



SKKU:MEET

Revolutionary Online Education Platform for School using Real-Time Eye Tracking Software

Software Test Plan Document

2021.05.28.

Introduction to Software Engineering 41

TEAM 9

Team Leader 이보현
Team Member 강병남
Team Member 이재훈
Team Member 임승재
Team Member 조효정

CONTENTS

1. Introduction	6
1.1. Purpose	6
1.2. Scope	6
1.3. Definitions, Acronyms, and Abbreviation	6
1.4. References	8
1.5. Overview	8
2. Approach	9
2.1. Test Method	9
2.1.1. Software Unit Test Methods	9
2.1.1.1. Backend	9
2.1.1.2. Frontend	11
2.1.2. Software Interface Test Methods	19
2.1.2.1. HTTP Server test methods	19
2.1.2.2. Database and Hardware test methods	19
2.1.2.3. Eye tracking API test methods	20
3. Software Unit Test	24
3.1. Backend	24
3.1.1. Live Lecture Network System	24
3.1.1.1. Live Lecture Data Storage test case	24
3.1.1.2. Live Lecture Data Storage test case - Negative	25
3.1.2. Real-time Survey System	25
3.1.2.1. Real-time Survey Creation test case	25
3.1.2.2. Real-time Survey Creation test case - Negative	26
3.1.2.3. Real-time Survey Creation test case	26
3.1.2.4. Real-time Survey Creation test case - Negative	26
3.1.2.5. Receive Real-time Survey test case	27
3.1.2.6. Real-time Survey Creation test case - Negative	27
3.1.3. Class Management System	27
3.1.3.1. Lecture Information Integration test case	28
3.1.3.2. Lecture Information Integration test case - Negative	28
3.2. Frontend	28
3.2.1. Profile	28
3.2.1.1. Class Diagram	29
3.2.1.2. Test Case	29
3.2.2. Score	32
3.2.2.1. Class Diagram	32
3.2.2.2. Test Case	33
3.2.3. Meeting Room	35
3.2.3.1. Class Diagram	35
3.2.3.2. Test Case	35
3.2.4. Lecture	39
3.2.4.1. Class Diagram	39

3.2.4.2. Test Case	39
3.2.5. Survey Result	43
3.2.5.1. Class Diagram	44
3.2.5.2. Test Case	44
4. Software Interface Test	46
4.1. HTTP Server	46
4.1.1. Valid register request	46
4.1.2. Inalid register request	46
4.1.3. Valid sign in request	46
4.1.4. Invalid sign in request	47
4.1.5. Valid view class list request	47
4.1.6. Invalid view class list request	48
4.1.7. Valid create meeting room request	48
4.1.8. Invalid create meeting room request	48
4.1.9. Valid join meeting request	49
4.1.10. Invalid join meeting request	49
4.1.11. Valid start survey request	50
4.1.12. Invalid start survey request	50
4.1.13. Valid answer survey request	50
4.1.14. Invalid answer survey request	51
4.1.15. Valid end survey request	51
4.1.16. Invalid end survey request	52
4.1.17. Stress Test	52
4.2. Database and Hardware	53
4.2.1. Query Fail Error Test	53
4.2.2. Connection Test	54
4.2.3. Performance Test	55
4.3. Eye-tracker API	56
4.3.1. Webcam Test	56
4.3.2. Webcam Test - Negative	57
4.3.3. 실시간 Webcam 데이터 수신 Test	58
4.3.4. 실시간 Webcam 데이터 수신 Test - Negative	58
4.3.5. 시선 인식 기능 Test 1	59
4.3.6. 시선 인식 기능 Test 1 – Negative	59
4.3.7. 시선 인식 기능 Test 2	60
4.3.8. 시선 인식 기능 Test2 – Negative	61
4.3.9. 색상 데이터 제거 Test	62
4.3.10. 색상 데이터 제거 Test – Negative	63
4.3.11. 눈동자 위치 검출 Test	63
4.3.12. 눈동자 위치 검출 Test – Negative	64
4.3.13 노이즈 제거 Test	65
4.3.14. 중심선 그리기 Test	65
4.3.15. 영상 크기에 따른 눈동자 인식 Test	66
5. Supporting Information	68
5.1. Document History	68

LIST OF FIGURES

[Figure 1] Back-end System Architecture	24
[Figure 2] Class Diagram -Profile	29
[Figure 3] Class Diagram - Score	33
[Figure 4] Class Diagram – Meeting Room	35
[Figure 5] Class Diagram – Lecture	39
[Figure 6] Class Diagram - Survey Result	44
[Figure 7] Iostat Output Example	56

LIST OF TABLES

[Table 1] Table of acronyms and abbreviations	6
[Table 2] Table of terms and definitions	7
[Table 3] Document History	68

1. Introduction

1.1. Purpose

본 문서는 스마트 캠퍼스 어플리케이션 SKKU-MEET의 테스트 계획 문서이다. 또한 본 문서는 SKKU-MEET의 개발 팀인 2021년도 성균관대학교 소프트웨어공학개론 수업 Team 9에 의해서 작성되었다. 본 문서에서 기술한 계획을 바탕으로 SKKU-MEET에 대한 테스트가 실시된다. 따라서 본 문서의 주 목적은 SKKU-MEET을 테스트하기 위한 전반적인 계획을 작성하고 독자들에게 이를 밝히는 것에 있다.

1.2. Scope

본 문서에서 다루고 있는 SKKU-MEET는 Oracle RDBMS에 데이터를 저장하는 client-server 방식의 web application이다. 따라서 SKKU-MEET의 테스팅은 크게 데이터베이스에 올바른 데이터가 저장되는가에 대한 테스트, 서버에서 적절한 처리와 클라이언트로의 데이터 전송이 이루어지는가에 대한 테스트, 클라이언트는 서버로부터 받은 데이터를 사용자에게 올바르게 제공하는가에 대한 테스트의 3가지로 나눌 수 있다. 본 문서는 이 3가지를 적절하게 분류하여 다루고 있다.

1.3. Definition, Acronyms, and Abbreviation

아래 표는 해당 문서에서 사용하는 약어들에 대해 설명한다.

[Table 1] Table of acronyms and abbreviations

Acronyms& Abbreviations	Explanation
JSP	Java Server Pages
UI	User Interface
HTML	HyperText Markup Language
AWS	Amazon Web Services

OS	Operating System
RAM	Random Access Memory
OLTP	On-Line Transaction Processing
DFD	Data Flow Diagram

표는 해당 문서에서 사용하는 용어들의 정의에 대해 설명한다.

[Table 2] Table of terms and definitions

Terms	Definitions
사용자	시스템을 사용하는 사람
알고리즘	문제 해결을 위해서 컴퓨터가 따르는 규칙이나 순서
서버	클라이언트에게 네트워크를 통해 정보나 서비스를 제공하는 시스템으로 컴퓨터 프로그램 또는 장치를 의미
클라이언트	서버에 연결된 사용자 디바이스 혹은 사용자
네트워크	디바이스들이 상호간에 정보를 주고받을 수 있도록 연결해 주며 여기서는 인터넷을 의미함
eye-tracking	시선 추적, 즉 눈동자의 움직임을 추적하여 시선의 위치를 알아내는 기술
소프트웨어	컴퓨터에서 사용하는 프로그램 및 기타 작동에 관한 정보

1.4. References

1. “Test plan document format”, posted on SKKU 2021 Introduction to Software Engineering course webpage.

1.5. Overview

본 소프트웨어 테스트 계획 문서는 현재 챕터를 포함하여 크게 네 개의 챕터로 구성된다. 첫번째 챕터에서는 본 테스트 계획 문서의 목적과 개요에 대해 기술하고, 문서에서 사용된 약어 및 용어에 대해 설명하였다. 두번째 챕터에서는 구체적으로 테스트를 어떻게 구현할지에 대해 기술하였다. 세번째 챕터에서는 SKKU-MEET을 구성하는 유닛들을 테스트하기 위한 계획을 작성하였다. 마지막 챕터에서는 SKKU-MEET의 각종 인터페이스들을 테스트하기 위한 계획을 작성하였다.

2. Approach

2.1 Test method

2.1.1 Software Unit Test Methods

2.1.1.1 Back-End

- 1) Live Lecture Storage test 의 pseudo-code (Negative 경우 포함)

Input 받음

```
if (student_id in Oracle DB)
    getStdInfo()

    if (student_concentration == NULL)
        print ("해당 학생의 집중도 데이터를 찾을 수
    없습니다.")

else
    print (해당 학생의 집중도)
```

Input 받음

```
if (student_id in Oracle DB)
    getStdInfo()

    if (student_survey == NULL)
        print ("해당 학생의 설문 결과를 찾을 수 없습니다.")

else
    printf (해당 학생의 설문 답변)
```

- 2) Real-time Survey Creation test 의 pseudo-code (Negative 경우 포함)

```
makeSurvey()

if(설문 생성 == TRUE)
    makeSurveyPage()

else
    print ("설문을 만들 수 없습니다.")
```

3) Real-time Survey Transfer test 의 pseudo-code (Negative 경우 포함)

```
sendSurvey()

if(설문 전송 == Success)
    print ("설문 전송에 성공하였습니다.")

else
    print ("설문 전송에 실패하였습니다.")
```

4) Receive Real-time Survey test의 pseudo-code (Negative 경우 포함)

```
Input 받음

if (student_id in Oracle DB)
    getStdInfo()

    if (student_survey == NULL)
        print ("해당 학생의 설문 결과를 찾을 수 없습니다.")

    else
        print (해당 학생의 설문 답변)
```

```
else
```

```
    print ("해당 학생의 데이터가 DB에 존재하지 않습니다.")
```

5) Lecture Information Integration test 의 pseudo-code (Negative 경우 포함)

```
Input 받음
```

```
if(user_name in 성균관대학교 DB)  
    getStdInfo()
```

```
while (수강 목록의 개수)  
    print (사용자가 수강중인 강의)
```

```
else ("해당 사용자의 수강 목록 데이터를 찾을 수 없습니다. ")
```

2.1.1.2 FrontEnd

1) Set profile 의 pseudo-code

```
Input 받음
```

```
if(user_id != 국제 이메일 주소 문법을 만족)  
    print("Email 형식에 맞지 않습니다")  
  
else if(name != 한글 혹은 영어)  
    print("○] 를은 한글 혹은 영어만 사용할 수 있습니다")  
  
else if(student_id != 성균관대학교 학번 규격)  
    print("학번 형식이 맞지 않습니다")  
  
else if(user_id == 이미 데이터베이스에 존재하는 user_id)
```

```
print("○] 미 사용중인 아이디입니다")  
else if(student_id == ○] 미 데이터베이스에 존재하는  
student_id)  
  
print("○] 미 등록된 학번입니다")  
  
else if(user_id == NULL)  
  
print("○]-○] 디를 입력해 주세요")  
  
else if(name == NULL)  
  
print("○] 름을 입력해 주세요")  
  
else if(student_id == NULL)  
  
print("학번을 입력해 주세요")  
  
else if(job == NULL)  
  
print("직업을 선택해 주세요")  
  
else{  
  
SaveProfileToDB(Input)  
  
print("프로파일이 저장되었습니다")  
  
}  
}
```

2) Get profile의 pseudo-code

```
if(GetProfile 작동중)  
  
print("프로파일 정보를 불러오고  
있습니다")
```

```
else
```

```
    GetProfile();
```

3) Get score ↴ pseudo-code

```
if(score < 0)
```

```
    score = 0
```

```
if(score > 100)
```

```
    score = 100
```

```
GetScore()
```

4) Get score graph ↴ pseudo-code

```
if(score < 0)
```

```
    score = 0
```

```
if(score > 100)
```

```
    score = 100
```

```
GetScoreGraph()
```

5) Survey ↴ pseudo-code

Input 받음

```
if(survey_data > 200바이트)
    print("200바이트 이내로 작성이 가능합니다")
else if(survey_data == NULL)
    print("설문조사는 빈칸으로 제출될 수 없습니다")
else
    Survey_dataToDB()
```

6) Quit 의 pseudo-code

```
if(Quit 작동중)
    print("퇴장 요청을 이미 처리중입니다")
else{
    print("미팅룸에서 퇴장하셨습니다")
    QuitMeetingRoom();
}
```

7) Start lecture 의 pseudo-code

Input 받음

```
if(meeting_room_name > 40바이트)
    print("미팅룸의 이름은 40바이트를 넘지 않습니다")

else if(meeting_room_manager_name != 한글 혹은 영어)
    print("미팅룸 관리자의 이름은 한글 혹은 영어만 설정 가능합니다")

else if(meeting_room_name == NULL)
    print("미팅룸의 이름을 입력해 주세요")

else if(meeting_room_manager_name == NULL)
    print("미팅룸의 관리자 이름을 입력해 주세요")

else{
    SaveProfileToDB(Input)
    print("프로파일이 저장되었습니다")
}
```

8) End lecture 의 pseudo-code

```
if(Endlecture작동중)
    print("미팅룸을 종료하고 있습니다")
else
```

```
EndMeetingRoom();
```

9) Add Lecture 의 pseudo-code

```
Input 받음  
if(lecture_id Not in 성균관대학교 DB)  
    print("존재하지 않는 강의입니다")  
else if(lecture_id == NULL)  
    print("강의 코드 값을 입력해주세요")  
else  
    AddLecture()
```

10) Delete Lecture 의 pseudo-code

```
if>DeleteLecture 작동중)  
    print("강의를 삭제 중입니다")  
else  
    DeleteLecture()
```

11) Show Lecture 의 pseudo-code

```
if(ShowLecture 작동중)

    print("강의 목록을 불러오는
        중입니다")

else

    ShowLectureList()
```

12) Join Lecture 의 pseudo-code

```
if(강의의 student_id 개수 >= 500)

    print("인원이 가득 찼습니다")

else{

    print("강의에
        참가하였습니다")

    JoinLecture()

}
```

13) Open Meeting 의 pseudo-code

```
if(OpenMeetingRoom 작동중)

    print("미팅룸 생성 페이지를 불러오는
        중입니다")

else
```

```
OpenMeetingRoomPage()
```

14) Edit Lecture 의 pseudo-code

```
Input 받음  
if(lecture_id Not in 성균관대학교 DB)  
    print("존재하지 않는 강의입니다")  
else if(lecture_id == NULL)  
    print("강의 코드 값을 입력해주세요")  
else  
    EditLecture()
```

15) Get Survey 의 pseudo-code

```
if(GetSurvey 작동 중)  
    print("설문조사 결과를 불러오는 중입니다")  
else  
    SurveyResultPage()
```

2.1.2 Software Interface Test Methods

2.1.2.1. HTTP Server test methods

1) 4.1.1. Valid register request ~ 4.1.16. Invalid end survey request 의 pseudo-code

```
response = request.METHOD("Server  
addr")  
  
assert response == expected  
  
return response
```

2) 4.1.17. Stress test의 pseudo-code

```
async def stress()  
  
    response = request.METHOD("Server  
addr")  
  
    assert response == expected  
  
    return response  
  
await stress()
```

2.1.2.2. Database and Hardware test methods

1) Query fail error test

오픈소스 소프트웨어인 MySQL/InnoDB의 자체 로그를 적극 활용할 예정이다.

2) Connection test

SKKU:MEET 애플리케이션은 OLTP 워크로드이기 때문에 TPC-C 벤치마크 툴을 이용해 Connection test를 수행한다. 본 테스트는 동시 사용자 수에 따른 데이터베이스의 오류 여부를 확인해본다. Requirements에 명시된 최대 동시 사용자 수까지 테스트하며, partition testing을 이용한다.

3) Performance test

성능 테스트는 connection 테스트와 마찬가지로 TPC-C 벤치마크를 이용하여 TpmC와 95% 99% latency를 측정한다. 또한 iostat 툴을 통해 IOPS 및 CPU, device 사용률을 측정하여 DB와 하드웨어 성능을 측정한다.

2.1.2.3 Eye tracking API test

1) 웹캠이 켜진 상태인지 꺼진 상태인지 확인을 위한 웹캠 Test의 pseudo-code

Webcam Test의 pseudo-code
While 웹캠 is off 웹캠이 꺼져있음을 출력. Eye-tracking procedure(영상데이터)

2) 아이트래킹 API가 웹캠의 데이터를 실시간으로 받는지 확인하기 위한 테스트의 pseudo-code

live data receive의 pseudo-code

```
If (data != webcam)  
  
    return Webcam Test()  
  
Else  
  
    If (live data receive.time ==  
        webcam.time)  
  
        return True
```

3) 시선 인식 기능 테스트의 pseudo-code

eye-tracking test의 pseudo-code
<pre>If (data == null) return Webcam Test() Else While (eye not in data) 시선이 인식되지 않는 것을 출력 While (user count >=2) 사용자는 한 명이어야 함을 출력 색상 제거 procedure call</pre>

4) 색상 제거 기능 Test의 pseudo-code

```
color_delete test의 pseudo-code
```

```
If (data != 웹캠 data)
    return Webcam Test()

Else
    delete data.color

눈동자 위치 검출 procedure
call
```

5) 눈동자 위치 검출 Test의 pseudo-code

```
eye_location test의 pseudo-code
```

```
If (data == null)
    return Webcam Test()

Else
    If (eye.location in range(Frame))
        return eye.location
```

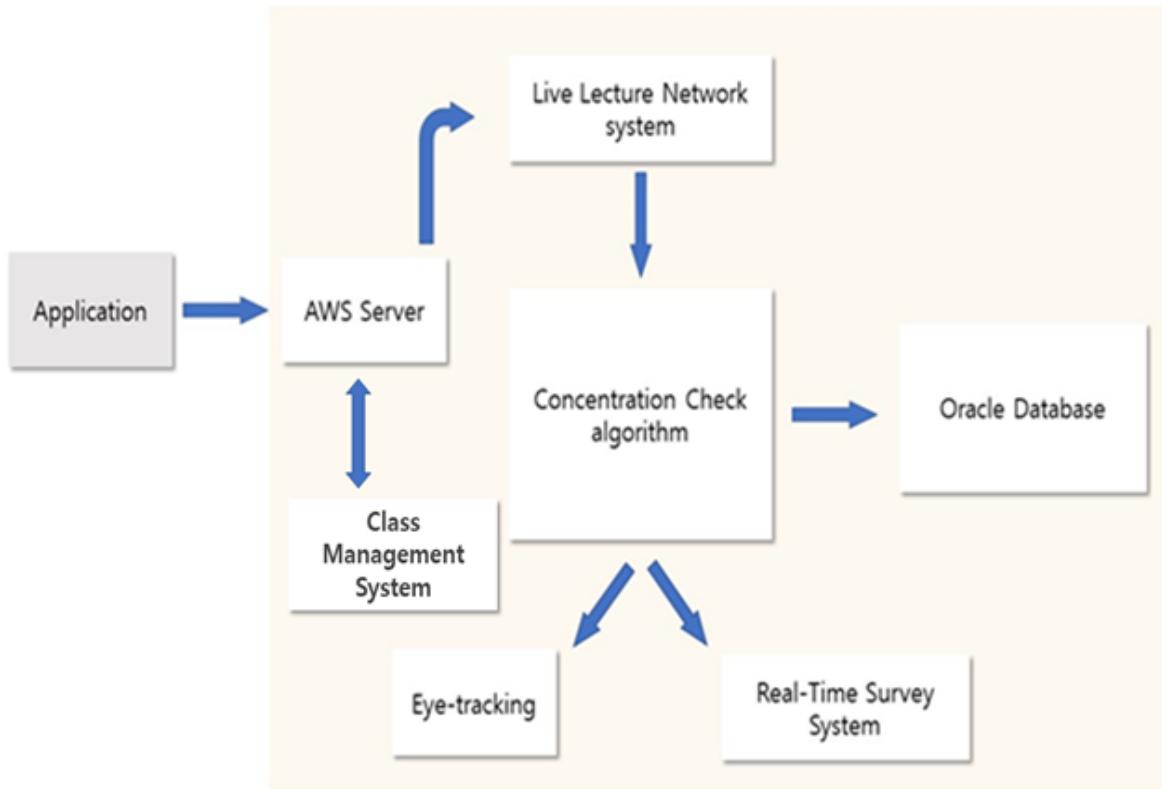
6) 노이즈 제거 Test의 pseudo-code

noise_eliminate test ↴ pseudo-code

```
If (data == null)  
    return Webcam Test()  
  
Else  
  
    If (data.color != null)  
        return color_delete()  
  
    else return blur(data)
```

3. Software Unit Test

3.1. Back-End



[Figure 1] Back-end System Architecture

SKKU:MEET의 백 엔드 시스템의 기능은 위와 같이 나타난다. 각 클래스들의 기능이 제대로 동작하고 결합이 존재하는지 확인할 수 있는 테스트를 진행해야 한다. 따라서 해당 파트에서는 백 엔드 시스템을 구성하는 클래스들이 가지고 있는 독립적인 기능들에 대한 테스트 계획에 대하여 설명하고자 한다. 이때 각 시스템의 클래스 다이어그램은 디자인 명세서에서 확인할 수 있다.

3.1.1. Live Lecture Network System

3.1.1.1. Live Lecture Data Storage test case

- Test case object: 실시간 수업을 하면서 eye-tracking을 통해서 얻을 수 있는 학생의 집중도 데이터 및 설문 결과 데이터가 오라클 데이터베이스에 잘 저장되었는지

확인하기 위해 수업에 속한 학생의 데이터를 쿼리를 통해서 얻어올 수 있는지 체크한다.

- Test Inputs: SQL 쿼리
- Expected Results: 해당 학생의 집중도 데이터 혹은 설문 결과 데이터

: 학생들의 집중도 데이터 및 실시간 설문 결과 데이터는 DB_Handler 클래스의 Write()를 통해 오라클 데이터베이스에 저장될 것이다. 이것이 잘 수행되었는지 테스트하기 위해서 쿼리를 통해 수업에 속한 학생의 데이터를 얻어오도록 질의하고 원하는 데이터를 정확하게 얻어오는지 그 결과를 확인한다.

3.1.1.2. Live Lecture Data Storage test case - Negative

- Test case object: 수업에 속하지 않은 학생의 데이터 및 오라클 데이터베이스에 해당 학생의 집중도 데이터가 없을 때 이를 쿼리를 통해서 얻어오려고 할 경우 실패 알림이 제대로 뜨는지 체크한다.
- Test Inputs: 잘못된 SQL 쿼리
- Expected Results: 데이터를 불러올 수 없다는 실패 메시지

: 오라클 데이터베이스에 존재하지 않는 데이터를 가져오려고 하는 경우 해당 요청이 실패하였음을 알려야 한다. 따라서 잘못된 쿼리를 질의하였을 때 데이터를 가져오는 것에 실패하였다는 알림이 뜨는지 그 결과를 확인한다.

3.1.2. Real-time Survey System

3.1.2.1. Real-time Survey Creation test case

- Test case object: makeSurvey()를 통해 설문이 잘 만들어지는지 체크한다.
- Test Inputs: 개설된 미팅룸에서 설문 만드는 버튼 클릭
- Expected Results: 설문 제작을 위한 페이지

: 교수가 미팅룸에서 설문 제작을 하기 위한 버튼을 클릭할 때 Real-time Survey 시스템의 makeSurvey 클래스가 제대로 수행되어야 한다. 이 작업이 원활히

수행되었는지 확인하기 위해서 버튼 클릭 후 설문 제작을 위한 페이지가 제대로 생성되는지 테스트한다.

3.1.2.2. Real-time Survey Creation test case - Negative

- Test case object: 설문 제작에 오류가 발생하였을 경우 이를 확인할 수 있는지 체크한다.
- Test Inputs: 개설된 미팅룸에서 설문 만드는 버튼 클릭
- Expected Results: 설문 제작 실패 메시지

: 교수가 설문 제작을 하려고 할 때 네트워크 문제나 프로그램 충돌 등의 예상치 못한 오류로 이것이 실패하는 상황이 발생할 수 있다. 이때 시스템은 이러한 오류가 발생하였다는 것을 알려줄 수 있어야 한다. 따라서 이러한 상황을 가정한 테스트 케이스를 만든 후 미팅룸에서 설문 제작 버튼을 눌렀을 때 설문 제작에 실패하였음을 알리는 메시지가 제대로 뜨는지 확인한다.

3.1.2.3. Real-time Survey Transfer test case

- Test case object: sendSurvey()를 통해 학생들에게 설문이 전송되었는지 체크한다.
- Test Inputs: 설문을 전송하기 위한 버튼 클릭
- Expected Results: 설문 전송 성공 메시지

: 만들어진 설문이 학생들에게 제대로 전송되려면 Real-time Survey 시스템의 sendSurvey()가 제대로 작동해야 한다. 설문이 학생들의 로컬 컴퓨터로 제대로 전송이 되었는지 확인하기 위한 테스트가 필요하다. 따라서 전송 버튼을 클릭하고 전송이 성공한 경우 이를 교수에게 알려주는 메시지가 정상적으로 출력되는지 확인한다.

3.1.2.4. Real-time Survey Transfer test case - Negative

- Test case object: 설문이 제대로 전송되지 않은 경우 이를 확인할 수 있는지 체크한다.
- Test Inputs: 설문을 전송하기 위한 버튼 클릭
- Expected Results: 설문 전송 실패 메시지

: 앞서 설문 제작 테스트 케이스에서 말한 것과 같이 설문 전송을 할 때 이것이 실패하는 상황이 벌어질 수 있다. 따라서 sendSurvey 동작 중 문제가 생기거나 학생들의 컴퓨터 및 모바일 기기에 설문이 제대로 전송되지 않은 경우 이를 교수에게 알려서 다시 설문을 전송할 수 있도록 해야 한다. 이러한 상황을 가정한 테스트 케이스를 만든 후 설문을 전송하기 위한 버튼을 클릭하였을 때 전송에 실패하였다는 메시지가 정상적으로 출력되는지 확인한다.

3.1.2.5. Receive Real-time Survey test case

- Test case object: 학생들로부터 설문 결과를 정상적으로 수집할 수 있는지 체크한다.
- Test Inputs: SQL 쿼리
- Expected Results: 해당 학생이 작성한 설문 결과

: 학생들이 작성한 설문 결과는 다시 교수에게 전송되고 이는 DB_Handler를 통해 시스템의 데이터베이스에 저장되어야 한다. 이 과정이 제대로 이루어졌는지 테스트하기 위해서 특정 학생의 설문 결과에 대한 데이터를 데이터베이스에서 가져오는 쿼리를 작성하고 원하는 데이터를 얻어오는지 확인한다.

3.1.2.6. Receive Real-time Survey test case - Negative

- Test case object: 학생이 설문 결과를 전송하지 않았거나 전송 중 오류가 발생하여 해당 데이터가 없는 경우 이를 확인할 수 있는지 체크한다.
- Test Inputs: 잘못된 SQL 쿼리
- Expected Results: 데이터를 불러올 수 없다는 실패 메시지

: 설문 결과 수집 과정에서 오류가 발생하거나 학생이 전송을 하지 않아 설문 결과에 대한 데이터가 오라클 데이터베이스 상에 존재하지 않을 수 있다. 이 경우 실패를 알리는 메시지가 출력되어야 한다. 이는 잘못된 쿼리를 오라클 데이터베이스에 질의한 후 데이터를 불러올 수 없다는 실패 메시지가 정상적으로 출력되는지 확인함으로써 테스트한다.

3.1.3. Class Management System

3.1.3.1. Lecture Information Integration test case

- Test case object: SKKU:MEET 사용자에 대한 데이터베이스가 성균관대학교 데이터베이스와 잘 연동되고 있는지 체크한다.
- Test Inputs: SQL 쿼리
- Expected Results: 학생이 참여중인 강의 목록

: SKKU:MEET의 사용자의 수강 중인 강의에 대한 정보는 성균관대학교 데이터베이스로부터 얻어온다. 따라서 두 데이터베이스 간의 연동이 정상적으로 이루어지고 있는지 확인해야 한다. 따라서 SKKU:MEET의 데이터베이스와 성균관대학교의 데이터베이스에 특정 사용자가 수강 중인 수업에 대한 데이터를 가져올 수 있는 쿼리를 작성한 후 그 결과를 비교하여 Class Management System을 테스트한다.

3.1.3.2. Lecture Information Integration test case - Negative

- Test case object: 성균관대학교 데이터베이스로부터 SKKU:MEET 사용자의 수강중인 수업 목록을 얻어오지 못하는 경우 이를 확인할 수 있는지 체크한다.
- Test Inputs: SKKU:MEET 로그인
- Expected Results: 수강 목록 데이터를 불러올 수 없다는 메시지

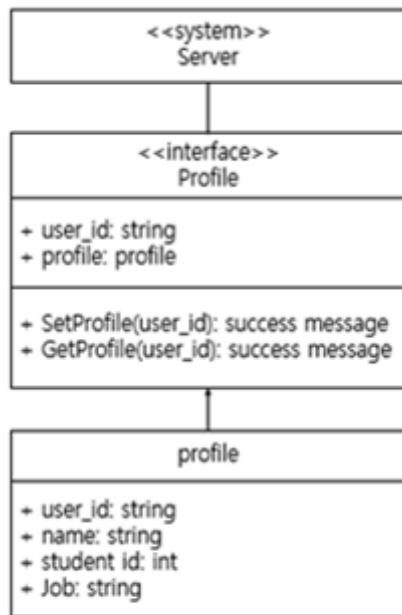
: SKKU:MEET과 성균관대학교 데이터베이스가 원활하게 연동되지 않을 경우에는 사용자의 수강중인 수업 목록을 얻어오는 데에 문제가 발생할 수 있고 이러한 오류가 발생한 것을 알려줄 수 있는 알림이 있어야 한다. 따라서 SKKU:MEET와 성균관대학교 데이터베이스가 연동되지 않은 환경의 테스트 케이스를 작성한 후 시스템에 로그인을 하였을 때 사용자의 수강 목록을 가져올 수 없다는 메시지가 정상적으로 출력되는지 확인한다.

3.2. Front-End

3.2.1. Profile

프로파일 클래스는 기본적인 사용자 정보를 처리한다. 사용자 등록 후 사용자는 프로파일을 입력해야 하며 사용자 로그인 후 사용자는 사용자 프로필을 수정할 수 있다.

3.2.1.1. Class Diagram



[Figure 2] Class Diagram - Profile

3.2.1.2. Test case

- Set Profile test case

Test case object	프로파일 설정 시 조건을 만족하는 입력 값이 들어왔을 때 Set Profile 함수가 정상적으로 동작하는지 확인
Test Inputs	<p>아래 조건을 모두 만족하는 입력 값</p> <ol style="list-style-type: none"> 1) user_id: 국제 이메일 주소 문법을 만족 2) name: 한글 혹은 영어를 사용 3) student_id: 성균관대학교 학번 규격 사용

	4) Job: 교수 혹은 학생 중 택 1
Expected Results	<p>정상적으로 프로파일 저장</p> <p>“프로파일이 저장되었습니다” 문구 출력</p>

- Set Profile test case – Negative

Test case object	프로파일 설정 시 조건을 만족하지 않는 입력 값이 들어왔을 때 Set Profile 함수가 정상적으로 동작하는지 확인
Test Inputs	<p>정해진 조건을 만족하지 못하는 입력 값</p> <p>1) user_id: 국제 이메일 주소 문법을 만족하지 않음</p> <p>예시) Abc.example.com (@ 문자 없음)</p> <p>A@b@c@example.com (인용 부호 밖에서는 오직 @만이 허용됨.)</p> <p>a"b(c)d,e:f;g<h>i[j\k]l@example.com (인용 부호 밖에서 로컬 부분에 특수 문자가 허용되지 않음)</p> <p>just"not"right@example.com</p> <p>this is"not\allowed@example.com</p> <p>this\ still\"not\\allowed@example.com</p> <p>123456789012345678901234567890123456789012345678901234+x@example.com (너무 긴 문자열)</p> <p>john..doe@example.com (@ 이전의 점 두 개)</p> <p>john.doe@example..com (@ 뒤의 점 두 개)</p> <p>2) name: 한글 혹은 영어를 사용하지 않음</p>

	<p>예시) ひろたか, 朱茵, رحمن</p> <p>3) student_id: 성균관대학교 학번 규격 만족하지 않음 예시) 201923, 202삼오, 2018313249324</p> <p>4) user_id: 이메일이 중복됨</p> <p>5) student_id: 학번이 중복됨</p> <p>6) user_id: 입력값이 없음</p> <p>7) name: 입력값이 없음</p> <p>8) student_id: 입력값이 없음</p> <p>9) Job: 입력값이 없음</p>
Expected Results	<p>1) “Email 형식에 맞지 않습니다” 문구 출력</p> <p>2) “이름은 한글 혹은 영어만 사용할 수 있습니다” 문구 출력</p> <p>3) “학번 형식이 맞지 않습니다” 문구 출력</p> <p>4) “이미 사용중인 아이디입니다” 문구 출력</p> <p>5) “이미 등록된 학번입니다” 문구 출력</p> <p>6) “아이디를 입력해 주세요” 문구 출력</p> <p>7) “이름을 입력해 주세요” 문구 출력</p> <p>8) “학번을 입력해 주세요” 문구 출력</p> <p>9) “직업을 선택해 주세요” 문구 출력</p>

- Get Profile test case

Test case object	데이터베이스에서 가져온 프로파일을 나타낼 때 Get Profile 함수가 정상적으로 동작하는지 확인
Test Inputs	프로파일 요청: 1회
Expected Results	프로파일 출력

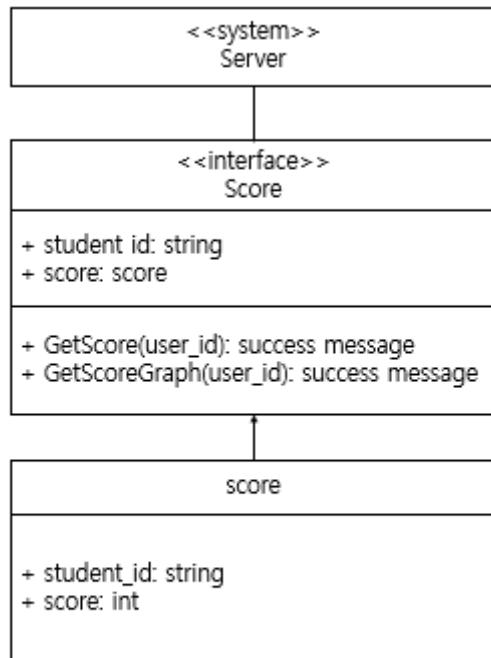
- Get Profile test case - Negative

Test case object	데이터베이스에서 가져온 프로파일을 나타낼 때 Get Profile의 동작을 확인
Test Inputs	프로파일 요청: 2회 이상
Expected Results	“프로파일 정보를 불러오고 있습니다” 문구 출력

3.2.2. Score

스코어 클래스는 Eye tracking을 통한 집중도 테스트를 통해 산출된 스코어 정보를 처리하며 교수로 등록된 사용자가 조회가 가능하다.

3.2.2.1. Class Diagram



[Figure 3] Class Diagram - Score

3.2.1.2. Test case

- Get Score test case

Test case object	데이터베이스에서 가져온 점수를 나타낼 때 Get Score 함수가 정상적으로 동작하는지 확인
Test Inputs	모든 student_id에 대해서 아래 조건을 만족하는 입력 값 1) score: 0 이상 100 이하의 점수
Expected Results	요청한 과목의 점수 리스트 출력

- Get Score test case - Negative

Test case object	데이터베이스에서 가져온 점수를 나타낼 때 Get Score 함수가 비정상적으로 동작하는지 확인
Test Inputs	모든 student_id에 대해서 정해진 조건을 만족하지 않는 입력 값 1) score: 0미만 100초과의 점수
Expected Results	Score를 가까운 border value로 지정 후 요청한 과목의 점수 리스트 출력

- Get Score Graph test case

Test case object	데이터베이스에서 가져온 점수를 나타낼 때 Get Score Graph 함수가 정상적으로 동작하는지 확인
Test Inputs	모든 student_id에 대해서 아래 조건을 만족하는 입력 값 1) score: 0이상 100이하의 점수
Expected Results	요청한 과목의 점수 그래프 출력

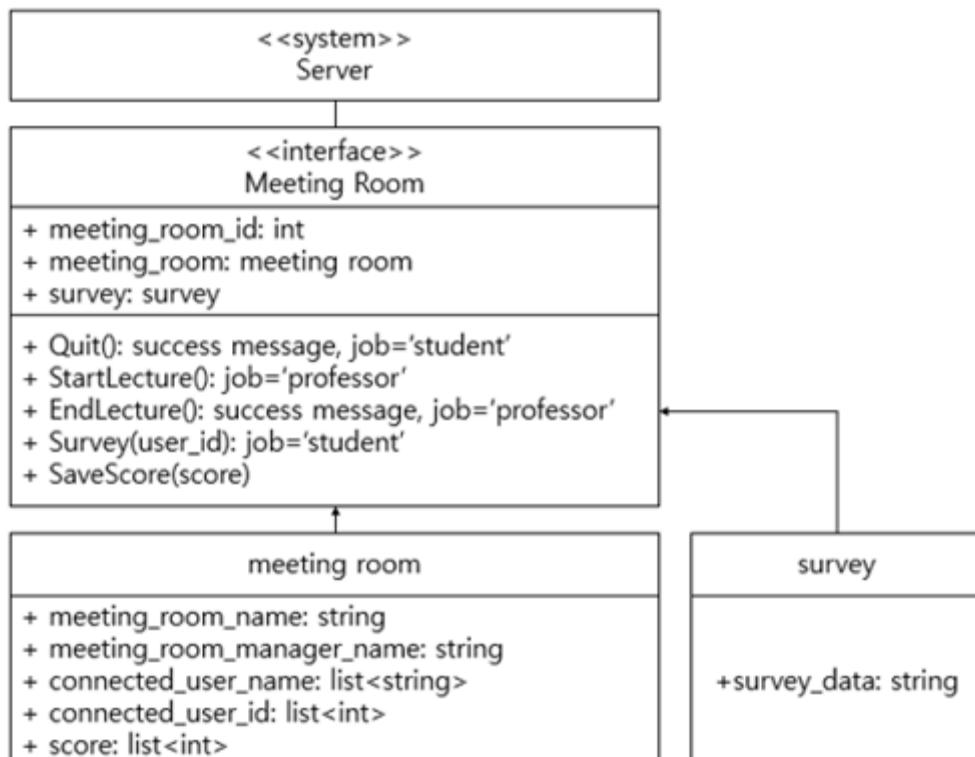
- Get Score Graph test case – Negative

Test case object	데이터베이스에서 가져온 점수를 나타낼 때 Get Score Graph 함수가 비정상적으로 동작하는지 확인
Test Inputs	모든 student_id에 대해서 정해진 조건을 만족하지 않는 입력 값 1) score: 0미만 100초과의 점수
Expected Results	Score를 가까운 border value로 지정 후 요청한 과목의 점수 그래프 출력

3.2.3 Meeting Room

미팅룸 클래스는 교수로 등록된 사용자가 개설한 미팅룸에 대한 정보를 처리한다. 미팅룸에서는 실시간 강의를 진행할 뿐만 아니라 Eye tracking을 활용한 집중도 체크와 그로 인해 산출된 점수를 저장하고 설문조사를 실시하여 산출된 결과를 저장한다. 설문조사 데이터는 작성자를 확인할 수 없게 익명으로 저장된다.

3.2.3.1. Class Diagram



[Figure 4] Class Diagram – Meeting Room

3.2.3.2. Test Case

- Survey test case

Test case object	학생들이 설문조사를 실행할 때 Survey 함수가 정상적으로 동작하는지 확인
Test Inputs	survey_data: 200바이트 이내의 문자열

Expected Results	정상적으로 설문조사 결과 저장 “설문조사가 제출되었습니다” 문자 출력
------------------	---

- Survey test case – Negative

Test case object	학생들이 설문조사를 실행할 때 조건에 맞지 않는 입력값에 대해 Get Profile 함수의 동작을 확인
Test Inputs	정해진 조건을 만족하지 못하는 입력 값 1) survey_data: 200바이트 초과의 문자열 2) survey_data: 입력 값이 없음
Expected Results	1) “200바이트 이내로 작성이 가능합니다” 문구 출력 2) “설문조사는 빙칸으로 제출될 수 없습니다” 문구 출력

- Quit test case

Test case object	Meeting Room을 퇴장할 때 Quit 함수가 정상적으로 동작하는지 확인
Test Inputs	퇴장 요청: 1회
Expected Results	정상적으로 퇴장 처리 “미팅룸에서 퇴장하셨습니다” 문구 출력

- Quit test case – Negative

Test case object	Meeting Room을 퇴장할 때 Quit 함수의 동작을 확인
Test Inputs	퇴장 요청: 2회 이상
Expected Results	“퇴장 요청을 이미 처리중입니다” 문구 출력

- Start lecture test case

Test case object	미팅룸을 개설할 때 조건에 맞게 주어진 입력 값에 대해 정상적으로 Start Lecture 함수가 동작하는지 확인
Test Inputs	<p>아래 조건을 모두 만족하는 입력 값</p> <p>1) meeting_room_name: 40바이트 이내의 문자열</p> <p>2) meeting_room_manager_name: 한글 혹은 영어를 사용</p>
Expected Results	<p>“미팅룸을 시작합니다” 문구 출력</p> <p>정상적으로 미팅룸 개설</p>

- Start lecture test case – Negative

Test case object	미팅룸을 개설할 때 조건에 맞지 않게 주어진 입력 값에 대한 Start Lecture 함수의 동작을 확인
Test Inputs	<p>1) meeting_room_name: 40바이트 초과의 문자열</p> <p>예시) 이것은 예시로 이뤄지는 강의의 미팅룸 이름입니다. (57바이트)</p> <p>40바이트를 맞춘 미팅룸의 이름입니다. (40바이트)</p> <p>this is the forty byte sentence to open. (40바이트)</p>

	<p>2) meeting_room_manager_name: 한글 혹은 영어를 사용 예시) ひろたか, 朱茵, رحمن</p> <p>3) meeting_room_name: 입력 값이 없음</p> <p>4) meeting_room_manager_name: 입력 값이 없음</p>
Expected Results	<p>1) “미팅룸의 이름은 40바이트를 넘지 않습니다” 문구 출력</p> <p>2) “미팅룸 관리자의 이름은 한글 혹은 영어만 설정 가능합니다” 문구 출력</p> <p>3) “미팅룸의 이름을 입력해 주세요” 문구 출력</p> <p>4) “미팅룸의 관리자 이름을 입력해 주세요” 문구 출력</p>

- End lecture test case

Test case object	미팅룸을 끝낼 때 End lecture 함수가 정상적으로 동작하는지 확인
Test Inputs	미팅룸 종료 요청: 1회
Expected Results	정상적으로 미팅룸을 종료

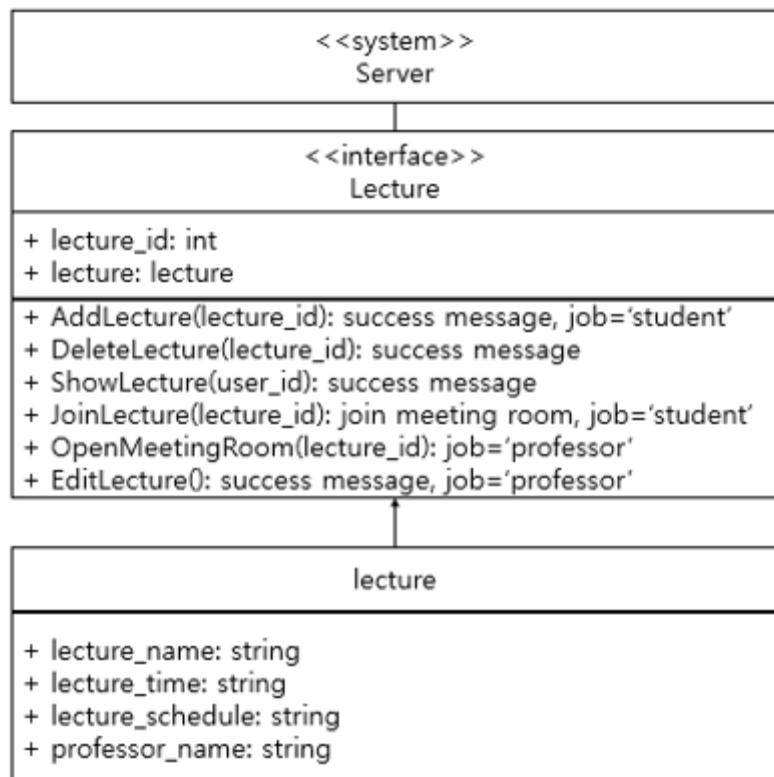
- End lecture test case – Negative

Test case object	미팅룸을 끝낼 때 End lecture 함수의 동작을 확인
Test Inputs	미팅룸 종료 요청: 2회 이상
Expected Results	“미팅룸을 종료하고 있습니다” 문구 출력

3.2.4 Lecture

강의 클래스는 사용자가 저장한 강의에 대한 정보를 처리한다. 강의에 대한 정보는 강의id, 강의 이름, 강의 진행시간, 강의 일정등이 저장된다. 사용자는 강의를 추가, 삭제할 수 있으며 생성된 강의에 입장할 수 있다. 교수는 자신이 관리하는 강의에서 미팅룸을 생성할 수 있으며 강의의 데이터를 수정할 수 있다.

3.2.4.1. Class Diagram



[Figure 5] Class Diagram – Lecture

3.2.4.2. Test Case

- Add Lecture test case

Test case object	강의를 계정에 등록할 때 조건에 맞게 주어진 입력 값에 대해 Add lecture 함수가 정상적으로 동작하는지 확인
Test Inputs	1) lecture_id: 성균관대학교에 등록된 강의의 코드를 등록

Expected Results	1) 정상적으로 강의를 등록
------------------	-----------------

- Add Lecture test case – Negative

Test case object	강의를 계정에 등록할 때 조건에 맞지 않게 주어진 입력 값에 대해 Add lecture 함수의 동작을 확인
Test Inputs	<p>정해진 조건을 만족하지 못하는 입력 값</p> <p>1) lecture_id: 성균관대학교에 등록되지 않은 강의의 코드를 등록</p> <p>2) lecture_id: 강의의 코드가 입력 값이 없음</p>
Expected Results	<p>1) “존재하지 않는 강의입니다” 문구 출력</p> <p>2) “강의 코드 값을 입력해 주세요” 문구 출력</p>

- Delete Lecture test case

Test case object	계정에 등록된 강의 목록에 있는 강의를 삭제할 때 Delete lecture 함수가 정상적으로 동작하는지 확인
Test Inputs	강의 삭제 요청: 1회
Expected Results	정상적으로 강의를 삭제

- Delete Lecture test case – Negative

Test case object	계정에 등록된 강의 목록에 있는 강의를 삭제할 때 Delete lecture 함수의 동작을 확인
------------------	---

Test Inputs	강의 삭제 요청: 2회 이상
Expected Results	“강의를 삭제 중입니다” 문구 출력

- Show Lecture test case

Test case object	계정에 등록된 강의 목록에 있는 강의를 삭제할 때 Delete lecture 함수의 동작을 확인
Test Inputs	강의 목록 요청: 1회
Expected Results	정상적으로 강의목록 페이지 출력.

- Show Lecture test case – Negative

Test case object	강의목록을 요청했을 때 End lecture 함수의 동작을 확인
Test Inputs	강의 목록 요청: 2회 이상
Expected Results	“강의목록을 불러오는 중입니다” 문구 출력

- Join Lecture test case

Test case object	강의에 참가할 때 특정 조건에서 Join lecture 함수가 정상적으로 동작하는지 확인
Test Inputs	인원수가 500명이 안되는 빈 자리가 있는 강의에 참가
Expected Results	“강의에 참가하였습니다” 문구 출력

	강의 미팅룸 참가
--	-----------

- Join Lecture test case – Negative

Test case object	강의에 참가할 때 특정 조건에서 Join lecture 함수의 동작을 확인
Test Inputs	인원수가 500명으로 가득 찬 강의에 참가
Expected Results	“인원이 가득 찼습니다” 문구 출력

- Open Meeting Room test case

Test case object	미팅룸 생성을 요청할 때 Open meeting room 함수가 정상적으로 동작하는지 확인
Test Inputs	미팅룸 생성 요청: 1회
Expected Results	정상적으로 미팅룸 생성 페이지 출력

- Open Meeting Room test case – Negative

Test case object	미팅룸 생성을 요청할 때 Open meeting room 함수의 동작을 확인
Test Inputs	미팅룸 생성 요청: 2회 이상
Expected Results	“미팅룸 생성 페이지를 불러오는 중입니다” 문구 출력

- Edit Lecture test case

Test case object	계정에 등록된 강의의 코드를 수정할 때 조건에 맞게 주어진 입력 값에 대해 Edit Lecture 함수가 정상적으로 동작하는지 확인
Test Inputs	1) lecture_id: 성균관대학교에 등록된 강의의 코드를 등록
Expected Results	1) 정상적으로 강의를 등록

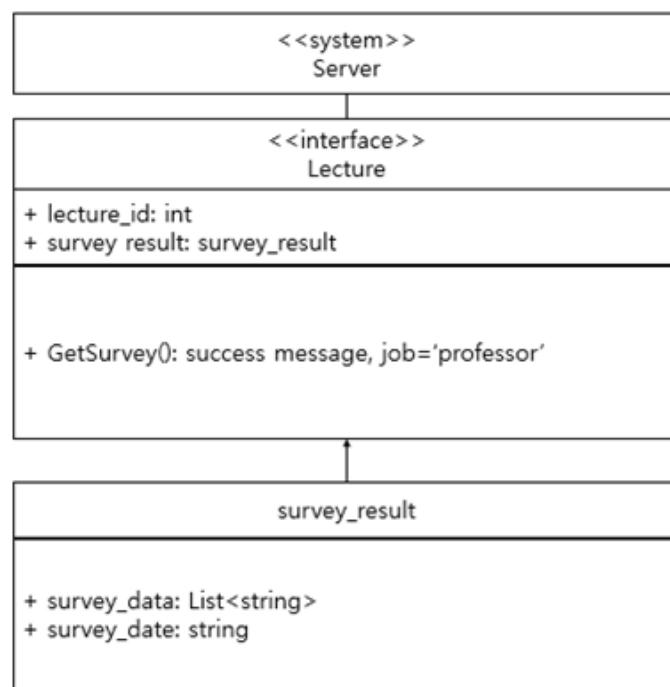
- Edit Lecture test case – Negative

Test case object	계정에 등록된 강의의 코드를 수정할 때 조건에 맞지 않게 주어진 입력 값에 대해 Edit lecture 함수의 동작을 확인
Test Inputs	정해진 조건을 만족하지 못하는 입력 값 1) lecture_id: 성균관대학교에 등록되지 않은 강의의 코드를 등록 2) lecture_id: 강의의 코드가 입력 값이 없음
Expected Results	1) “존재하지 않는 강의입니다” 문구 출력 2) “강의 코드 값을 입력해 주세요” 문구 출력

3.2.5 Survey Result

설문조사 결과 클래스는 미팅룸에서 저장된 설문조사 데이터를 처리한다. 교수는 자신이 관리하는 강의에 해당하는 설문조사 데이터를 조회할 수 있다.

3.2.5.1. Class Diagram



[Figure 6] Class Diagram - Survey Result

3.2.5.2. Test Case

- Get Survey test case

Test case object	설문조사 결과를 요청할 때 Open meeting room 함수가 정상적으로 동작하는지 확인
Test Inputs	설문조사 결과 요청: 1회
Expected Results	정상적으로 설문조사 결과 페이지 출력

- Get Survey test case - Negative

Test case object	설문조사 결과를 요청할 때 Open meeting room 함수의 동작을 확인
Test Inputs	설문조사 결과 요청: 2회 이상
Expected Results	“설문조사 결과를 불러오는 중입니다” 문구 출력

4. Software Interface test

4.1. HTTP server

4.1.1. Valid register request

4.1.1.1. Test case object

올바른 회원가입 요청 HTTP에 서버가 정확한 응답을 보내는지 확인한다

4.1.1.2. Test Input

HTTP 요청: POST 메소드. HTTP 파라미터로 올바른 Id, Pass, Type, Affiliation을 포함시킨다.

4.1.1.3. Expected Results

HTTP 응답: 200 OK 코드를 포함한다. HTTP 바디에 사용자 인증에 사용될 액세스 토큰 및 회원가입 성공 메시지를 포함한다.

4.1.2. Invalid register request

4.1.2.1. Test case object

유효하지 않은 회원가입 요청 HTTP에 서버가 정확한 응답을 보내는지 확인한다

4.1.2.2. Test Input

HTTP 요청: POST 메소드. HTTP 파라미터로 유효하지 않은 Id, Pass, Type, affiliation을 포함시킨다. 혹은 파라미터 중 일부를 포함시키지 않는다.

4.1.2.3. Expected Results

HTTP 응답: 400 에러 코드를 포함한다. HTTP 바디에 회원가입 실패 메시지를 포함한다.

4.1.3. Valid sign in request

4.1.3.1. Test case object

올바른 로그인 요청 HTTP에 서버가 정확한 응답을 보내는지 확인한다.

4.1.3.2. Test Input

HTTP 요청: POST 메소드. HTTP 파라미터로 올바른 Id, Pass를 포함시킨다.

4.1.3.3. Expected Results

HTTP 응답: 200 OK 코드를 포함한다. HTTP 바디에 사용자 인증에 사용될 액세스 토큰 및 로그인 성공 메시지를 포함한다.

4.1.4. Invalid sign in request

4.1.4.1. Test case object

유효하지 않은 로그인 요청 HTTP에 서버가 정확한 응답을 보내는지 확인한다

4.1.4.2. Test Input

HTTP 요청: POST 메소드. HTTP 파라미터로 유효하지 않은 Id, Pass를 포함시킨다. 혹은 파라미터 중 일부를 포함시키지 않는다.

4.1.4.3. Expected Results

HTTP 응답: 400 에러 코드를 포함한다. HTTP 바디에 로그인 실패 메시지를 포함한다.

4.1.5. Valid view class list request

4.1.5.1. Test case object

올바른 클래스 목록 보기 요청 HTTP에 서버가 정확한 응답을 보내는지 확인한다.

4.1.5.2. Test Input

HTTP 요청: GET 메소드. HTTP 파라미터로 올바른 User, Authorization를 포함시킨다.

4.1.5.3. Expected Results

HTTP 응답: 200 OK 코드를 포함한다. HTTP 바디에 사용자가 운영/수강 중인 강의들의 목록 및 정보를 포함한다.

4.1.6. Invalid view class list request

4.1.6.1. Test case object

유효하지 않은 클래스 목록 보기 요청 HTTP에 서버가 정확한 응답을 보내는지 확인한다.

4.1.6.2. Test Input

HTTP 요청: GET 메소드. HTTP 파라미터로 유효하지 않은 User, Authorization를 포함시킨다. 혹은 파라미터 중 일부를 포함시키지 않는다.

4.1.6.3. Expected Results

HTTP 응답: 403 에러 코드를 포함한다. HTTP 바디에 강의 리스트 불러오기 실패 메시지를 포함한다.

4.1.7. Valid create meeting room request

4.1.7.1. Test case object

올바른 미팅룸 생성 요청 HTTP에 서버가 정확한 응답을 보내는지 확인한다.

4.1.7.2. Test Input

HTTP 요청: POST 메소드. HTTP 파라미터로 올바른 Class_id, Authorization를 포함시킨다.

4.1.7.3. Expected Results

HTTP 응답: 200 OK 코드를 포함한다. HTTP 바디에 생성된 미팅룸의 url을 비롯한 정보를 포함한다.

4.1.8. Invalid create meeting room request

4.1.8.1. Test case object

유효하지 않은 미팅룸 생성 요청 HTTP에 서버가 정확한 응답을 보내는지 확인한다.

4.1.8.2. Test Input

HTTP 요청: POST 메소드. HTTP 파라미터로 유효하지 않은 Class_id, Authorization를 포함시킨다. 혹은 파라미터 중 일부를 포함시키지 않는다.

4.1.8.3. Expected Results

HTTP 응답: 403 에러 코드를 포함한다. HTTP 바디에 미팅룸 생성 실패 메시지를 포함한다.

4.1.9. Valid join meeting room request

4.1.9.1. Test case object

올바른 미팅룸 참여 요청 HTTP에 서버가 정확한 응답을 보내는지 확인한다.

4.1.9.2. Test Input

HTTP 요청: POST 메소드. HTTP 파라미터로 올바른 Class_id, Authorization를 포함시킨다.

4.1.9.3. Expected Results

HTTP 응답: 200 OK 코드를 포함한다. HTTP 바디에 참가 요청된 강의의 미팅룸 url 및 정보를 포함한다

4.1.10. Invalid join meeting room request

4.1.10.1. Test case object

유효하지 않은 미팅룸 참여 요청 HTTP에 서버가 정확한 응답을 보내는지 확인한다.

4.1.10.2. Test Input

HTTP 요청: POST 메소드. HTTP 파라미터로 유효하지 않은 Class_id, Authorization를 포함시킨다. 혹은 파라미터 중 일부를 포함시키지 않는다.

4.1.10.3. Expected Results

HTTP 응답: 403 에러 코드를 포함한다. HTTP 바디에 미팅룸 참여 실패 메시지를 포함한다.

4.1.11. Valid start survey request

4.1.11.1. Test case object

올바른 실시간 설문 시작 요청 HTTP에 서버가 정확한 응답을 보내는지 확인한다.

4.1.11.2. Test Input

HTTP 요청: POST 메소드. HTTP 파라미터로 올바른 Survey_info, Authorization를 포함시킨다.

4.1.11.3. Expected Results

HTTP 응답: 200 OK 코드를 포함한다. HTTP 바디에 실시간 설문 시작 성공 메시지를 포함한다.

4.1.12. Invalid start survey request

4.1.12.1. Test case object

유효하지 않은 실시간 설문 시작 요청 HTTP에 서버가 정확한 응답을 보내는지 확인한다.

4.1.12.2. Test Input

HTTP 요청: POST 메소드. HTTP 파라미터로 유효하지 않은 Survey_info, Authorization를 포함시킨다. 혹은 파라미터 중 일부를 포함시키지 않는다.

4.1.12.3. Expected Results

HTTP 응답: 403 에러 코드를 포함한다. HTTP 바디에 실시간 설문 시작 실패 메시지를 포함한다.

4.1.13. Valid answer survey request

4.1.13.1. Test case object

올바른 실시간 설문 응답 요청 HTTP에 서버가 정확한 응답을 보내는지 확인한다.

4.1.13.2. Test Input

HTTP 요청: POST 메소드. HTTP 파라미터로 올바른 Answer, Authorization를 포함시킨다.

4.1.13.3. Expected Results

HTTP 응답: 200 OK 코드를 포함한다. HTTP 바디에 실시간 설문 응답 성공 메시지를 포함한다.

4.1.14. Invalid answer survey request

4.1.14.1. Test case object

유효하지 않은 실시간 설문 응답 요청 HTTP에 서버가 정확한 응답을 보내는지 확인한다.

4.1.14.2. Test Input

HTTP 요청: POST 메소드. HTTP 파라미터로 유효하지 않은 Answer, Authorization를 포함시킨다. 혹은 파라미터 중 일부를 포함시키지 않는다.

4.1.14.3. Expected Results

HTTP 응답: 400 에러 코드를 포함한다. HTTP 바디에 실시간 설문 응답 실패 메시지를 포함한다.

4.1.15. Valid end survey request

4.1.15.1. Test case object

올바른 실시간 설문 종료 요청 HTTP에 서버가 정확한 응답을 보내는지 확인한다.

4.1.15.2. Test Input

HTTP 요청: POST 메소드. HTTP 파라미터로 올바른 Survey_info, Authorization를 포함시킨다.

4.1.15.3. Expected Results

HTTP 응답: 200 OK 코드를 포함한다. HTTP 바디에 실시간 설문 종료 성공 메시지를 포함한다.

4.1.16. Invalid end survey request

4.1.16.1. Test case object

유효하지 않은 실시간 설문 시작 요청 HTTP에 서버가 정확한 응답을 보내는지 확인한다.

4.1.16.2. Test Input

HTTP 요청: POST 메소드. HTTP 파라미터로 유효하지 않은 Survey_info, Authorization를 포함시킨다. 혹은 파라미터 중 일부를 포함시키지 않는다.

4.1.16.3. Expected Results

HTTP 응답: 403 에러 코드를 포함한다. HTTP 바디에 실시간 설문 종료 실패 메시지를 포함한다.

4.1.17. Stress test

4.1.17.1. Test case object

동시 다발적인 클라이언트들의 요청에 대해 서버가 적절한 응답을 보내는지 확인한다.

4.1.17.2. Test Input

비동기(asynchronous) 프로그래밍 기법을 이용하여 어플리케이션 서버에 동시다발적으로 여러 개의 요청을 보낸다. 해당 요청들의 구체적인 내용은 4.1.1.에서 4.1.16.까지의 test input들 중에서 선택한다.

4.1.17.3. Expected Results

마찬가지로 비동기 프로그래밍 기법을 이용해 서버로부터 되돌아오는 HTTP응답을 하나씩 확인하며 4.1.1.에서 4.1.16까지 언급된 expected results와 다른 응답이 있는지 검사한다.

4.2. Database and Hardware Test

본 애플리케이션은 OLTP(On-Line Transaction Processing) 워크로드이므로 데이터베이스 대표적인 벤치마크인 TPC-C(Transaction Processing Performance Council) 벤치마크를 기반으로 데이터베이스 connection 그리고 성능 테스트를 수행하고자 한다. Error 테스트의 경우 사용하는 MySQL/InnoDB 오픈 소스 코드에 디버깅 코드를 부분적으로 추가하여 수행할 예정이다. 본 테스트는 코드 구현 과정에서 error handling은 모두 마친 상태로 간주한다.

4.2.1 Query Fail Error Test

4.2.1.1. Test Case Object

본 테스트는 Insert, Update, Delete, Select문이 제대로 실행되는지 확인한다. 모든 테이블과 각각의 인덱스가 지정된 상태에서 테스트를 수행한다. mysqld 서버를 실행할 때 로그 파일 위치를 옵션으로 지정하면 쿼리가 실패할 경우 error 로그가 찍혀 확인할 수 있다.

4.2.1.2. Test Input

옵션 추가 방법 예시:

```
$ "log-error      = /dir/to/mysql_error.log" >> my.cnf
```

Mysqld 서버 옵션 파일로 실행하는 방법:

```
$ ./bin/mysqld_safe --defaults-file=/dir/to/my.cnf
```

Mysql user로 접속하는 방법:

```
$ ./bin/mysql -uroot -p password -S/tmp/mysql.sock
```

테스트 할 쿼리

INSERT 쿼리의 경우, 랜덤함수를 통해 value를 지정하도록 한다.

4.2.1.3. Expected Results

모든 쿼리가 성공적으로 실행된다면, mysql_error.log 파일의 로그에 에러가 있어서는 안 된다.

4.2.2. Connection Test

4.2.2.1. Test Case Object

Connection test는 TPC-C 벤치마크를 이용해 동시 사용자 수에 따른 데이터베이스의 오류 여부를 확인해본다. Requirements에 명시된 최대 동시 사용자 수까지 테스트하며, partition testing을 이용한다.

4.2.2.2. Test Input

[Percona GitHub repositories](#)에서 tpcc-mysql을 복사해온다.

```
$ git clone https://github.com/Percona-Lab/tpcc-mysql.git
```

tpcc-mysql directory에 가서 binary 파일들을 빌드한다.

```
$ cd tpcc-mysql/src
```

```
$ make
```

Connection 테스트를 하기 전 데이터를 로드한다. 데이터를 로드하기 위해서 주어진 load.sh 스크립트를 수정하여 사용한다.

```
$ cd tpcc-mysql
$ vi load.sh

export LD_LIBRARY_PATH=/home/lbh/mysql-5.7.24/lib
DBNAME=$1
WH=$2
HOST=127.0.0.1
STEP=100

./tpcc_load -h $HOST -d $DBNAME -u root -p "evia6587" -P3306 -w $WH -l 1 -m 1 -n $WH >> 1.out &
x=1

while [ $x -le $WH ]
do
echo $x $(( $x + $STEP - 1 ))
./tpcc_load -h $HOST -d $DBNAME -u root -p "evia6587" -P3306 -w $WH -l 2 -m $x -n $(( $x + $STEP - 1 )) >> 2_$x.out &
./tpcc_load -h $HOST -d $DBNAME -u root -p "evia6587" -P3306 -w $WH -l 3 -m $x -n $(( $x + $STEP - 1 )) >> 3_$x.out &
./tpcc_load -h $HOST -d $DBNAME -u root -p "evia6587" -P3306 -w $WH -l 4 -m $x -n $(( $x + $STEP - 1 )) >> 4_$x.out &
x=$(( $x + $STEP ))
done

for pid in `jobs -p`
do
echo wait for $pid
wait $pid
done

$ sudo chmod 777 load.sh
```

- Load data:

```
$ ./load.sh tpcc1000 1000
```

데이터 로드가 끝난 뒤, 동시 사용자 수를 달리하여 tpcc 벤치마크를 수행하고 오류 여부를 확인한다. 동시 사용자 수는 -c 옵션으로 조정 가능하다. 동시 사용자 수는 최솟값인 300과 최댓값인 50,000 그리고 60,000으로 테스트를 진행한다.

```
$ ./tpcc_start -h127.0.0.1 -S/tmp/mysql.sock -dtpcc1000 -uroot -pyourPassword -w100 -c32 -r10 -l1200
```

4.2.2.3. Expected Results

동시 사용자 수가 300부터 500,000까지일 경우에는 mysql 서버 connection이 끊기지 않아야 한다. 다만 트랜잭션 처리 속도는 줄어들 것이다. 이는 다음 테스트인 성능 테스트에서 더 자세히 다룰 예정이다.

4.2.3 Performance Test

4.2.3.1. Test Case Objective

성능 테스트의 경우, 애플리케이션의 지연 시간이 목표 지연 시간을 넘지 않는지 확인하기 위해 진행한다. Connection 테스트와 동일하게 TPC-C 벤치마크를 수행하여 95%, 99% latency를 비교하며, 추가적으로 iostat 소프트웨어 툴을 이용해 io 성능과 device 사용률, 그리고 CPU 사용률을 확인한다.

4.2.3.2. Test Input

다음과 같이 TPC-C 벤치마크와 iostat을 동시에 수행한다. 각 결과는 tpcc.txt와 iostat.txt에 저장된다.

```
$ ./tpcc_start -h127.0.0.1 -S/tmp/mysql.sock -dtpcc1000 -uroot -pyourPassword -w100 -c32 -r10 -l1200 >>
tpcc.txt | iostat -x 10 120 >> iostat.txt
```

다음은 tpcc.txt에 저장되는 예시 결과값이다. 다음 결과값을 통해 TpmC와 99% 95% latency를 확인한다.

avg-cpu:	%user	%nice	%system	%iowait	%steal	%idle									
Device:	rrqm/s	wrrqm/s	r/s	w/s	rkB/s	wkB/s	avgrq-sz	avgqu-sz	await	r_await	w_await	svctm	%util		
loop0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
nvme0n1	0.00	278.10	5017.20	3796.60	27445.20	33060.00	13.73	1.52	0.26	0.28	0.23	0.05	43.52		
nvme1n1	0.00	0.50	10.90	0.70	244.80	4.80	43.03	0.01	0.90	0.70	4.00	0.17	0.20		
sda	0.00	133.20	316.20	819.00	1264.80	5624.80	12.14	0.05	0.05	0.09	0.03	0.05	5.24		
sdb	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

[Figure 7] Iostat Output Example

저장장치의 사용 비율과 CPU 사용률을 확인함으로써 r/s와 w/s, TpmC 그리고 latency를 보다 정확히 분석할 수 있다.

4.2.3.3. Expected Result

최대 동시 사용자 수로 벤치마크를 수행 시 99% latency 최대값이 기준 5초를 넘기지 않는다. 또한 평균 99% latency는 2초를 넘기지 않는다.

4.3. Eye-tracker API

4.3.1. Webcam Test

Test case object	아이트래킹 API가 사용자의 Webcam이 켜져 있을 때의 입력을 정상적으로 받는지 테스트한다.
테스트 입력 값	Webcam의 on/off 상태 데이터 1) Webcam이 켜져 있을 때의 처리 Webcam이 켜져 있을 때 아이트래킹 API가 영상 입력을 정상적으로 받는지 확인한다.
기대 결과값	1) Webcam 입력 값 인식, 시선 인식 procedure call
발생 가능 error	1) Interface misuse: Webcam 데이터를 정상적으로 불러오지 못하거나, 다른 데이터를 불러와 에러가 발생한다 2) Timing errors: 시선 인식 procedure의 call이 지연되면 timing errors가 발생 가능하다.

4.3.2. Webcam Test - Negative

Test case object	아이트래킹 API가 사용자의 Webcam이 꺼져 있을 때의 처리를 정상적으로 하는지 테스트한다.
테스트 입력 값	Webcam의 on/off 상태 데이터 1) Webcam이 꺼져 있을 때의 처리 Webcam이 꺼져 있을 때 아이트래킹 API가 꺼진 상태, 즉 영상데이터 입력이 없는 Null 값 입력을 정상적으로 처리하는지를 확인한다.
기대 결과값	1) Null 값 인식, 'webcam이 꺼져 있습니다.' 출력 Webcam에서 영상 입력이 수신될 때까지 대기

발생 가능 error	1) Null 값을 인식하지 못하고 다음 procedure로 넘어간다면 에러가 발생할 수 있다.
----------------	---

4.3.3. 실시간 Webcam 데이터 수신 Test

Test case object	아이트래킹 API가 Webcam의 데이터를 실시간으로 입력 받는지 테스트한다.
테스트 입력 값	실시간 Webcam 영상 데이터 1) Webcam 영상 데이터 영상 데이터를 지연 없이 실시간으로 입력 받는지 테스트한다.
기대 결과값	1) 실시간 Webcam 영상 데이터를 입력 받은 후 시선 인식 procedure를 call한다.
발생 가능 error	1) Timing errors: 아이트래킹 API가 실시간 webcam 데이터가 아닌 과거의 데이터를 받는다. 영상 데이터를 수신할 때 지연도가 너무 크면 발생할 수 있다.

4.3.4. 실시간 Webcam 데이터 수신 Test - Negative

Test case object	아이트래킹 API가 Webcam의 데이터를 받지 못했을 때의 결과를 테스트한다.
테스트 입력 값	1) Null 값 Null 값 입력에 대하여 에러를 일으키지 않고 정상적으로 대응하는지 확인한다.

기대 결과값	1) Null 값 인식, 웹캠 Test procedure로 복귀 Null 값 입력 시, 정상적인 영상 데이터를 얻기 위해 다시 웹캠 test procedure로 복귀한다.
발생 가능 error	1) Null 값을 인식하지 못하고 다음 procedure로 넘어간다면 에러가 발생할 수 있다.

4.3.5. 시선 인식 기능 Test 1

Test case object	아이트래킹 API가 입력 받은 Webcam의 데이터에서 사용자의 시선을 인식하는지 테스트한다.
테스트 입력 값	Webcam으로부터 받는 사용자 영상 데이터 1) 안경을 착용하지 않은 사용자의 눈이 나오는 화면 안경을 착용하지 않은 사용자의 시선을 정상적으로 인식하는지 확인한다. 2) 안경을 착용한 사용자의 눈이 나오는 화면 안경을 착용한 사용자에 대하여 안경에 관계없이 시선을 정상적으로 인식하는지 확인한다.
기대 결과값	1) 정상적으로 사용자의 시선을 인식한 후 색상 데이터 제거 procedure call한다.
발생 가능 error	1) Interface misuse: 데이터를 Webcam가 아닌 다른 component로부터 받아온다. 2) Interface misunderstanding: 기대 입력 값으로 예상하지 못한 데이터가 Webcam으로부터 들어온다.

4.3.6. 시선 인식 기능 Test 1 – Negative

Test case object	아이트래킹 API가 입력 받은 Webcam의 데이터에서 사용자의 시선을 인식하지 못할 때의 결과를 테스트한다.
테스트 입력 값	<p>Webcam으로부터 받는 사용자 영상 데이터</p> <p>1) 사용자의 얼굴이 나오지 않는 webcam 화면 사용자의 시선이 인식될 수 없는, 얼굴이 없는 webcam 화면에 대해서 정상적인 예외처리를 하는지 확인한다.</p> <p>2) 사용자의 얼굴은 나오지만 눈을 감고 있는 화면 사용자의 시선이 인식될 수 없는, 눈을 감고 있는 webcam 화면에 대해서 정상적인 예외처리를 하는지 확인한다.</p> <p>3) 두 명 이상의 사용자가 나오는 화면 웹캠의 화면에 두 명 이상의 사용자가 나올 때 수업 이용자의 시선을 정상적으로 인식하는지 확인한다.</p> <p>4) 영상 데이터가 아닌 Null 값 Null 값에 대해서 에러를 발생하지 않고 예외 처리를 하는지 확인한다.</p>
기대 결과값	<p>1) ‘사용자의 얼굴이 인식되지 않습니다.’ 출력 사용자의 얼굴이 포함된 영상 데이터가 Webcam으로부터 들어올 때까지 대기한다.</p> <p>2) ‘사용자의 시선이 인식되지 않습니다.’ 출력 사용자의 눈이 포함된 영상 데이터가 Webcam으로부터 들어올 때까지 대기한다.</p> <p>3) ‘두 명 이상의 사용자가 나옵니다.’ 출력 수업에서의 아이트래킹 시스템은 수업 참여자 한 명만 파악할 수 있어야 하기 때문에 두 명 이상의 아이트래킹 사용자는 허용하지 않는다.</p> <p>4) Null 값 인식, 웹캠 Test procedure로 복귀</p>

	Null 값 입력 시, 정상적인 영상 데이터를 얻기 위해 다시 웹캠test procedure로 복귀한다.
발생 가능 error	<ul style="list-style-type: none"> 1) Null 값에 대한 정상적인 처리를 하지 못한다. 2) 사용자의 시선이 잡히지 않아야 하는데 영상의 다른 물체를 기준으로 시선이 잡힐 수 있다.

4.3.7. 시선 인식 기능 Test 2

Test case object	아이트래킹 API가 Roi를 정상적으로 계산하는지 확인한다.
테스트 입력 값	<p>Webcam으로부터 받는 사용자의 영상 데이터</p> <ul style="list-style-type: none"> 1) 사용자의 얼굴 및 눈동자가 나오지 않는 영상 사용자의 얼굴 및 눈동자가 나오지 않을 때의 예외 처리를 정상적으로 하는지 확인한다. (Null) 2) 사용자의 얼굴과 눈동자가 나오는 영상 사용자의 눈동자의 좌표 데이터가 정상적으로 계산되는지 확인한다.
기대 결과값	<ul style="list-style-type: none"> 1) ‘사용자의 얼굴이 인식되지 않습니다.’ 출력 사용자의 얼굴이 포함된 영상 데이터가 Webcam으로부터 들어올 때까지 대기한다. 2) Frame 범위에 포함된 눈동자의 Roi(Region of Interest) 좌표 데이터
발생 가능 error	<ul style="list-style-type: none"> 1) Interface misuse: 데이터를 Webcam가 아닌 다른 component로부터 받아온다. 2) Interface misunderstanding: Frame 범위에서 벗어난 결과가 발생한다.

4.3.8. 시선 인식 기능 Test2 – Negative

Test case object	아이트래킹 API가 Roi 계산을 위한 데이터를 받지 못했을 때의 결과를 확인한다.
테스트 입력 값	Webcam으로부터 받는 사용자의 영상 데이터 1) 사용자의 얼굴 및 눈동자가 나오지 않는 영상 사용자의 얼굴 및 눈동자가 나오지 않을 때의 예외 처리를 정상적으로 하는지 확인한다. (Null)
기대 결과값	1) ‘사용자의 얼굴이 인식되지 않습니다.’ 출력 사용자의 얼굴이 포함된 영상 데이터가 Webcam으로부터 들어올 때까지 대기한다.
발생 가능 error	1) Null 값에 대한 정상적인 처리를 하지 못한다. 2) 사용자의 시선이 잡히지 않아야 하는데 영상의 다른 물체를 기준으로 시선이 잡힐 수 있다.

4.3.9. 색상 데이터 제거 Test

Test case object	트래킹에 필요 없는 데이터 제거를 위한 색상 데이터 제거를 test한다.
테스트 입력 값	Webcam으로부터 받는 색상이 포함된 영상 데이터 1) 정상적인 컬러 영상 데이터 컬러 값의 중간 (0~255 중 128) 값을 가지는 컬러 영상 데이터 입력을 통해 색상 데이터가 정상적으로 제거되는지 확인한다. 2) 색상이 없는 흑백 영상 데이터 (extreme ends of range: 0/255)

기대 결과값	1) 색상 데이터가 제거된 영상 데이터 생성 후 눈동자 위치 검출 Procedure call 2) 색상 데이터가 제거된 영상 데이터 생성 후 눈동자 위치 검출 Procedure call
발생 가능 error	1) 색상 데이터의 제거 과정에서 영상의 다른 데이터도 제거한다. 2) Timing errors: 눈동자 위치 검출 procedure call이 지연되어 과거의 데이터를 넘겨준다.

4.3.10. 색상 데이터 제거 Test – Negative

Test case object	색상 데이터 제거를 위한 영상 데이터가 들어오지 않았을 때의 결과를 확인한다.
테스트 입력 값	영상 데이터가 아닌 Null 값
기대 결과값	1) Null 값 인식, 웹캠 Test procedure로 복귀 Null 값 입력 시, 정상적인 영상 데이터를 얻기 위해 다시 test procedure로 복귀한다.
발생 가능 error	1) Null 값을 인식하지 못하고 데이터의 입력을 무한정 대기할 수 있다.

4.3.11. 눈동자 위치 검출 Test

Test case object	아이트래킹을 위한 사용자의 눈동자 위치 검출을 test한다.
테스트 입력 값	색상 데이터가 제거된 영상 데이터 1) 화면의 끝, 즉 모서리 4개에 눈동자가 위치한 영상 데이터 (Extreme ends of ranges) 2) 화면의 중앙에 눈동자가 위치한 영상 데이터
기대 결과값	1) 사용자 눈동자의 위치 데이터
발생 가능 error	1) Roi에서 지정한 좌표에서 벗어난 위치 데이터가 발생한다.

4.3.12. 눈동자 위치 검출 Test – Negative

Test case object	아이트래킹을 위한 사용자의 눈동자 위치 검출을 하지 못할 경우의 결과를 test한다.
테스트 입력 값	색상 데이터가 제거된 영상 데이터 1) 화면의 끝, 즉 모서리 4개에 눈동자가 결치도록 위치한 영상 데이터 (Extreme ends of ranges) 2) 눈동자가 없는 영상 데이터 (Null)
기대 결과값	1) 눈동자가 인식되지 않을 경우 Null과 동일한 처리 2) Null 값 인식, 시선 인식 Test procedure로 복귀 Null 값 입력 시, 정상적인 영상 데이터를 얻기 위해 다시 시선 인식 Test procedure로 복귀 한다.

발생 가능 error	1) Null에 대한 정상적인 처리를 하지 않는다. 2) 눈동자가 인식되지 않는데 Null로 인식하지 않는다.
----------------	--

4.3.13 노이즈 제거 Test

Test case object	아이트래킹의 정확도 향상을 위한 영상 데이터 노이즈 제거
테스트 입력 값	영상 데이터 1) 색상 데이터가 포함된 영상 데이터 2) 색상 데이터가 제거된 영상 데이터 3) Null 값
기대 결과값	1) 색상 데이터 제거를 위해 색상 데이터 제거 procedure로 복귀 색상이 제거되지 않은 데이터를 추후 이용하면 문제가 발생할 수 있으므로 색상 데이터 제거 procedure로 복귀한다. 2) 블러 처리된 영상 데이터 3) Null 값 인식, 웹캠 Test procedure로 복귀 Null 값 입력 시, 정상적인 영상 데이터를 얻기 위해 다시 test procedure로 복귀한다.
발생 가능 error	1) 노이즈 제거 과정에서 필요한 데이터가 제거된다.

4.3.14. 중심선 그리기 Test

Test case object	눈동자의 중심선 그리기 Test
테스트 입력 값	<p>Webcam을 통해 받은 영상 데이터</p> <ul style="list-style-type: none"> 1) 화면의 끝, 즉 모서리 4개에 눈동자가 위치한 영상 데이터 (Extreme ends of ranges) 2) 화면의 중앙에 눈동자가 위치한 영상 데이터 3) 눈동자가 없는 영상 데이터 (Null)
기대 결과값	<ul style="list-style-type: none"> 1) 사용자 눈동자의 중심점 좌표 데이터 및 중심선 2) 사용자 눈동자의 중심선 데이터 3) Null 값 입력 시, 웹캠 Test procedure call <p>Null 값 입력 시, 정상적인 영상 데이터를 얻기 위해 다시 시선인식 test procedure로 복귀한다.</p>
발생 가능 error	<ul style="list-style-type: none"> 1) Roi에서 지정한 프레임의 좌표에서 벗어난 위치 데이터가 발생한다.

4.3.15. 영상 크기에 따른 눈동자 인식 Test

Test case object	아이트래킹의 영상 크기에 따른 정확도 Test
테스트 입력 값	<p>영상 크기에 따른 사용자 영상 데이터</p> <ul style="list-style-type: none"> 1) 사용자의 얼굴이 영상에 다 포함되지 못하며 눈동자가 나오는 영상 2) 사용자의 얼굴이 영상의 전체에 꽉 차 있는 영상 3) 사용자의 얼굴이 영상의 절반을 차지하는 영상 4) 사용자의 얼굴이 영상 크기의 1%를 차지하는 영상

	5) 사용자의 얼굴이 없는 영상 (Null)
기대 결과값	1) 사용자의 얼굴 인식 불가, 인식될 때까지 대기 2) 사용자의 얼굴 및 눈동자 인식 3) 사용자의 얼굴 및 눈동자 인식 4) 사용자의 얼굴 및 눈동자 인식 불가, 인식될 때까지 대기 5) 사용자의 얼굴 및 눈동자 인식 불가, 인식될 때까지 대기
발생 가능 error	1) 영상에서의 사용자 얼굴 크기에 따른 인식 불가 error 발생 가능

5. Supporting Information

5.1 Document History

[Table 3] Document History

Date	Version	Description	Writer
2020/05/24	0.1	Style	Lee, Bohyun
2020/05/24	1.0	Addition of 3.1	Jo, Hyojung
2020/05/25	1.1	Addition of 1, 4.1	Rim, Seungjae
2020/05/25	1.2	Addition of 4.3	Lee, Jaehun
2020/05/26	1.3	Addition of 2	Jo, Hyojung
2020/05/26	1.4	Addition of 3.2	Kang, Byeongnam
2020/05/26	1.5	Addition of 4.2	Lee, Bohyun
2020/05/27	1.6	Revision of 2	Lee, Jaehun
2020/05/27	1.7	Revision of 2	Kang, Byeongnam
2020/05/28	1.8	Revision of 2	Jo, Hyojung
2020/05/28	1.9	Revision of 2	Rim, Seungjae
2020/05/28	2.0	Revision of 2, final	Lee, Bohyun