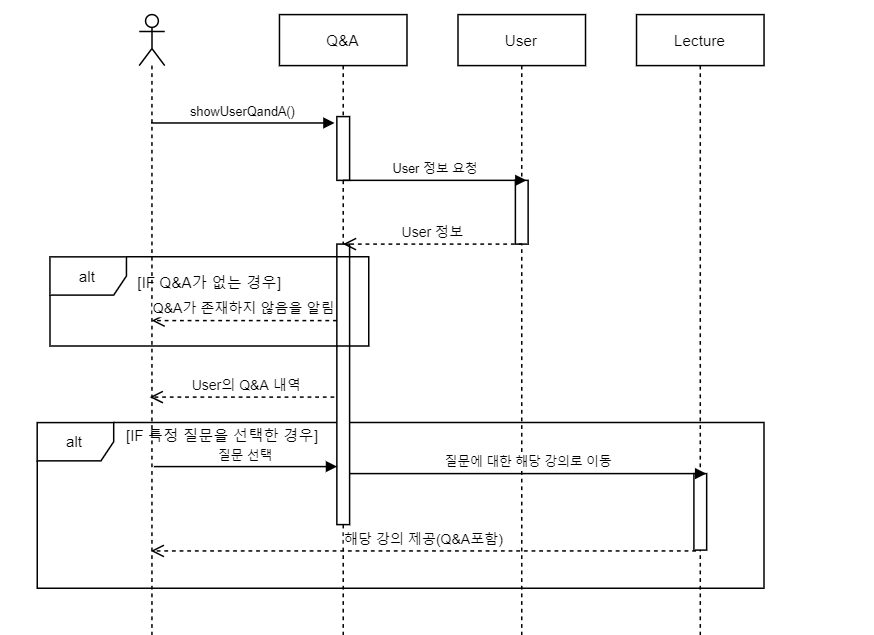
2.2.1 사용자가 특정 강좌로 들어갈 때 LearningCourse에 있는 특정 강좌를 선택한다.

2.2.2. LearningCourse는 Course에게 해당 강좌에 대한 목차로서 강의들을 불러올 것을 요청한다.

2.2.3. Course는 해당 강좌에 대한 강의 리스트들을 사용자에게 제공한다.

## 사용자의 Q&A 내역 조회

사용자가 자신의 Q&A 내역을 조회하는 과정을 나타낸다.



1. 사용자는 마이페이지에서 '내 강좌'를 선택한 뒤 '내 Q&A'를 선택하여 Q&A에게 showUserQandA()를 요청한다.

1.1 QandA는 User에게 현재 사용자 정보를 요청한다.

1.2 User는 Q&A에게 현재 사용자 정보를 제공한다.

2. 사용자의 Q&A 목록을 제공한다.

2.1 사용자의 Q&A 목록이 존재하지 않는 경우 이를 알린다.

2.2.1 사용자가 특정 Q&A 로 들어갈 때 Q&A에 있는 질문을 클릭한다.

2.2.2 Q&A는 Lecture에게 질문에 대한 해당 강의로 이동할 것을 요청한다.

2.2.3 Lecture는 사용자에게 Q&A를 포함한 해당 강의를 제공한다.