
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, 김지 혜, 최수연

Table of Contents

목차

1. INTRODUCTION	5
1.1. OBJECTIVE	5
2. USE CASE DIAGRAM.....	6
3. SEQUENCE DIAGRAM	9
3.1. 회원가입	9
3.2. 로그인	11
3.3. 아이디 찾기	12
3.4. 비밀번호 찾기.....	13
3.5. 레벨 테스트	14
3.6. 강의 추천	15
3.7. 이론 학습	16
3.8. 복습 테스트	17
3.9. 강의 평가	18
3.10. 실습 학습	19
3.11. 환경 가이드라인.....	21
3.12. 자동 환경설정.....	22
3.13. 강좌 업로드	23
3.14. 테스트 생성	26
3.15. 수강 중 강좌 목록 조회.....	28
3.16. 강좌 이어서 수강	29
3.17. 오답노트 확인.....	30
3.18. 찜한 강좌 조회	31
3.19. 찜한강좌 수강.....	32
3.20. 찜한 강좌 삭제	33
3.21. 강좌 책갈피 조회	34
3.22. 강의 책갈피 사용	35
3.23. Q&A 내역 조회	36
3.24. QANDA 내역 상세 조회.....	37
3.25. 비밀번호 변경.....	38

3.26. 강좌 알림 관리	40
3.27. 이메일 수신 관리	41

1. Introduction

1.1. 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.

2. Use Case Diagram

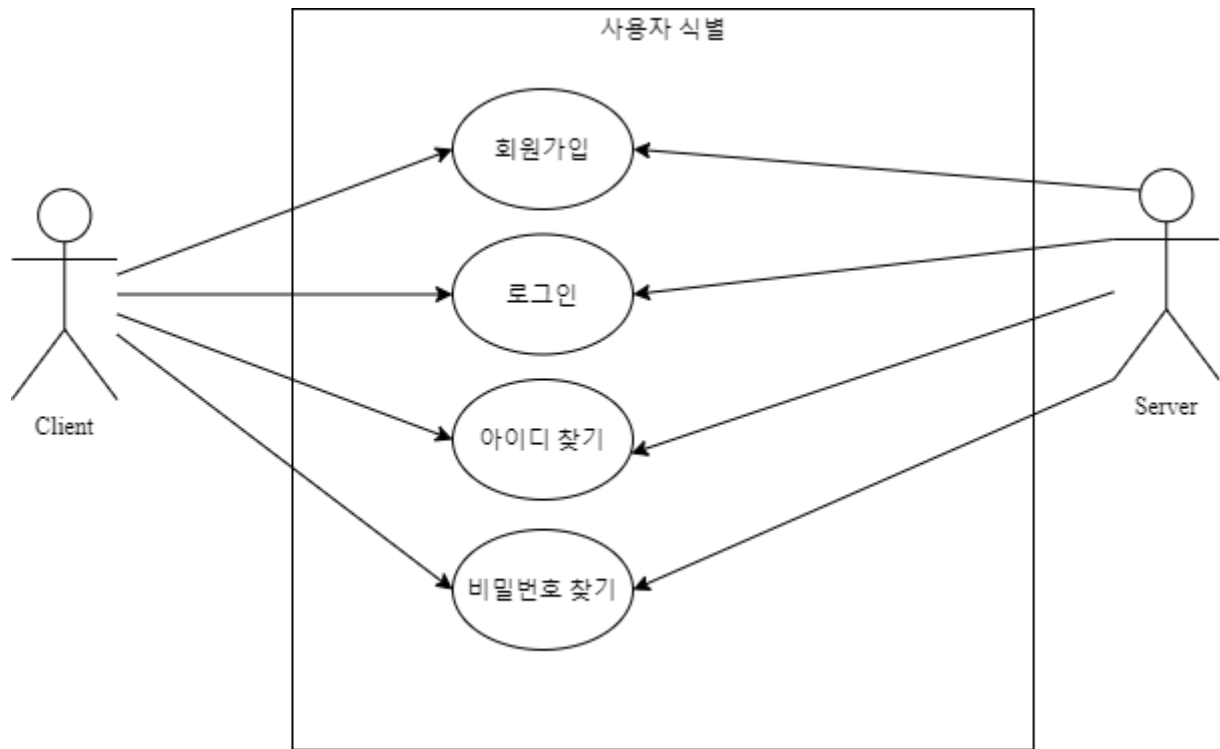


Figure 1 – 유저 식별 Use Case Diagram

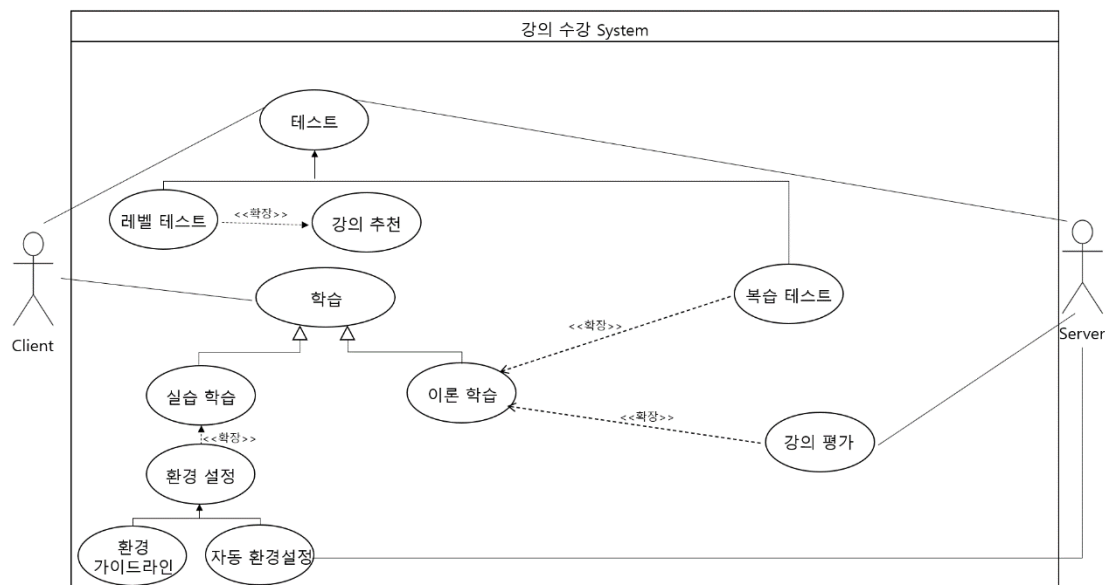


Figure 2 – 강의 수강 Use Case Diagram

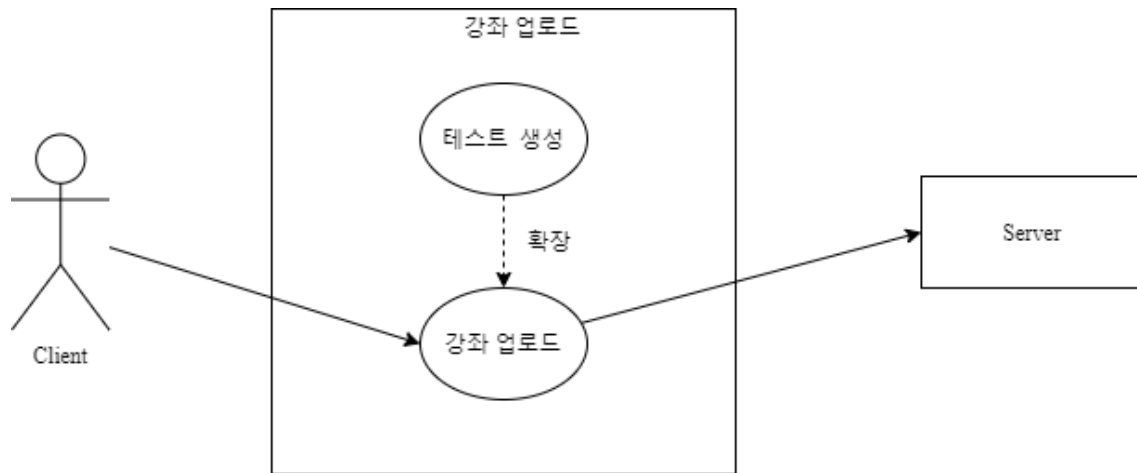


Figure 3 – 강좌 업로드 Use Case Diagram

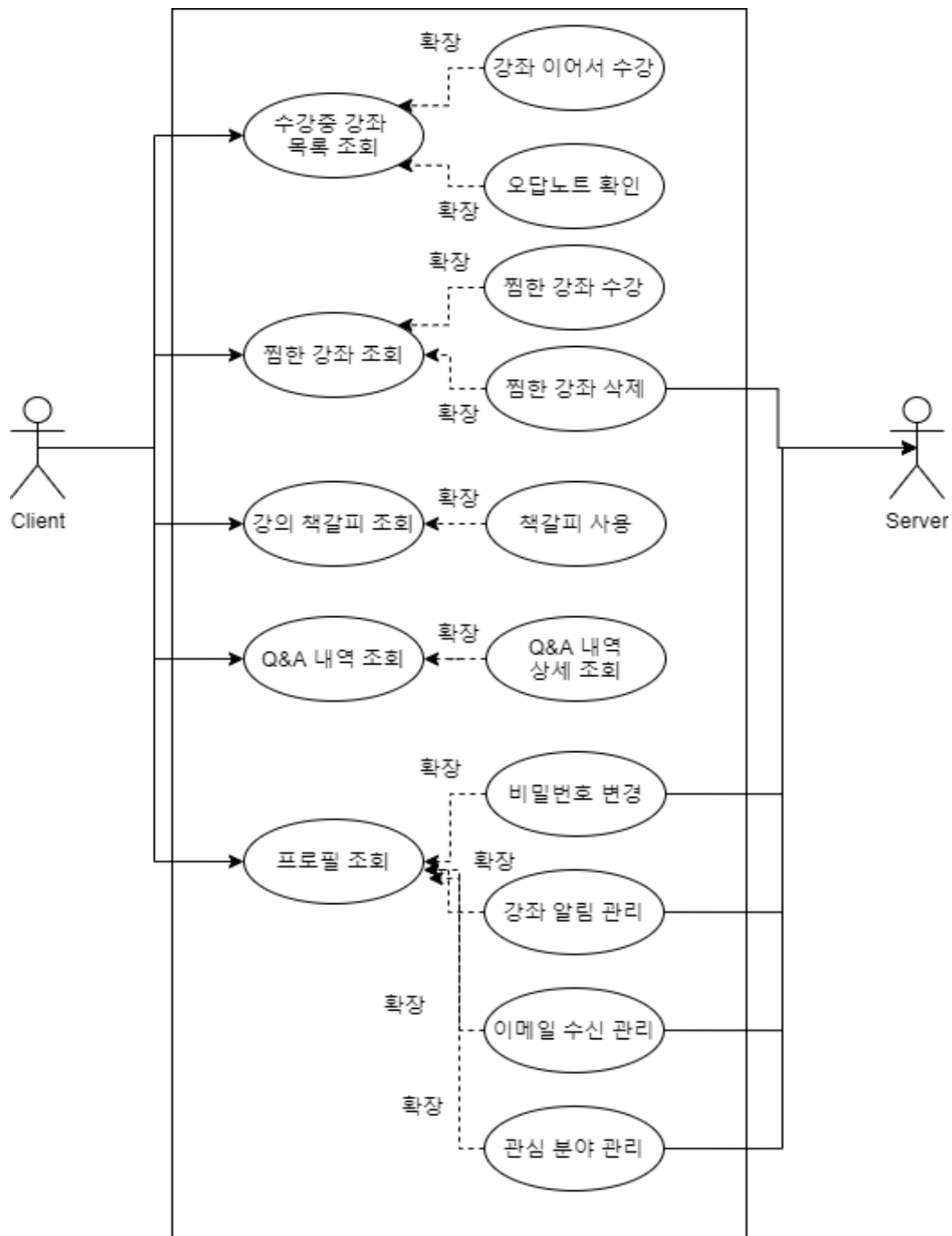
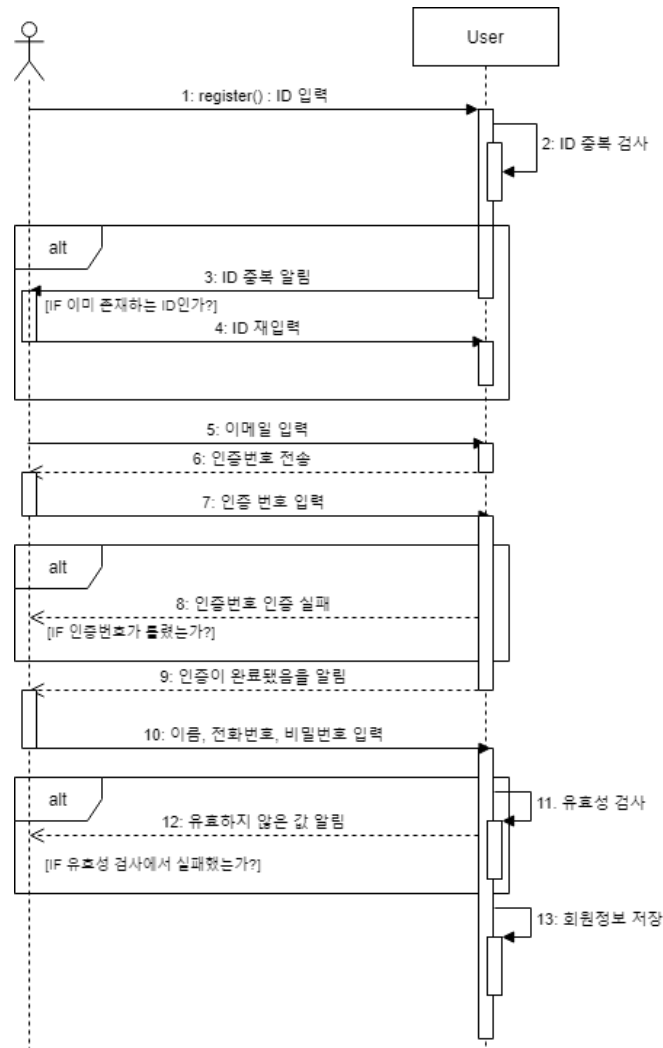


Figure 4- 마이페이지 Use Case Diagram

3. Sequence Diagram

3.1. 회원가입

User의 요청에 따라 사용자 계정을 생성한다.

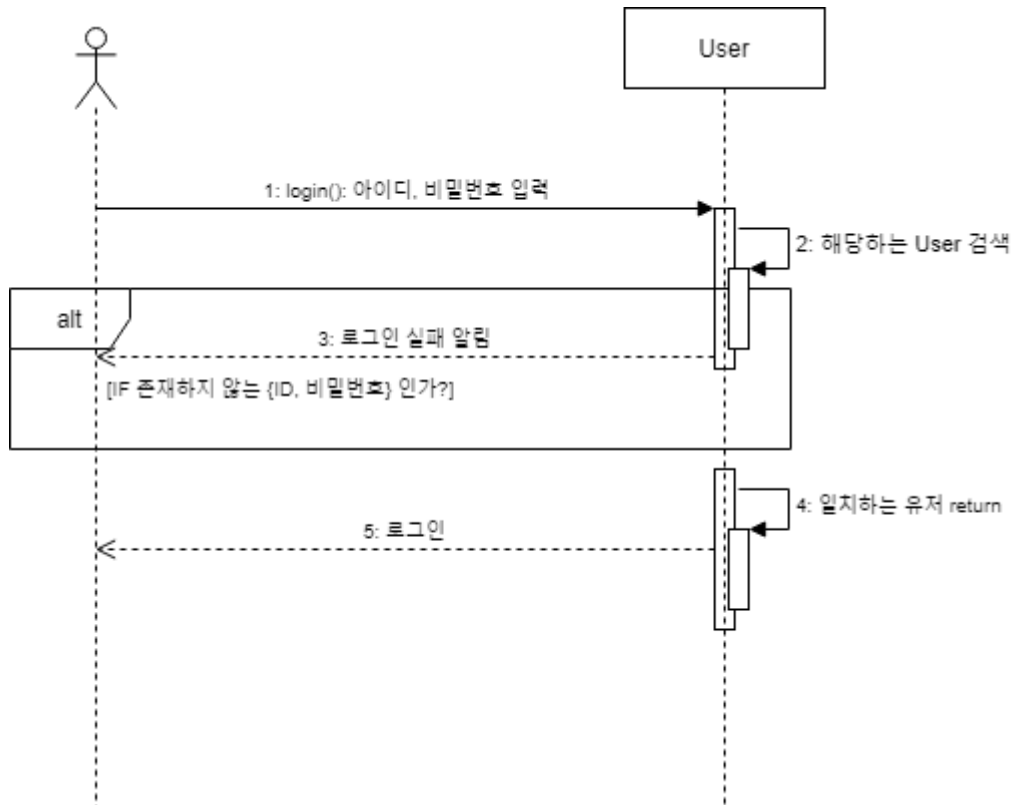


1. 사용자는 회원가입을 위해 ID를 입력한다.
2. 서버는 해당 ID에 대한 유효성을 검사한다.
3. 중복된 ID일 경우에는 중복되었음을 알린다.
4. 중복된 ID가 아닌 정상적인 ID를 다시 입력 받는다.
5. 정상적으로 ID를 입력했을 경우 이메일을 입력한다.

6. 입력된 이메일을 통해 인증번호를 전송한다.
7. 발급된 인증번호를 입력한다.
8. 잘못된 인증번호를 입력했을 경우 인증이 실패했음을 알린다.
9. 인증이 성공적으로 진행했을 경우 유저에게 알린다.
10. 이름, 전화번호, 비밀번호를 입력한다.
11. 서버에서 유효성을 검사한다.
12. 만약 유효하지 않은 정보들일 경우 유효하지 않음을 알린다.
13. 유효할 경우 User 데이터를 저장한다.

3.2. 로그인

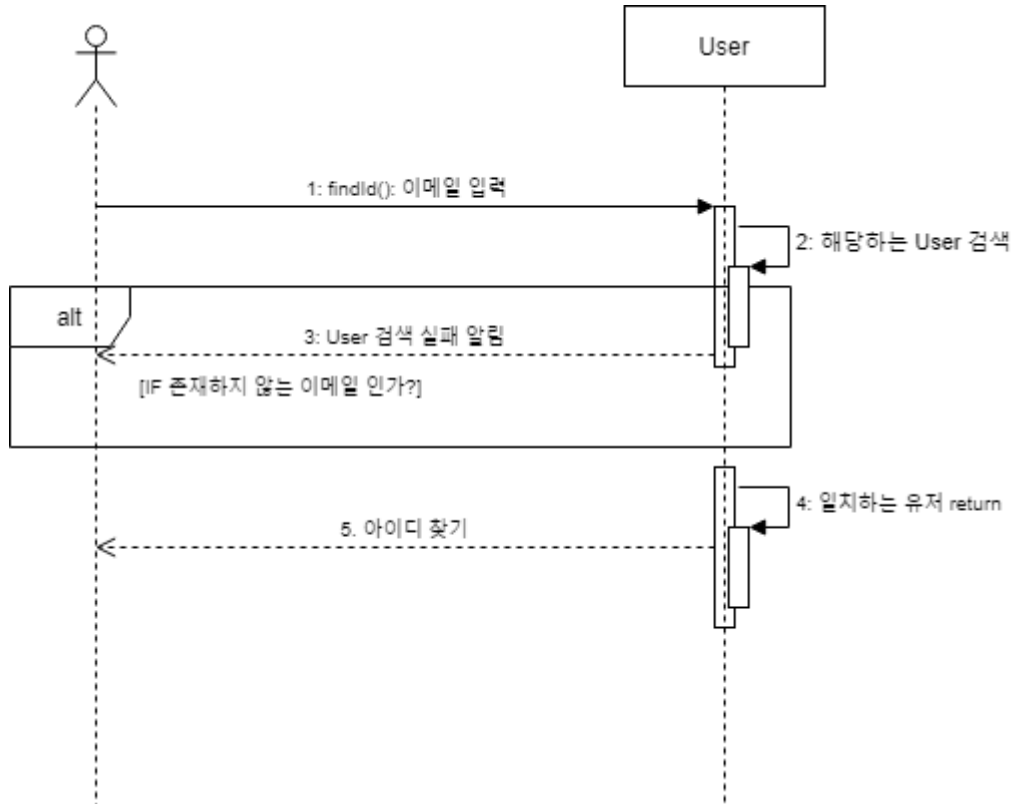
사전에 회원가입을 통해 요청된 User의 정보에 따라 인증된 사용자임을 구분한다.



1. 사용자는 로그인을 위해 ID, 비밀번호를 입력한다.
2. 서버는 해당 ID와 비밀번호에 대한 User를 탐색한다.
3. 존재하지 않는 ID, 비밀번호 쌍일 경우 로그인이 실패했음을 알린다.
4. 만약 일치하는 유저를 찾았을 경우 Return 한다.
5. 사용자에게 탐색한 User에 대한 데이터 및 인증 토큰을 보낸다.

3.3. 아이디 찾기

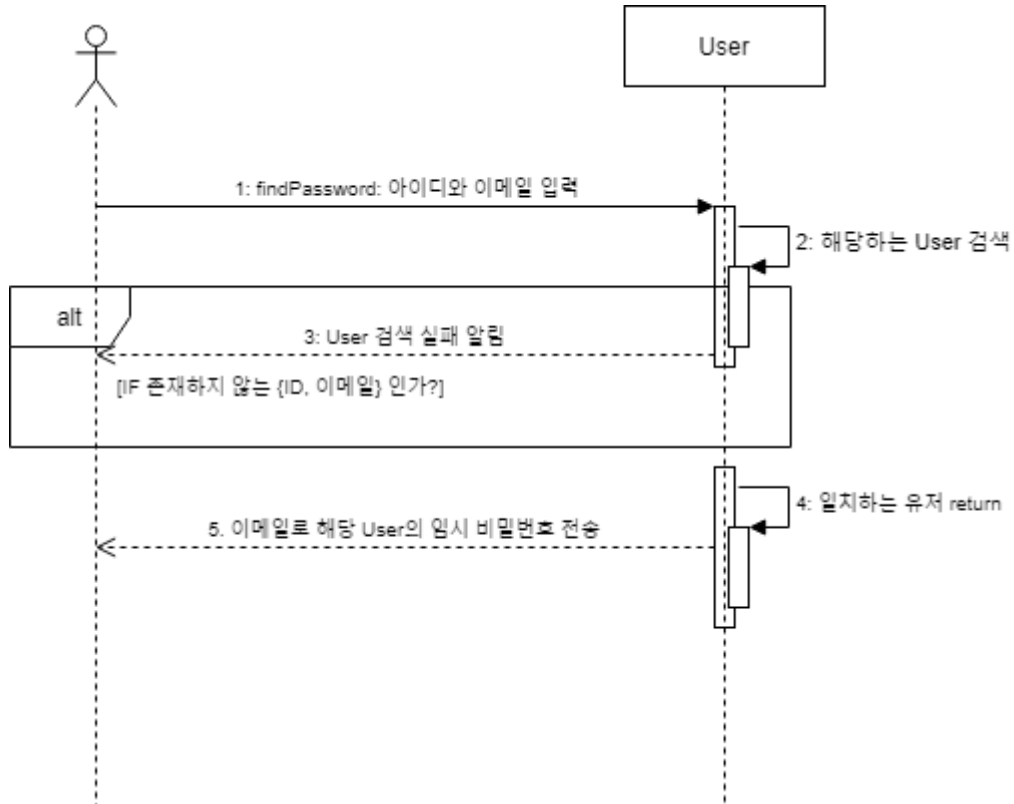
사전에 회원가입을 통해 요청된 User의 정보에 해당하는 아이디를 분실했을 때 찾을 수 있는 기능을 제공한다.



1. 사용자는 아이디 찾기를 위해 이메일을 입력한다.
2. 서버는 해당 아이디에 해당하는 User를 탐색한다.
3. 존재하지 않는 이메일일 경우 아이디 찾기를 실패 했음을 알린다.
4. 만약 일치하는 유저를 찾았을 경우 Return 한다.
5. 사용자에게 탐색한 User에 대한 ID 정보를 이메일로 보낸다.

3.4. 비밀번호 찾기

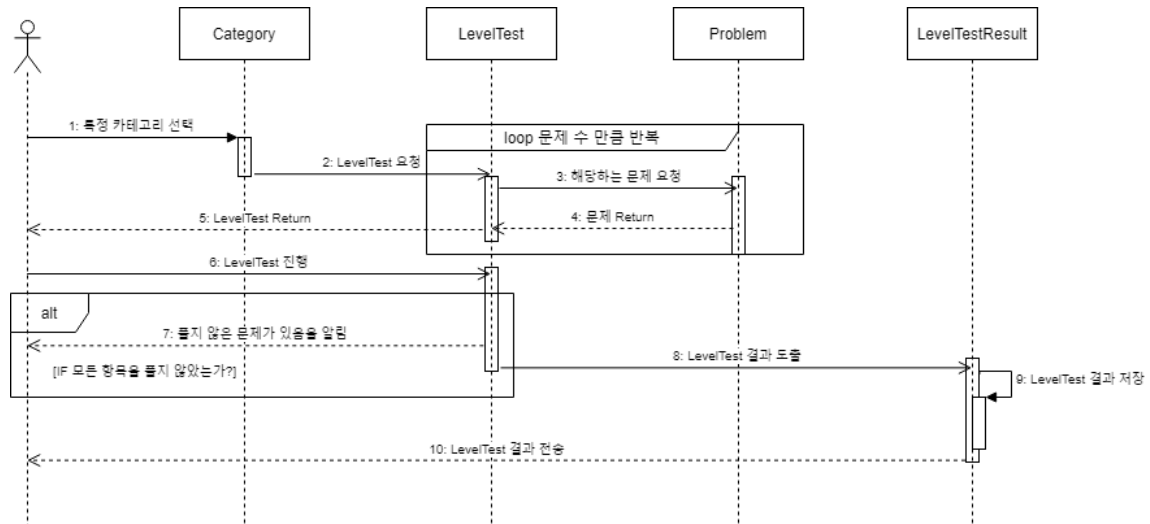
사전에 회원가입을 통해 요청된 User의 정보에 해당하는 비밀번호를 분실했을 때 찾을 수 있는 기능을 제공한다.



1. 사용자는 비밀번호 찾기를 위해 아이디와 이메일을 입력한다.
2. 서버는 해당 아이디와 이메일에 해당하는 User를 탐색한다.
3. 존재하지 않는 ID, 이메일일 경우 비밀번호 찾기를 실패 했음을 알린다.
4. 만약 일치하는 유저를 찾았을 경우 Return 한다.
5. 사용자에게 탐색한 User에 대한 임시 비밀번호 정보를 이메일로 보낸다.

3.5. 레벨 테스트

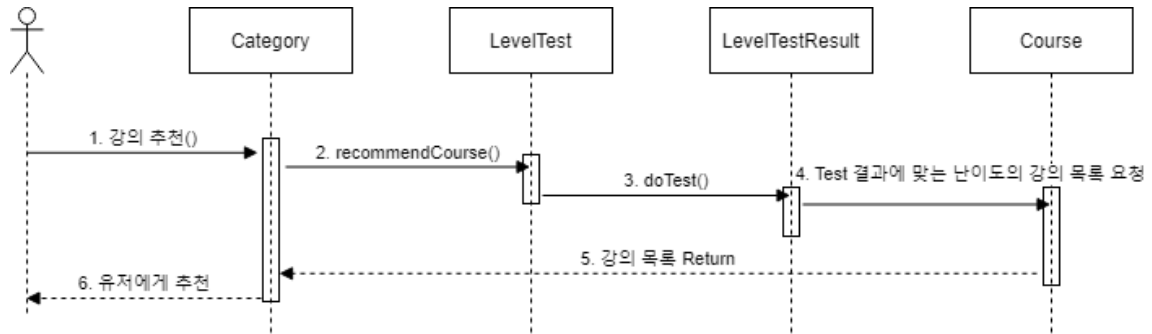
특정 카테고리에서 사용자의 레벨을 측정할 수 있다.



1. 특정 카테고리를 선택한다.
2. LevelTest를 요청한다.
3. 이 때 LevelTest는 Problem으로 이루어져있기 때문에 문제를 불러오기 위해 문항들을 요청한다.
4. 문제에 대한 문항을 Return한다.
5. 만들어진 LevelTest를 Actor에게 Return한다.
6. Actor가 Return된 LevelTest를 진행한다.
7. 만약 풀지 않은 문제가 있다면 이를 알린다.
8. LevelTest 결과를 도출하기 위해 LevelTestResult에게 요청한다.
9. LevelTestResult는 결과를 저장한다.
10. LevelTestResult는 User에게 도출된 결과를 Return한다.

3.6. 강의 추천

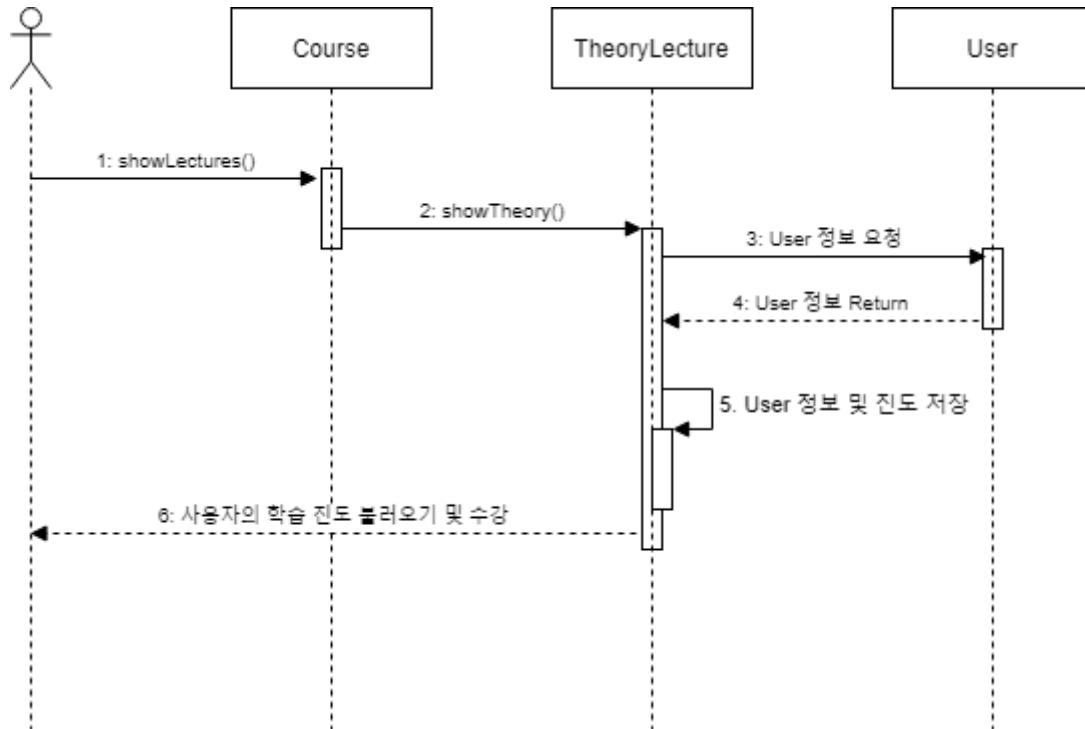
Level Test 결과를 바탕으로 사용자에게 적절한 강의를 추천할 수 있다.



1. 강의 추천을 요청한다.
2. Category 객체에서 recommendCourse()를 호출한다.
3. 강의를 추천하기 위해 레벨 테스트를 진행한다.
4. Test 결과에 맞는 난이도의 강의 목록을 도출하도록 요청한다.
5. 도출된 강의 목록을 Category에게 보낸다.
6. 사용자가 적절한 난이도의 강의 목록을 얻어낸다.

3.7. 이론 학습

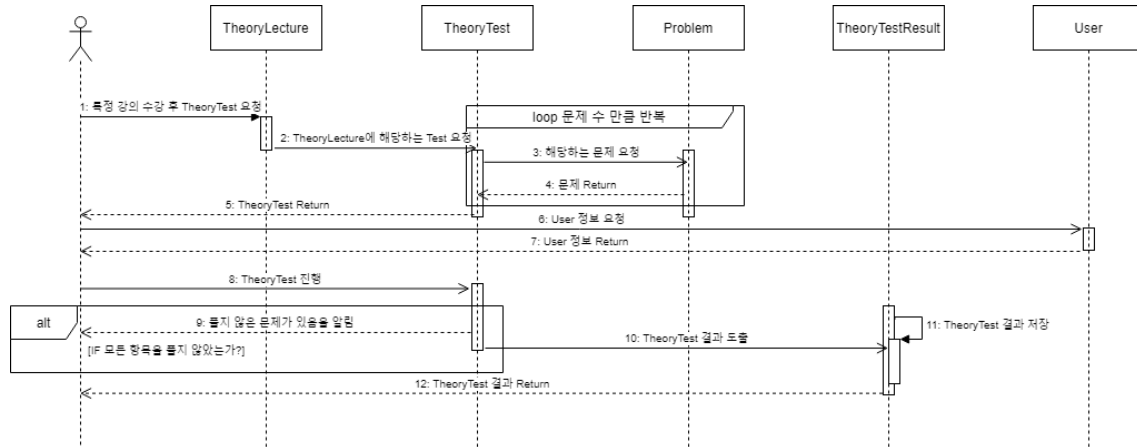
사용자는 이론 학습을 진행할 수 있다.



1. Course에게 강의가 무엇이 있는지 요청한다.
2. 특정 강의를 수강하기 위해 TheoryLecture로 부터 불러온다.
3. 이 때 사용자가 얼마나 수강했는지 User정보를 받아오기 위해 User 정보를 요청한다.
4. User 정보를 Return한다.
5. TheoryLecture는 받아온 User 정보와 진도를 저장한다.
6. 저장된 진도를 바탕으로 User에게 학습을 불러오고 수강하게 된다.

3.8. 복습 테스트

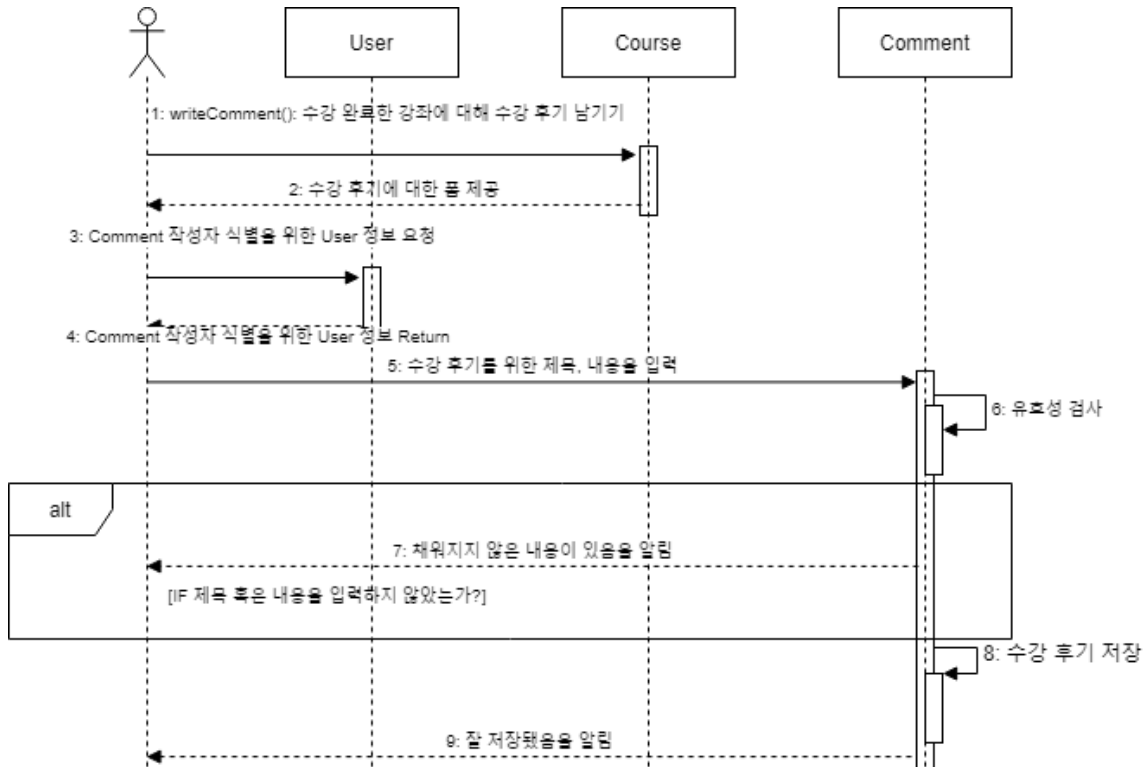
사용자는 이론 학습에 대한 복습 테스트를 진행할 수 있다.



1. 사용자는 학습 완료한 이론 강의에 대해 Test를 요청한다.
2. TheoryLecture에서 TheoryTest에게 해당하는 강의의 Test를 요청한다.
3. Test에 속한 문제를 불러오기 위해 Problem에 요청한다.
4. Problem을 Return한다.
5. 전달된 Test를 유저에게 Return한다..
6. 전달된 Test를 진행하기 전에 어떤 User인지 식별하기 위해 User에게 User 정보를 요청한다.
7. User 정보 Return
8. 사용자는 TheoryTest를 진행한다.
9. 만약 사용자가 제출하기 전 풀지 않은 문제가 있다면 알린다.
10. TheoryTest 결과를 도출하기 위해 TheoryTestResult에게 요청한다.
11. TheoryTestResult 결과를 저장한다.
12. TheoryTest 결과를 사용자에게 Return한다.

3.9. 강의 평가

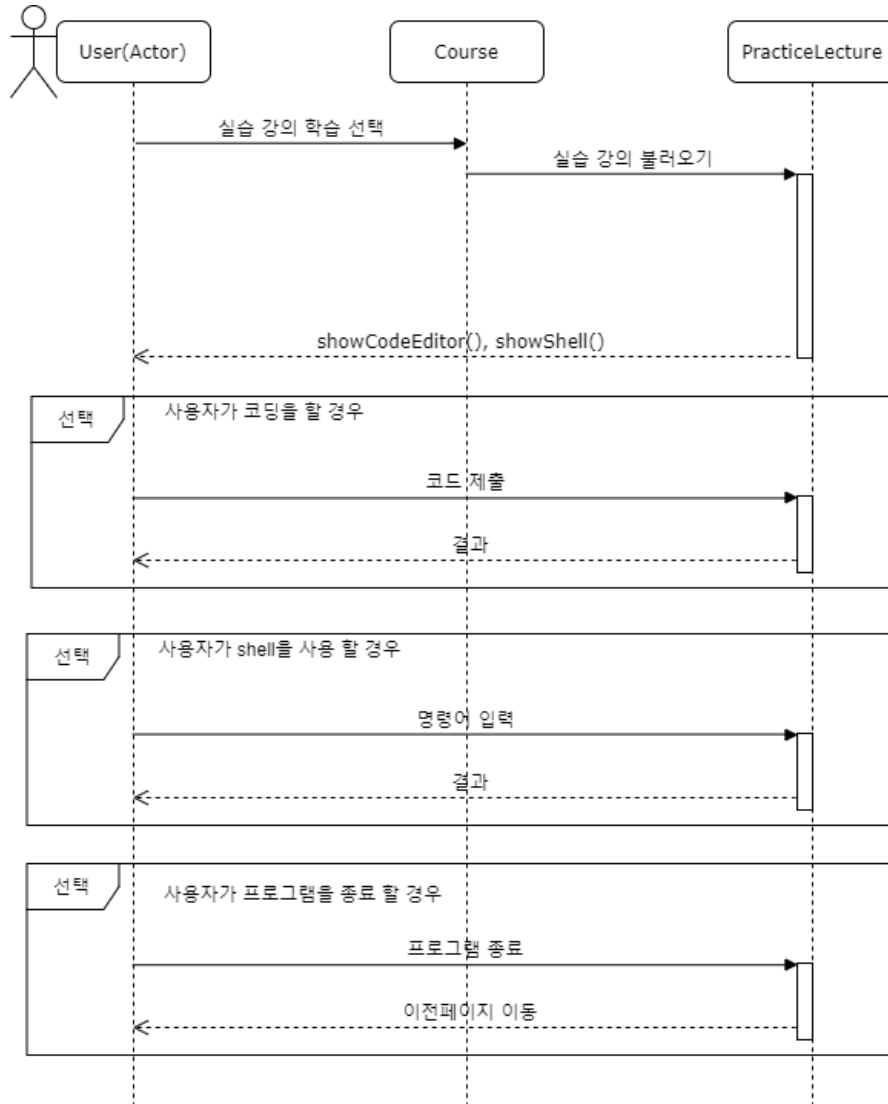
사용자는 학습 완료한 강좌에 대해 강의 평가를 진행할 수 있다.



1. 사용자는 수강 완료한 강좌에 대해 수강 후기를 남길 수 있다.
2. Course에서 수강 후기에 대한 form을 제공한다.
3. Comment 작성자를 식별하기 위해 User 정보를 요청한다.
4. User 정보 Return
5. 사용자는 수강 후기를 위한 제목, 내용을 입력한다.
6. 올바른 형식인지 유효성을 검사한다.
7. 만약 채워지지 않은 내용이 있다면 이를 알린다.
8. 올바른 형식이라면 사용자의 수강 후기를 Comment에 저장한다.
9. 잘 저장됐음을 사용자에게 알린다.

3.10. 실습 학습

‘실습 학습’은 사용자가 들은 강의에 대해 직접 실습을 하여 학습 할 수 있도록 돕는다.

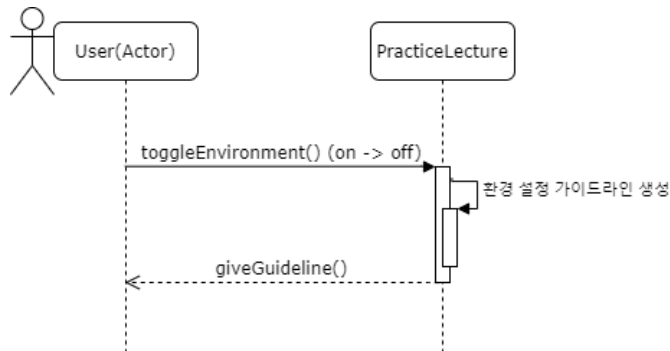


1. 사용자는 특정 과목에 대한 목차 페이지에서 “실습 강의 학습”을 선택한다.
 - 1.1 Course는 PracticeLecture에게 실습 강의를 불러올 것을 요청한다.
2. PracticeLecture는 실습 강의를 불러와서 코드 편집기와 shell을 사용자에게 보여준다. (showCodeEditor(), showShell()) 사용자는 코딩을 하거나 shell을 이용 할 수 있다.
 - 2.1. 사용자가 코딩을 할 경우
 - 2.1.1. 사용자는 실습에 맞는 코드를 작성 후 코드를 제출한다.
 - 2.1.2. PracticeLecture에서 코드의 결과를 사용자에게 보여준다.
 - 2.2. 사용자가 shell을 사용 할 경우

- 2.2.1. 사용자는 shell에 명령어를 입력한다.
- 2.2.2. PracticeLecture에서 입력한 명령어에 대한 결과를 사용자에게 보여준다.
- 2.3. 사용자가 프로그램을 종료 할 경우
 - 2.3.1. 사용자가 PracticeLecture 에게 프로그램 종료 요청을 한다.
 - 2.3.2. PracticeLecture는 실습 강의를 종료하고 이전 페이지로 이동하도록 한다.

3.11. 환경 가이드라인

‘환경 가이드라인’에서는 사용자가 자동 환경 설정에 대한 변수 ‘environmentOnOff’를 ‘off’ 상태로 하여 환경 설정에 대한 가이드라인을 제공 받는다.



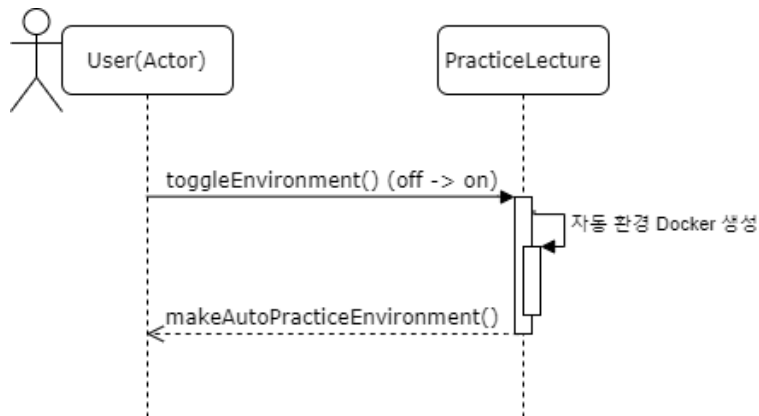
1. 사용자가 PracticeLecture에게 toggleEnvironment()을 요청하여 환경 설정에 대한 변수를 on에서 off로 바꿀 것을 요청한다.

1.1 PracticeLecture는 사용자가 진행하는 실습에 대한 환경 설정 가이드라인을 생성한다.

2. PracticeLecture는 giveGuideline()으로 가이드라인을 사용자에게 제공한다.

3.12. 자동 환경설정

‘자동 환경설정’에서는 사용자가 자동 환경 설정에 대한 변수 ‘environmentOnOff’를 ‘on’상태로 하여 실습에 필요한 환경을 제공받는다.



1. 사용자가 PracticeLecture에게 toggleEnvironment()을 요청하여 환경 설정에 대한 변수를 off에서 on로 바꿀 것을 요청한다.
 - 1.1 PracticeLecture는 사용자의 실습 환경에 맞는 Docker를 구축하여 자동 환경 생성을 한다.
2. PracticeLecture는 makeAutoPracticeEnvironment()로 사용자에게 자동 설정된 개발 환경을 제공한다.

1. 사용자는 강좌 업로드 버튼을 클릭하여 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 각 목차 별 제목을 입력한다.
5. 사용자는 각 목차 별 강좌 업로드를 요청한다.
- 5.1 사용자는 Lecture에 대한 객체들에게 강좌를 생성 할 것을 요청하는 데,
 - 5.1.1 Course는 실습 강좌를 생성 할 경우 makeLecture()를 PracticeLecture에게 요청한다.
 - 5.1.2 PracticeLecture는 목차 별 강좌를 생성한다.
 - 5.2.1 Course는 이론 강좌를 생성 할 경우 makeLecture()를 TheoryLecture에게 요청한다.
 - 5.2.1.1 TheoryLecture는 문서로 이루어진 강좌를 생성 할 경우 makeContextAsDocument()를 실행한다.
 - 5.2.1.2. TheoryLecture는 동영상 강좌를 생성 할 경우 makeContextAsVideo()를 실행한다.
 - 5.2.2 TheoryLecture는 목차 별 강좌 생성을 성공적으로 마친다.
6. Course는 목차 별 강좌 생성을 성공적으로 마친다
7. 사용자는 TheoryLecture에게 각 목차 별 테스트 업로드를 요청한다.
 - 7.1 TheoryLecture는 TheoryTest에게 makeTest()로 테스트를 생성 할 것을 요청한다.
 - 7.1.1 TheoryTest는 Problem에게 makeProblem()으로 문제를 생성 할 것을 요청한다.
 - 7.1.2. Problem은 사용자에게 문제 하나 당 문제, 답, 점수를 요청한다.
 - 7.1.3. 사용자는 문제, 답, 점수를 입력한다.
 - 7.1.4. Problem은 문제를 생성한다.
 - 7.1.5 7.1.1부터 7.1.4를 문제의 수 만큼 반복한다.
 - 7.2. TheoryTest는 문제들을 모아 테스트 생성을 완료한다.
8. TheoryLecture는 목차 별 테스트 생성을 완료한다.
9. 4부터 8을 강좌의 수 만큼 반복한다.
10. 사용자는 Course에게 카테고리를 요청한다.
 - 10.1. Course는 Category에게 카테고리를 요청한다.
 - 10.2 Category는 Course에게 카테고리를 제공한다.
11. Course는 사용자에게 카테고리를 제공한다.
12. 사용자는 제공된 카테고리 중에서 카테고리를 설정할 것을 Course에게 요청한다.
13. Course는 카테고리 설정을 완료한다.
14. 사용자는 Course에게 썸네일을 설정 할 것을 요청한다.
15. Course는 썸네일 설정을 완료한다.
 - 15.1 사용자가 썸네일 설정을 하지 않을 경우 Course는 기본 썸네일 설정을 완료한다.
16. 사용자는 자신이 업로드하려는 강좌에 대한 설정을 끝냈을 경우 제출 버튼을 클릭한다.
 - 16.1. Course는 User에게 사용자의 정보를 요청한다.
 - 16.2. User는 Course에게 사용자의 정보를 제공한다.

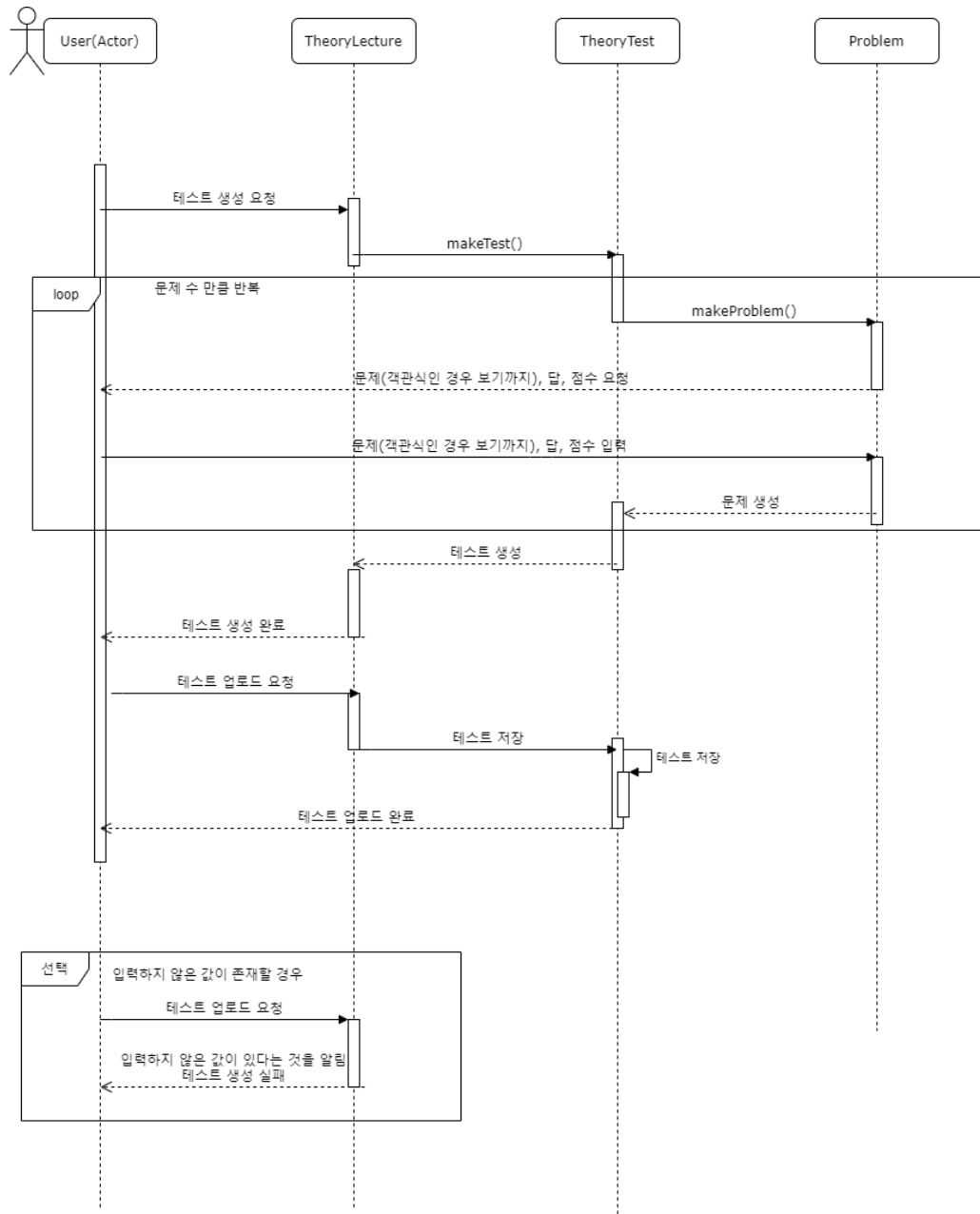
16.3. Course는 사용자가 생성한 강좌와 사용자의 정보를 이용하여 강좌를 생성한다.

17. Course는 강좌 업로드를 완료한다.

17.1. 사용자가 썸네일 설정을 제외하고 입력하지 않은 값이 하나라도 있을 경우 Course는 강좌 업로드를 실패한다.

3.14. 테스트 생성

‘테스트 생성’은 사용자가 자신이 만든 이론 강의에 대해 테스트를 생성하는 것이다.



1.사용자는 TheoryLecture에게 테스트 생성을 요청한다.

1.1 TheoryLecture는 TheoryTest에게 makeTest()로 테스트를 생성할 것을 요청한다.

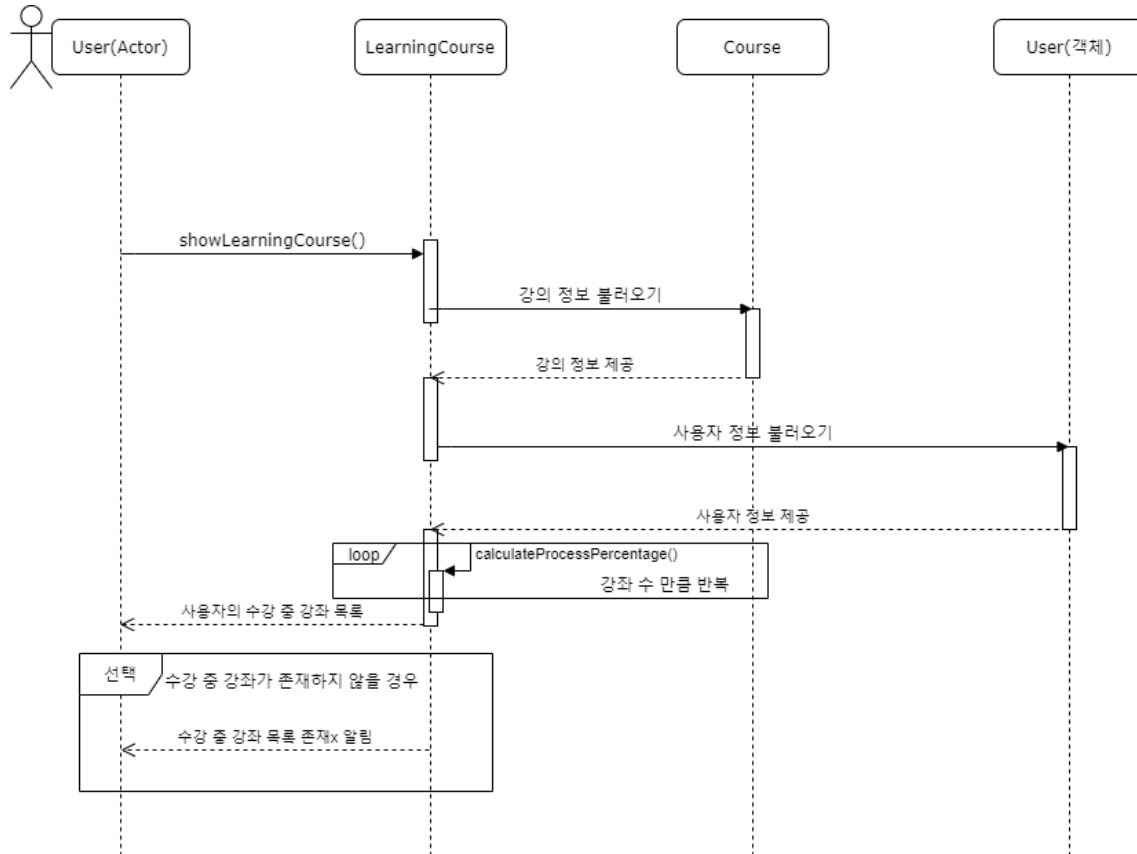
1.1.1 TheoryTest는 Problem에게 makeProblem()으로 문제를 생성할 것을 요청한다.

1.1.2. Problem은 사용자에게 문제 하나 당 문제, 답, 점수를 요청한다.

- 1.1.3. 사용자는 문제, 답, 점수를 입력한다.
- 1.1.4. Problem은 문제를 생성한다.
- 1.1.5 1.1.1부터 1.1.4를 문제의 수만큼 반복한다.
- 1.2. TheoryTest는 문제들을 모아 테스트 생성을 완료한다.
- 2. TheoryLecture는 테스트 생성을 완료한다.
- 3. 사용자는 TheoryLecture에게 테스트 업로드 요청을 한다.
- 4. TheoryLecture는 테스트 업로드를 완료하며, TheoryTest에게 테스트 저장을 요청한다.
 - 4.1. 사용자가 입력하지 않은 값이 존재할 경우 TheoryLecture는 테스트 업로드를 완료하지 않는다.
- 5. TheoryTest는 테스트를 저장한다.

3.15. 수강 중 강좌 목록 조회

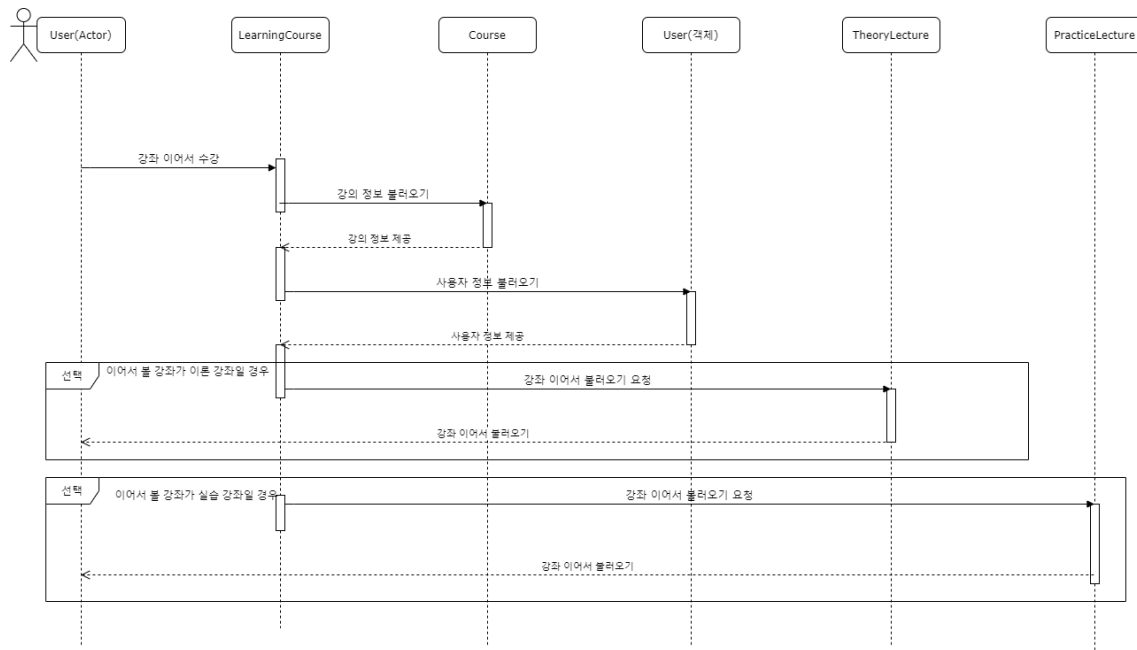
‘수강 중 강의 목록 조회’는 사용자가 현재 수강 중인 강좌들의 목록을 조회하는 것이다.



1. 사용자는 마이페이지에서 '내 강좌'를 선택한 뒤 '수강 중인 강좌'를 선택하여 LearningCourse에게 `showLearningCourse()`를 요청한다.
 - 1.1. LearningCourse는 Course에게 강의 정보를 불러올 것을 요청한다.
 - 1.2. Course는 LearningCourse에게 강의 정보를 제공한다.
 - 1.3. LearningCourse는 User에게 사용자의 정보를 불러올 것을 요청한다.
 - 1.4. User는 LearningCourse에게 사용자의 정보를 제공한다.
 - 1.5. LearningCourse는 사용자가 수강한 강좌들에 대한 진도율을 보여주기 위해 `calculateProcessPercentage()`를 강좌 수 만큼 실행한다.
2. 사용자의 수강 중 강좌 목록을 진도율을 포함하여 보여준다.
 - 2.1. 사용자의 수강 중 강좌가 존재하지 않을 경우 수강 중 강좌 목록이 존재하지 않는다는 것을 알린다.

3.16. 강좌 이어서 수강

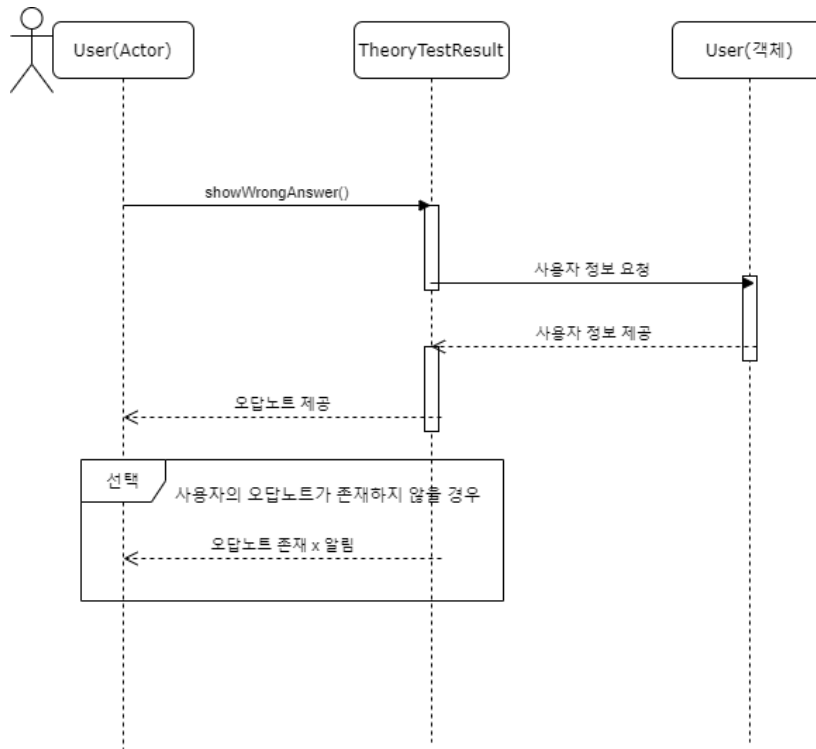
‘강좌 이어서 수강’은 사용자가 수강 중인 강좌들 중 특정 강좌에 대해 사용자가 제일 최근 수강하였던 지점부터 이어서 수강을 하는 것이다.



1. 사용자는 수강 중인 강좌 페이지에서 특정 강좌를 이어서 수강할 것을 요청한다.
 - 1.1. LearningCourse는 Course에게 강의 정보를 불러올 것을 요청한다.
 - 1.2. Course는 LearningCourse에게 강의 정보를 제공한다.
 - 1.3. LearningCourse는 User에게 사용자의 정보를 불러올 것을 요청한다.
 - 1.4. User는 LearningCourse에게 사용자의 정보를 제공한다.
 - 1.5. LearningCourse는 Lecture 관련 객체에게 강좌를 이어서 불러올 것을 요청하는 데,
 - 1.5.1.1. 이어서 볼 강좌가 이론 강좌일 경우 TheoryLecture 객체에게 요청한다.
 - 1.5.1.2. TheoryLecture는 사용자에게 강의를 이어서 불러온다.
 - 1.5.2.1. 이어서 볼 강좌가 실습 강좌일 경우 PracticeLecture 객체에게 요청한다.
 - 1.5.2.2. PracticeLecture는 사용자에게 강의를 이어서 불러온다.

3.17. 오답노트 확인

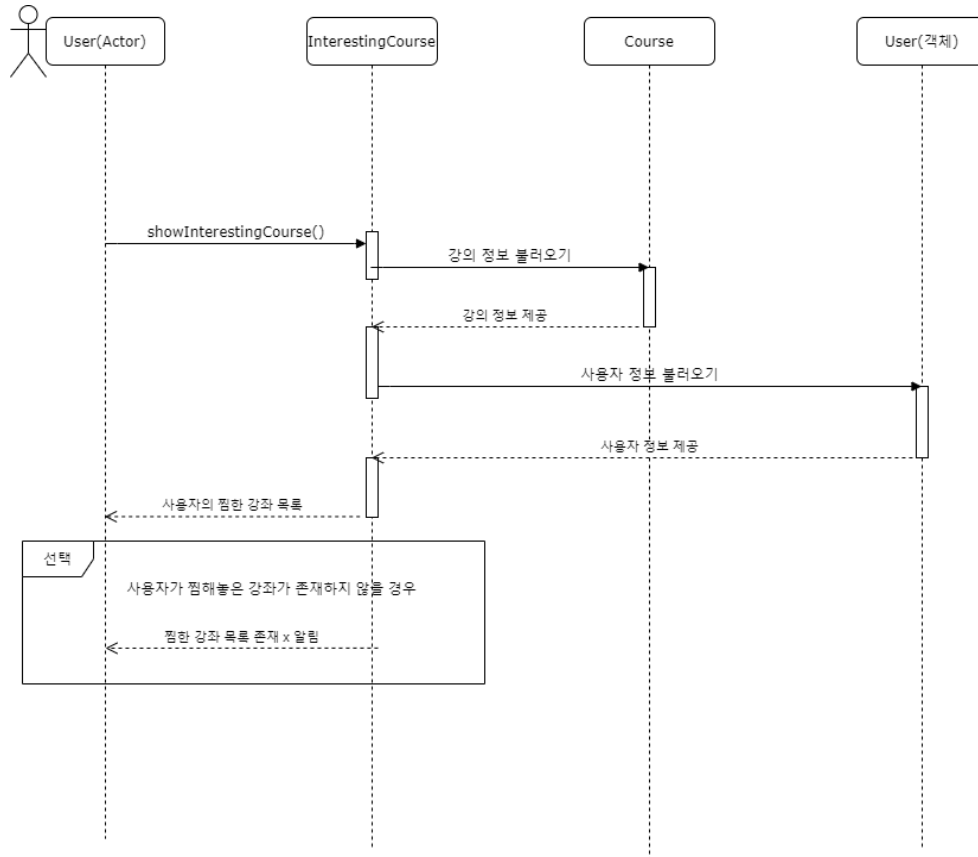
‘오답노트 확인’은 사용자가 이론 테스트를 한 후 사용자가 틀렸던 문제와 그 문제의 답을 오답노트 형식으로 보여준다.



1. 수강 중인 강좌 페이지에서 "오답 노트" 버튼을 눌러 TheoryTestResult에게 showWrongAnswer()을 요청한다.
 - 1.1. TheoryTestResult에서 User에게 사용자 정보를 요청한다.
 - 1.2. User는 TheoryTestResult에게 사용자 정보를 제공한다.
2. TheoryTestResult는 사용자에게 오답노트를 제공한다.
 - 2.1. 사용자의 오답노트가 존재하지 않을 경우 오답노트가 존재하지 않다는 것을 알린다.

3.18. 찜한 강좌 조회

‘찜한 강좌 조회’는 사용자가 찜한 강좌로 넣었던 강좌들의 목록을 조회하는 것이다.



1. 사용자는 마이페이지의 내 강좌 페이지에서 “찜한 강좌”를 선택하여 InterestingCourse에게 `showInterestingCourse()`를 요청한다.

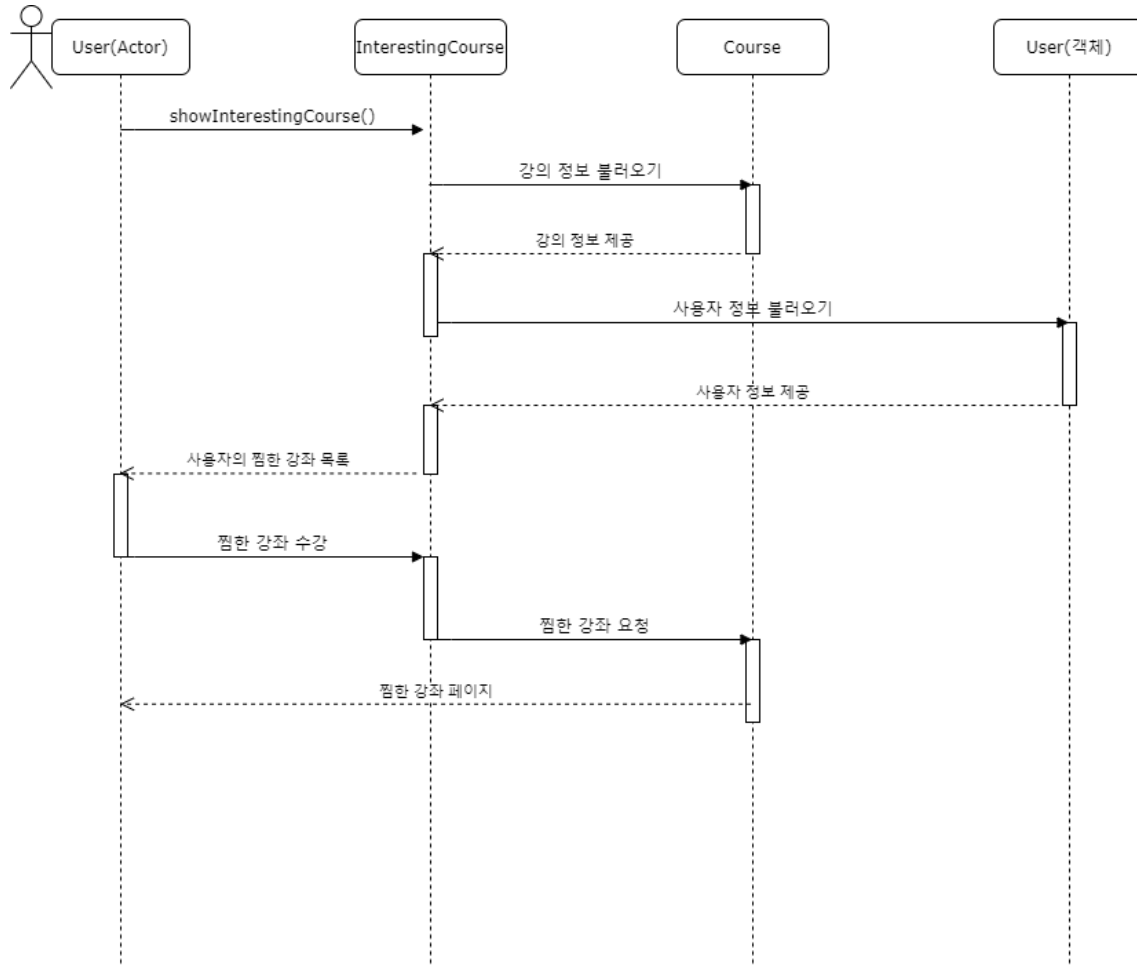
- 1.1. InterestingCourse는 Course에게 강의 정보를 불러올 것을 요청한다.
- 1.2. Course는 InterestingCourse에게 강의 정보를 제공한다.
- 1.3. InterestingCourse는 User에게 사용자의 정보를 불러올 것을 요청한다.
- 1.4. User는 InterestingCourse에게 사용자의 정보를 제공한다.

2. InterestingCourse는 사용자에게 사용자의 찜한 강좌 목록을 제공한다.

- 2.1. 사용자가 찜해놓은 강좌가 존재하지 않을 경우 찜한 강좌가 존재하지 않는다는 것을 알린다.

3.19. 찜한강좌 수강

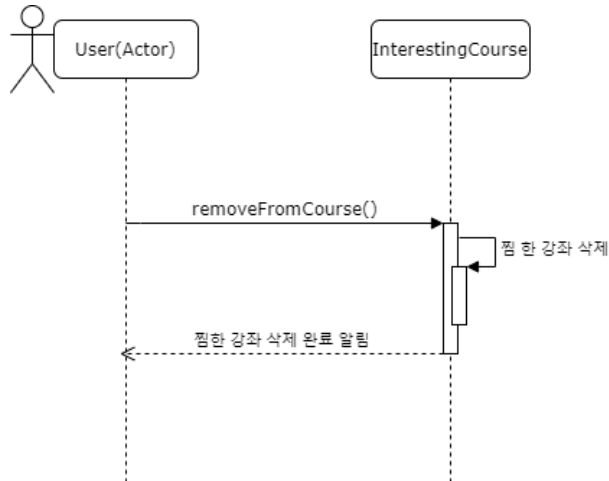
‘찜한 강좌 수강’은 사용자가 찜한 강좌들 중 특정 강좌를 수강하도록 한다.



1. 사용자는 InterestingCourse에게 showInterestingCourse()를 요청한다.
 - 1.1. InterestingCourse는 Course에게 강의 정보를 불러올 것을 요청한다.
 - 1.2. Course는 InterestingCourse에게 강의 정보를 제공한다.
 - 1.3. InterestingCourse는 User에게 사용자의 정보를 불러올 것을 요청한다.
 - 1.4. User는 InterestingCourse에게 사용자의 정보를 제공한다.
2. InterestingCourse는 사용자에게 사용자의 찜한 강좌 목록을 제공한다.
3. 사용자는 InterestingCourse에게 찜한 강좌 수강을 요청한다.
 - 3.1. InterestingCourse는 Course에게 사용자가 찜한 강좌를 요청한다.
4. Course는 사용자에게 찜한 강좌 페이지를 제공한다.

3.20. 찜한 강좌 삭제

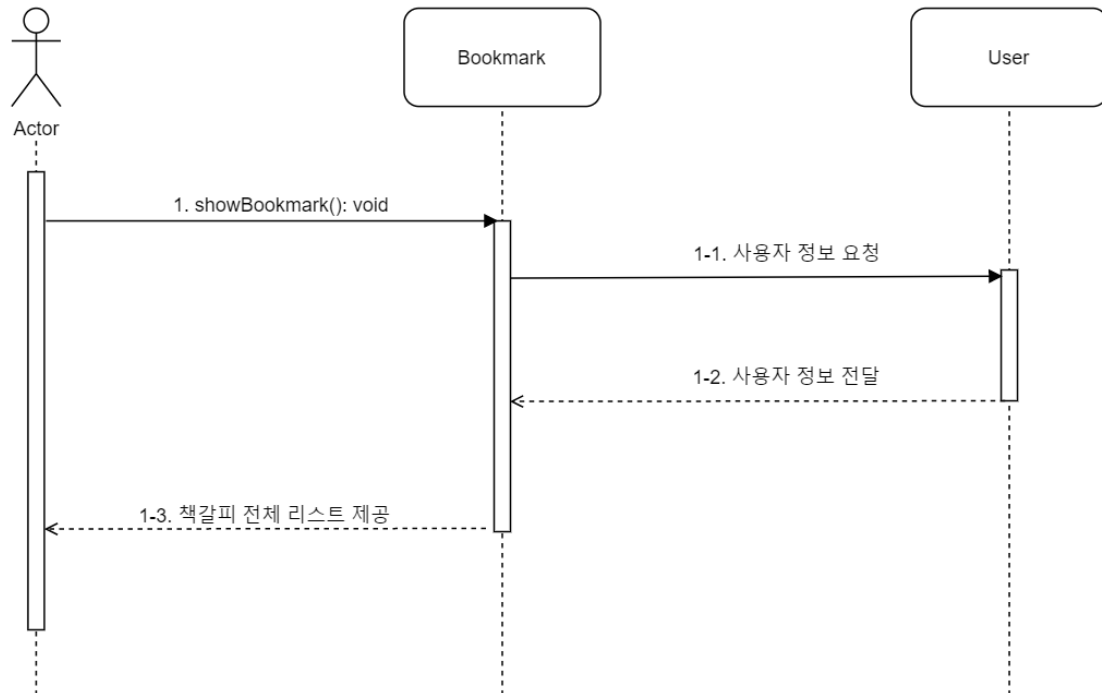
‘찜한 강좌 삭제’는 사용자가 찜한 강좌들 중 특정 강좌를 “찜한 강좌” 목록에서 삭제한다.



1. 사용자는 "찜한 강좌" 페이지에서 찜 취소 버튼을 누름으로써 InterestingCourse에게 `removeFromCourse()`를 요청한다.
2. InterestingCourse는 사용자가 찜한 강좌 중 사용자가 삭제하고자 하는 강좌를 삭제한다.
3. InterestingCourse는 찜한 강좌 삭제를 완료한다.

3.21. 강좌 책갈피 조회

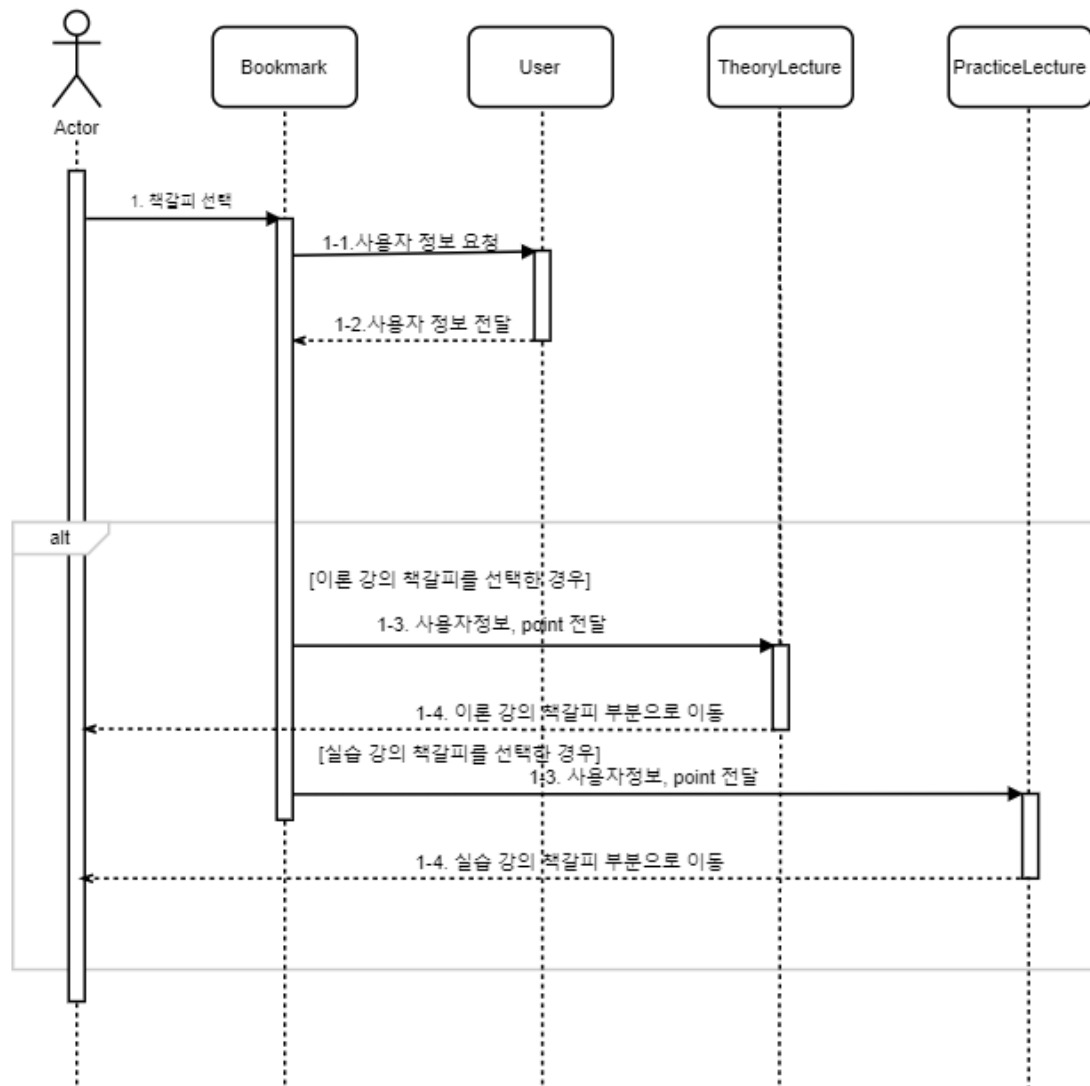
강좌 책갈피 조회에서는 사용자가 현재 강의 별로 등록해 놓은 책갈피 목록을 조회한다.



1. 사용자가 Bookmark에게 showBookmark()를 실행한다.
 - 1.1 Bookmark는 User에게 현재 사용자 정보를 요청한다.
 - 1.2 User는 Bookmark에게 현재 사용자 정보를 제공한다.
 - 1.3 Bookmark는 현재 사용자에게 해당하는 책갈피 리스트를 불러오고 사용자는 이를 확인가능하다.

3.22. 강의 책갈피 사용

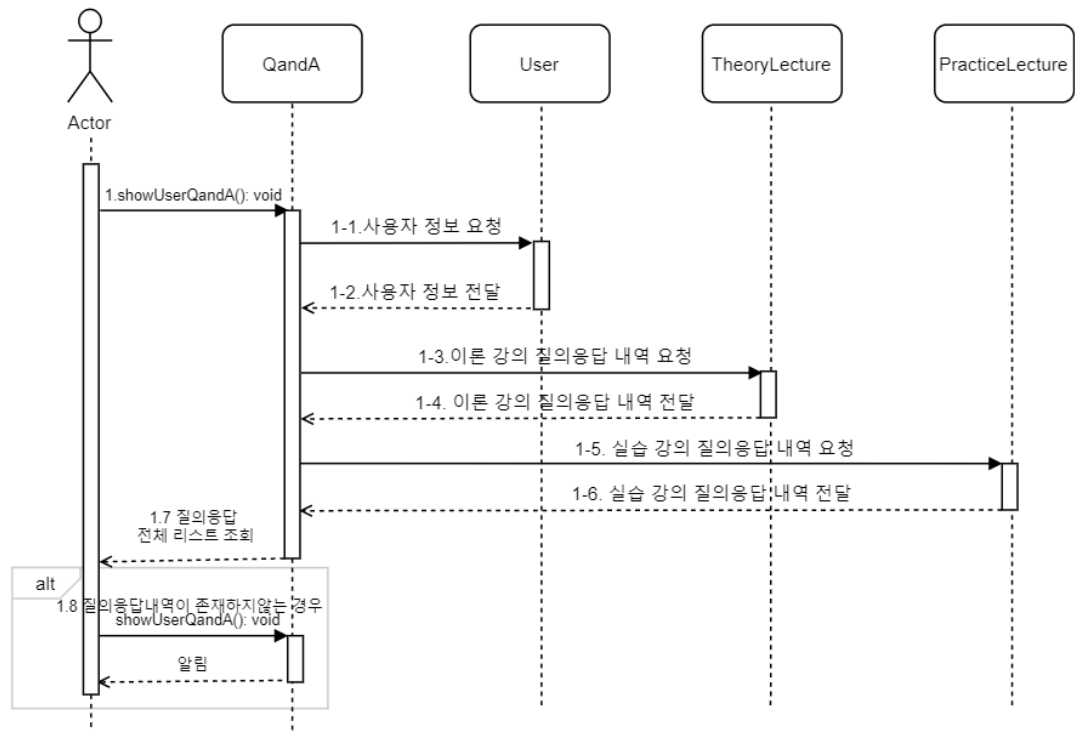
강좌 책갈피 사용에서는 사용자의 특정 책갈피 부분부터 강의 학습을 가능하게 한다.



1. 사용자가 책갈피 리스트에서 특정 책갈피를 선택한다.
 - 1-1. Bookmark에서 User에게 사용자의 강좌정보를 요청한다.
 - 1-2. Bookmark에서 User에게 사용자의 강좌정보를 전달받는다.
 - 1-3. 사용자가 선택한 책갈피가 이론 강의일 경우 TheoryLecture에게, 실습강의인 경우 PracticeLecture에게 각각 사용자의 강좌정보와 책갈피 정보를 전달한다.
 - 1-4. 사용자는 선택한 책갈피 부분부터 강의수강이 가능하다.

3.23. Q&A 내역 조회

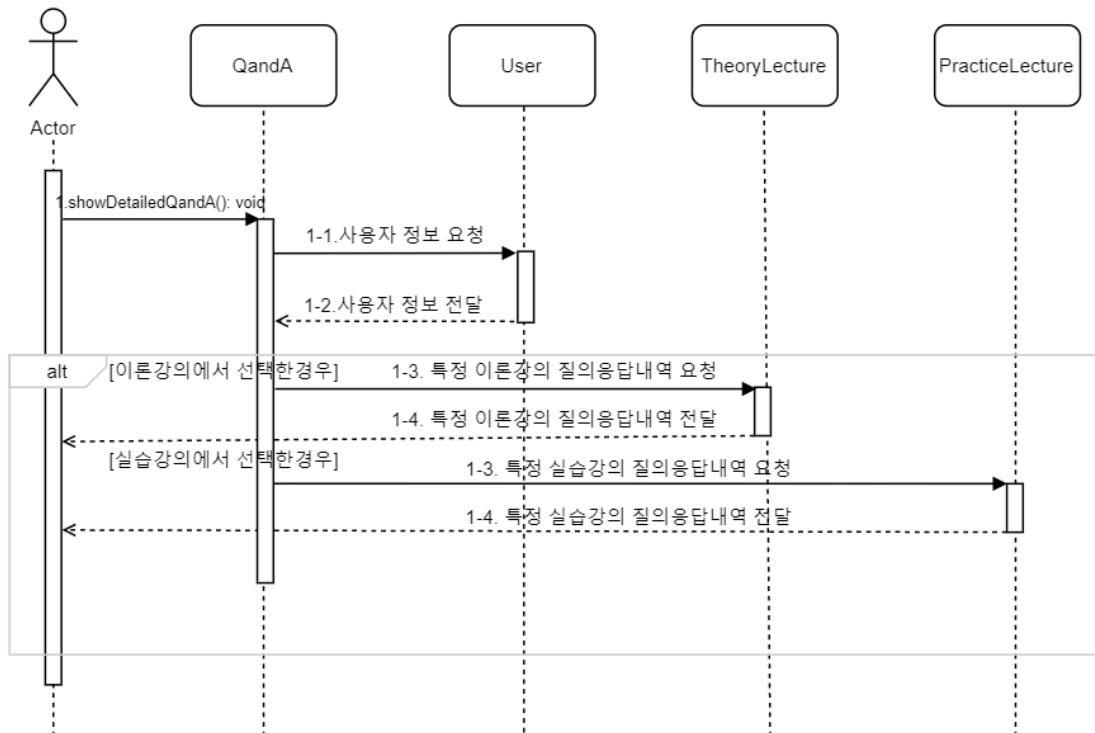
Q&A 내역 조회에서는 사용자가 강의를 수강하면서 작성한 전체 Q&A 내역을 조회할 수 있다.



1. 사용자가 QandA에게 showUserQandA()를 실행한다.
 - 1.1 QandA는 User에게 현재 사용자 정보를 요청한다.
 - 1.2 User는 QandA에게 현재 사용자 정보를 제공한다.
 - 1.3 QandA는 TheoryLecture에게 사용자의 이론 강의 질의응답내역을 요청한다.
 - 1.4 TheoryLecture는 QandA에게 사용자의 이론 강의 질의응답내역을 전달한다.
 - 1.5 QandA는 PracticeLecture에게 사용자의 실습 강의 질의응답내역을 요청한다.
 - 1.6 PracticeLecture는 QandA에게 사용자의 실습 강의 질의응답내역을 전달한다.
 - 1.7 QandA는 현재 사용자에게 해당하는 질의응답 리스트를 불러오고 사용자는 이를 확인가능하다.
 - 1.8 사용자에게 대한 질의 응답 내역이 존재하지 않는 경우 존재하지 않음을 알려준다.

3.24. QandA 내역 상세 조회

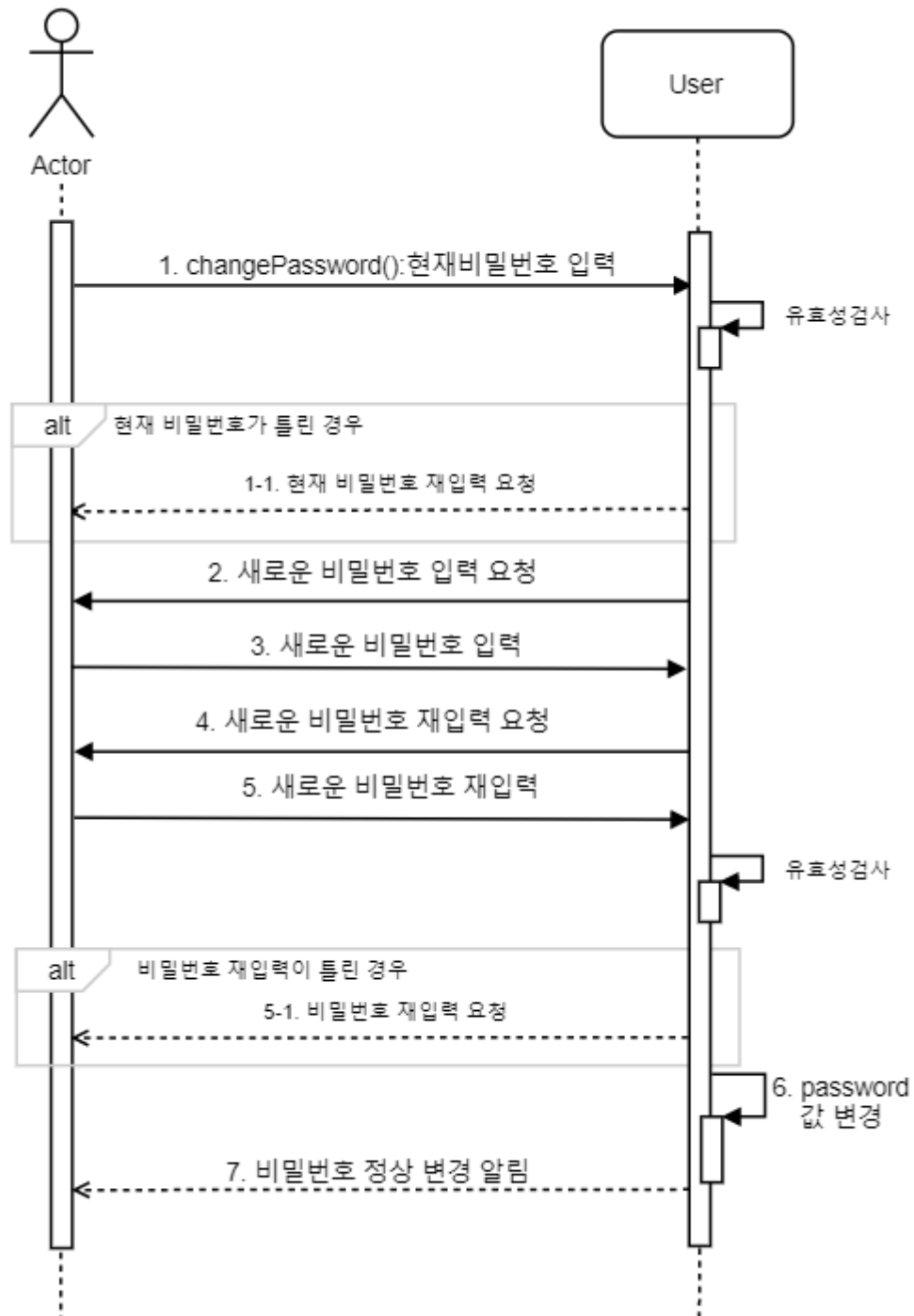
Q&A 내역 상세 조회에서는 사용자가 강의를 수강하면서 작성한 특정 Q&A내역 상세 내용을 조회할 수 있다.



1. 사용자가 QandA에게 showDetailedQandA()를 실행한다.
 - 1-1. QandA에서 User에게 사용자 정보를 요청한다.
 - 1-2. QandA에서 User에게 사용자 정보를 전달받는다.
 - 1-3. 사용자가 선택한 질의응답 내역이 이론 강의일 경우 TheoryLecture에게, 실습 강의인 경우 PracticeLecture에게 각각 질의응답 내역을 요청한다.
 - 1-4. 사용자는 선택한 질의응답 내역을 상세히 조회할 수 있다.

3.25. 비밀번호 변경

비밀번호 변경에서는 사용자가 로그인시 사용하는 비밀번호를 변경할 수 있다.



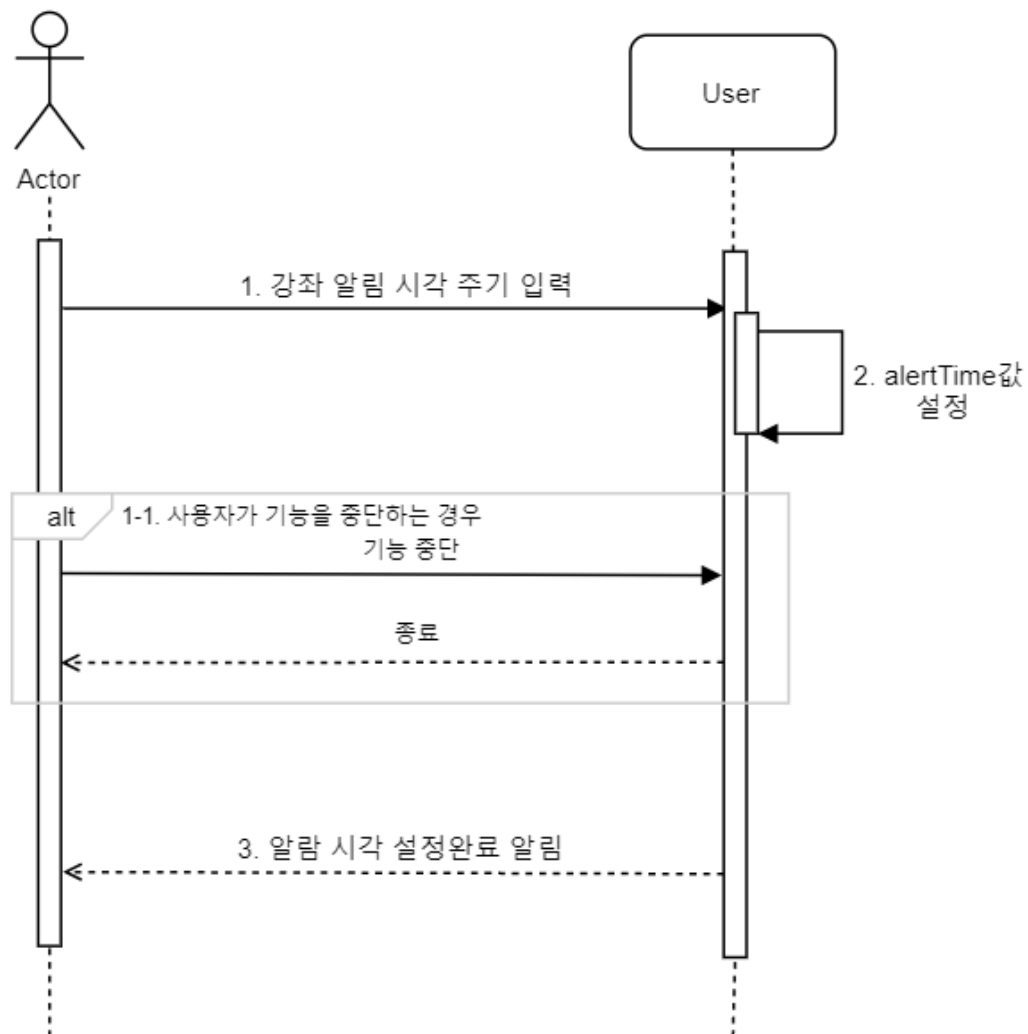
1. 사용자는 User에게 현재 비밀번호를 입력하여 `changePassword()`를 요청한다.
- 1-1. 만약 입력한 비밀번호가 현재 비밀번호와 일치하지 않는 경우 인증실패를 알리

고 다시 입력할 것을 요청한다.

2. User는 사용자에게 새로운 비밀번호를 입력을 요청한다.
3. 사용자는 새로운 비밀번호를 입력한다.
4. User는 사용자에게 새로운 비밀번호를 재입력을 요청한다.
5. 사용자는 3에서 입력하였던 새로운 비밀번호를 다시 한번 입력한다.
 - 5-1. 다시 입력한 비밀번호가 3과 일치하지 않는 경우 이를 알리고 다시 입력할 것을 요청한다.
6. User는 입력된 비밀번호로 새롭게 password의 값을 설정한다.
7. 사용자는 비밀번호가 정상적으로 변경되었음을 알 수 있다.

3.26. 강좌 알림 관리

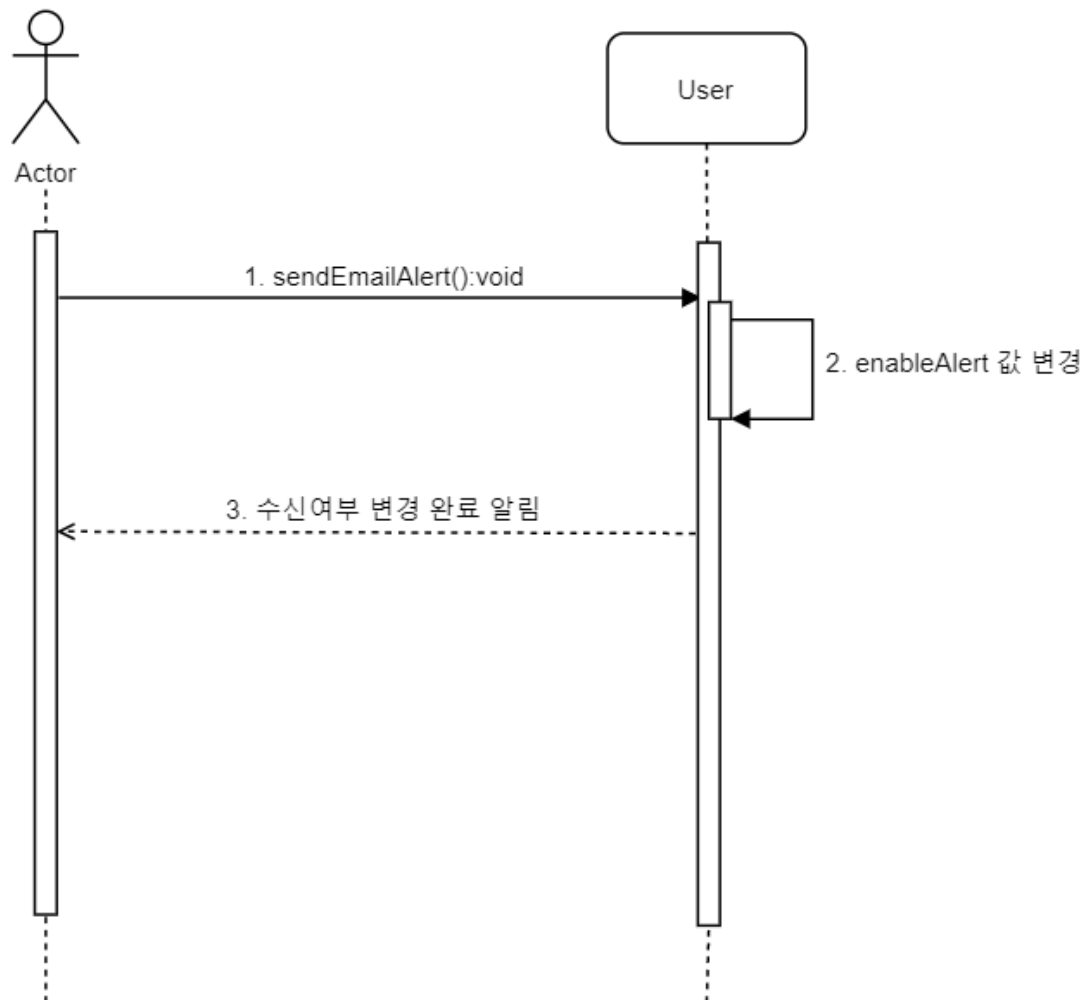
강좌 알림 관리에서는 사용자가 강좌 수강에 대한 알림을 메일로 받을 시, 알림 시간을 설정할 수 있다.



1. 사용자는 언제 어느 시간 주기로 알림을 받을 것인지 선택하여 User에게 전달한다.
 - 1-1. 사용자가 강좌 알림 시각 변경을 중단할 경우 기능이 종료된다.
2. User는 새로운 강좌 알림 시각을 설정한다.
3. 사용자는 강좌 알림 시각이 정상적으로 설정되었음을 확인한다.

3.27. 이메일 수신 관리

이메일 수신 관리에서는 사용자가 수강하는 강좌에 대한 알림을 메일로 받을지에 대한 여부를 설정할 수 있다.



1. 사용자는 `sendEmailAlert()`를 요청한다.
2. User는 이메일 수신 여부 값을 변경한다.
3. 사용자는 이메일 수신여부가 정상적으로 변경되었음을 확인한다.