**Kingo Learning**

**Test Plan Specification**

**Team 12**

**팀장 조건희**

**조원 박성빈**

**조원 박민지**

**조원 송재현**

**조원 이태기**

**조원 김정안**

Contents

**1. Introduction6**

1.1. Purpose6

1.2. Scope6

1.3. Definitions, Acronyms, and Abbreviation6

1.4. References7

1.5. Overview7

**2. Approach7**

2.1. Test methods8

2.1.1. Software Unit Test Methods8

2.1.1.1. Objectives8

2.1.1.2. Pseudo Code8

2.1.2. Software Interface Test Methods14

2.1.2.1. Objectives14

2.2.2.2. Pseudo Code15

**3. Software Unit Test18**

3.1. 로그인18

3.1.1. GLS 로그인 Test Case18

3.1.2. GLS 로그인 Test Case - Negative18

3.1.3. GLS 로그인 Test Case – Repeated Same Input18

3.2. 메인 페이지18

3.2.1. 사용자 수강 정보 Test Case18

3.2.2. 사용자 수강 정보 Test Case - Negative18

3.2.3. 과목 정보 Test Case19

3.2.4. 과목 정보 Test Case - Negative19

3.2.5. 페이지 이동 Test Case19

3.2.6. 페이지 이동 Test Case - Negative19

3.3. 강의 상세 페이지19

3.3.1. 탭 메뉴 불러오기 Test Case19

3.3.2. 탭 메뉴 불러오기 Test Case - Negative19

3.3.3. 탭 메뉴 이동 Test Case19

3.3.4. 탭 메뉴 이동 Test Case - Negative20

3.4. 실시간 강의20

3.4.1. 카메라 접근 권한 Test Case20

3.4.2. 카메라 접근 권한 Test Case - Negative20

3.4.3. 카메라 화면 공유 Test Case20

3.4.4. 카메라 연결 Test Case - Negative20

3.4.5. 실시간 강의 자동 출석 Test Case20

3.4.6. 실시간 강의 자동 출석 Test Case - Negative21

3.4.7. 강의룸 채팅 Test Case21

3.4.8. 강의룸 채팅 Test Case - Negative21

3.5. 익명게시판21

3.5.1. 사용자가 게시판에 글을 업로드하려는 경우21

3.5.2. 비정상적 접근으로 글을 업로드하려는 경우21

3.5.3. 사용자가 게시판 글에 댓글을 업로드하려는 경우21

3.5.4. 비정상적 접근으로 댓글을 업로드하려는 경우22

3.6. 채팅방22

3.6.1. 사용자가 채팅방에 입장하려는 경우22

3.6.2. 사용자가 채팅방에 글을 업로드하려는 경우22

3.6.3. 사용자가 채팅방 내에서 수강생 목록을 확인하는 경우22

3.6.4. 사용자가 푸쉬알람을 설정함22

3.6.5. 사용자가 푸쉬알람을 해제함22

3.7. 할 일22

3.7.1. 사용자가 관련 과목을 통해 할일 목록을 확인하려는 경우22

3.7.2. 할일을 완료하면 완료상태 표시23

3.8. 캘린더23

3.8.1. 사용자가 캘린더를 확인하고 싶은 경우23

3.8.2. 사용자가 일정을 추가하고 싶은 경우23

3.8.3. 사용자가 일정을 삭제하고 싶은 경우23

3.8.4. 사용자가 할일의 날짜를 확인하고 싶은 경우23

3.9. 알림23

3.9.1. 강의/공지/과제가 업로드 된 Test Case23

3.9.2. 강의/공지/과제가 업로드 된 Test Case - Negative23

**4. Software Interface Test24**

4.1. Login Test24

4.1.1. 로그인 승인 시간 Test Case24

4.1.2. 로그인 승인 시간 Test Case - Negative24

4.1.3. 로그인 유지 기능 Test Case24

4.1.4. 로그아웃 Test Case24

4.2. Real-Time Streaming Lecture Connecting Test24

4.2.1. 강의 전달 시간 Test Case24

4.2.2. 강의 채팅 속도 Test Case - Negative25

4.2.3. 강의 전달 끊김 현상 Test Case25

4.3. Data Delivery Test25

4.3.1. 적절한 사용자 정보 전달 Test Case25

4.3.2. 데이터 업로드 Test Case25

4.4. Alarm Test25

4.4.1. 강의자 컨텐츠 업로드 알림 Test Case25

4.4.2. 할 일&캘린더 알림 Test Case26

**5. Supporting Information26**

5.1. Document History26

1. **Introduction**

Introduction 장은 본 Test Plan 문서의 목적, 문서의 범위와 테스트의 범위, 문서를 작성하며 사용한 낯선 단어, 참조한 문서, Test Plan 문서의 전체적인 내용에 대해 짤막하게 설명한다.

**1.1. Purpose**

본 Test Plan 문서는 기본적으로 요구사항 명세서에서 계획했던 테스트를 더 구체화 한다. Test Plan 문서를 작성하는 목적은 테스트를 진행할 때 어떻게 해야할 지에 대한 자세한 설명을 남기기 위해서이다. 특히 이 문서로 개발 팀과 테스트 팀의 분업을 도울 뿐만 아니라, 팀원들이 테스트를 통일성 있게 진행할 수 있어 팀원들 사이에 원활한 의사소통이 가능해진다.

**1.2. Scope**

본 Test Plan 문서에는 우선 킹고 러닝 프로젝트에서 테스트를 진행할 때 필요한 기본적인 정보를 제공해 준다. 이 정보에는 테스트 실행을 위한 명령어, 각 테스트의 목적과 해당 테스트 시에 사용할 입력 값과 출력 값 쌍들, 테스트의 작동 방식이 담긴 추상화된 수도 코드들이 포함된다.

Unit Test와 Interface Test, 이렇게 두가지 종의 테스트를 진행할 것이다. 시스템을 구성하는 코드들을, 기능이 존재하는 최소한의 단위로 쪼개 모든 기능에 대해 하나씩 Unit Test를 진행한다. Interface Test는 시스템이 사용하고 있는 모든 External Interface에 대해 진행한다.

**1.3. Definitions, Acronyms, and Abbreviation**

|  |  |
| --- | --- |
| 팝업창 | Popup을 의미한다. |
| 수강정보 | 사용자가 현재 수강중인 과목에 대한 정보를 의미한다. |
| 수업자료실 | 수업과 관련된 자료(교안 등)을 받을 수 있는 페이지를 의미한다. |
| 강의룸 | 실시간 강의가 진행되고 있는 페이지를 의미한다. |
| 푸쉬알람 | Push Notification을 의미한다. |
| todo | 할 일의 리스트를 의미한다. |

**1.4. References**

본 Test Plan 문서를 작성하며 참고한 자료는 아래와 같다.

* *Test plan document format*
* Ian Sommerville, *Software Engineering*
* IEEE Std 830-1998

**1.5. Overview**

Test Plan 문서는 해당 장인 Introduction을 포함해 총 5개의 장으로 구성된다. 2장, 3장, 4장은 전부 Test Case에 관한 장이다. 2장은 Test Case를 추상적인 수도 코드를 이용해 어떤 방식으로 테스트가 작동할 지에 대해 보여준다. 3장과 4장은 각각 Unit Test, Interface Test에 대해 기술하는데, 어떤 목적을 가지고 해당 Test Case를 만들었는지, 특정 Input에 대해 나와야 하는 Output이 어떻게 되는지 쌍을 지어 자연어로 서술한다. 5장에서는 수정이 생길 때 마다 수정 이력에 대해 기록해둔다.

1. **Approach**

Approach 장은 Test Case를 만들고 실제로 Test 할 때, 어떤 식으로 시험을 할 것인지에 대해 다룬다. 시스템을 팀이 직접 구현한 것과, 외부의 시스템을 차용한 것 두 가지로 나누어서 살펴볼 것이다. 팀이 직접 구현한 것은 기능적인 부분이 올바르게 구현되어 있는가에 대해 중점을 둬야 할 것이다. 그리고 외부의 시스템을 부분적으로 차용하였다면 외부의 시스템과 잘 소통하고 있는지에 중점을 둬야 할 것이다.

Approach 장에서는 중점을 둔 내용을 바탕으로 어떤 식으로 test를 진행할 것인지 대략적인 이해를 돕기 위해 추상화된 Pseudo Code를 이용해 설명한다. Input과 Output, Test Case와 그 목적에 대한 더 자세한 설명은 3과 4 장에서 다룰 예정이다.

**2.1. Test Methods**

미리 만들어둔 input과 output 쌍들을 바탕으로, 특정 input 값들을 넣었을 때, 나와야 하는 output 값이 나오는지를 바탕으로 통과했는지, 실패하였는지 알아본다. 성공해야 하는 input에서 성공하고, 실패해야 하는 input에서 실패했다면 올바르게 작동하고 있는 것이다. 아래는 주요 기능들과 외부 인터페이스들을 테스트하기 위한 수도코드이다.

* + 1. **Software Unit Test Methods**
       1. **Objectives**

Software Unit Test는 시스템 내의 클래스, 메소드 등이 각각 잘 작동하는지 테스트하는 단계이다. 이를 검증하기 위한 대표적인 방법으로는,

* Normal한 상황에서 의도한 대로 작동하는지 검사하기
* Input 형식에 맞지 않는 데이터 넣어서 오류가 발생하는지 체크하기
* 같은 input 데이터를 반복적으로 넣어 실행하기

등이 존재한다.

* + - 1. **Pseudo Code**

<로그인 test>

main:

Show login interface

Execute Login\_test\_1

Execute Login\_test\_2

Execute Login\_test\_3

Execute Login\_test\_4

Login\_test\_1:

Set virtual user information

Try to login with correct id & correct password

Check whether print correct message

Login\_test\_2:

repeat 100 times:

Try to login with wrong id & correct password

Check whether print wrong message or system error

Login\_test\_3:

repeat 100 times:

Try to login with correct id & wrong password

Check whether print wrong message or system error

Login\_test\_4:

repeat 100 times:

Try to login with wrong id & wrong password

Check whether print wrong message or system error

// 잘못된 Input의 경우, 어느 부분이 틀렸는지에 따라 에러 문구가 달라져야 한다.

<강의 페이지 test>

main:

Set virtual user information

Check system shows correct interface correspond to user information

If interface is correct:

repeat for subject\_list:

Show main page interface

Click lecture

Check enter to class page well

Check whether print error if student\_id is wrong // ex) student\_id = -1

<실시간 강의 test>

main:

Set several virtual user information // 카메라의 경우 임의의 이미지로 설정

Set student\_list with virtual users

Execute real\_time\_test\_1

Execute real\_time\_test\_2

Execute real\_time\_test\_3

Execute real\_time\_test\_4

Execute real\_time\_test\_5

real\_time\_test\_1:

Show students\_list

Do attendance check

repeat for student\_list:

Turn on student’s camera

Check whether Student’s attendance is reflected well

real\_time\_test\_2:

for one student:

while(True):

Write same chat & check error for so many chatting

real\_time\_test\_3:

Set wrong students information // student\_id = -1 or student\_list = Null…

Check whether print error when print student\_list, turn on camera or mic, chatting for wrong student information

real\_time\_test\_4:

for one student:

Check whether print error when chatting with no contents

real\_time\_test\_5:

Assume wrong condition about mic & camera

// students try to turn on mic without allow mic permissions &  
students try to turn on camera without allow camera permissions

Check whether print error

<익명 게시판 test>

main:

Set virtual writer student information

Set virtual commenter student information

Execute board\_test\_1

Execute board\_test\_2

Execute board\_test\_3

Execute board\_test\_4

Execute board\_test\_5

board\_test\_1:

Check writer can upload a post fine

Check commenter can write a comment fine

board\_test\_2:

Check whether print error when writer try to upload a post with no content

board\_test\_3:

Check whether print error when commenter try to write a comment with no content

board\_test\_4:

Check whether print error when commenter try to write a comment a lot of time

board\_test\_5:

Set wrong student information

Check whether print error when writer or commenter is wrong student

<채팅방 test>

main:

Set several virtual students information

chat\_test\_1

chat\_test\_2

chat\_test\_3

chat\_test\_4

chat\_test\_5

chat\_test\_6

chat\_test\_7

chat\_test\_1:

Make chatting room with one or more members

Check members can send a chat fine

chat\_test\_2:

Check student enter to chatting room fine if user is student of lecture which corresponding to the chat room

chat\_test\_3:

Turn on student’s receiving push

Check push sending is working fine depending on member’s agree to receive push

Turn off student’s receiving push

Check push sending is working fine depending on member’s agree to receive push

chat\_test\_4:

Check print member’s list fine if click member list

chat\_test\_5:

Set wrong student information

Check whether print error when make chatting room with wrong student

chat\_test\_6:

Check whether print error when try to send a chat with no comment

chat\_test\_7:

Check whether print error when user who is not student of lecture which   
corresponding to the chat room try to enter the chatting room

<할 일 test>

main:

Set virtual student & todo information

Execute todo\_test\_1

Execute todo\_test\_2

todo\_test\_1:

Check show student’s todo\_list fine if click Todo

todo\_test\_2:

Check subject is deleted fine when student complete it

<캘린더 test>

main:

Set virtual student & calendar information

Execute calendar\_test\_1

Execute calendar\_test\_2

Execute calendar\_test\_3

Execute calendar\_test\_4

Execute calendar\_test\_5

calendar\_test\_1:

Check showing a student’s calendar is working fine

calendar\_test\_2:

Check adding a new schedule and reflect this on calendar are working fine

calendar\_test\_3:

Check deleting a new schedule and reflect this on calendar are working fine

calendar\_test\_4:

Check showing todo list on a certain date is working fine if click any date in calendar

calendar\_test\_5:

Check whether print error when adding a new schedule with no content

<알림 test>

main:

Set virtual student & upload information

for all upload information:

Execute push\_test\_1

push\_test\_1:

if lecture is uploaded & student agree to receive lecture push

Check sending a push working fine // it must be push about lecture

else if notice is uploaded & student agree to receive notice push

Check sending a push working fine // it must be push about notice

else if assignment is uploaded & student agree to receive assignment push

Check sending a push working fine // it must be push about assignment

* + 1. **Software Interface Test Methods**
       1. **Objectives**

Software Interface Test는 Integration Test 한 부분으로, 시스템이 차용한 외부 인터페이스의 오류나, 인터페이스의 작동 방식에 대한 잘못된 예측으로 생긴 오류가 없는지 찾아야하는 단계이다. 이를 검증하기 위한 대표적인 방법으로는,

* 인자에 양 극한의 값 전달하기
* 인자에 널 포인터 전달하기
* 무리한 상황 설정하기

등이 존재한다.

* + - 1. **Pseudo Code**

실행 명령어는 다음과 같은 꼴이다.

$ test interface “interface name” <command>

<로그인 테스트>

login\_test(input set, output set)

get database information from system

looping the set of inputs

print success if log in success with correct input

print fail if log failed with incorrect input

re-refer the login page

print success if it auto logged in with auto login checked

print failed if it didn’t auto log in when auto login unchecked.

else, something is wrong

<실시간 강의 접속 테스트>

rt\_lecture\_connection\_test(input set, output set)

get server information from the system

when command is

none, do everything below

ping, ping\_request(server)

chat, chat\_test(server)

delay, delay\_test(server)

ping\_request

send packet to server and record the current time(time1)

get packet from server and record the current time(time2)

print (time2 – time1)

chat\_test

make a chat as a student1 and record the current time(time1)

record the time when chat is received as student2(time2)

print (time2 – time1)

delay\_test

record the time when sending a frame as a lecturer(time1)

record the time when receiving a frame as a student(time2)

print (time2 – time1)

when time2 – time1 is over 1ms, something is wrong

<데이터 전송 테스트>

data\_delivery\_test(input set, output set)

when command is

none, do everything below

account, account\_test

text, text\_upload\_test

file, file\_upload\_test

contents, contents\_upload\_test

todo, todo\_set\_test

data\_request(database, target)

access the given database

find the target

if exist, return the found target

else, return null

change\_request(database, target)

access the given database

add a new data or subtract target in list

check if the change has been made

if change made as intended,

print change success message and changed item

else, something is wrong

account\_test

loop the input set

data\_request(student database, target)

if id and password same as data,

print out student’s information

if id or password isn’t same as data,

print out login failed message

else something is wrong

text\_upload\_test

change\_request(database, text)

file\_upload\_test

change\_request(database, file)

contents\_upload\_test

change\_request(database, contents)

check notification has made as a student

if it has made, print success message

else, something is wrong

todo\_set\_test

change\_request(database, todo)

check notification has made as a student

if it has made, print success message

else, something is wrong

**3. Software Unit Test**

**3.1. 로그인**

**3.1.1 GLS 로그인 test case**

Test case object: 유효한 입력 값이 들어왔을 때 로그인 function이 올바르게 작동하는지 확인한다.

Test Inputs: 사용자의 아이디와 비밀번호

Expected Results: 로그인 성공 메시지를 띄운다.

**3.1.2 GLS 로그인 test case – Negative**

Test case object: 유효하지 않은 입력 값이 들어왔을 때 로그인 function이 올바르게 작동하는지 확인한다.

Test Inputs: 존재하지 않는 아이디 혹은 잘못된 비밀번호

Expected Results: 로그인 실패 메시지와 함께 로그인을 실패한 원인을 팝업창으로 보여준다.

**3.1.3 GLS 로그인 test case – Repeated same input**

Test case object: 동일한 입력 값이 반복적으로 들어왔을 때 로그인 function이 올바르게 작동하는지 확인한다.

Test Inputs: 존재하지 않는 아이디 혹은 잘못된 비밀번호

Expected Results: 로그인 실패 메시지를 띄운다. 만약 5회 이상 잘못된 입력이 반복적으로 들어왔다면 악성 사용자로 판단하고 30분간 사용자의 로그인 시도를 금지한다.

**3.2. 메인 페이지**

**3.2.1. 사용자 수강 정보 test case**

Test case object: 사용자의 아이디에 해당하는 수강정보를 올바르게 불러오는지 확인한다.

Test Inputs: 사용자의 아이디

Expected Results: 사용자가 현재 수강중인 과목들의 리스트를 보여준다.

**3.2.2. 사용자 수강 정보 test case – Negative**

Test case object: 사용자의 아이디에 해당하는 수강정보를 불러오지 못한 경우를 확인한다.

Test Inputs: 사용자의 아이디

Expected Results: 과목 리스트 불러오기를 실패했다는 메시지를 띄운다.

**3.2.3. 과목 정보 test case**

Test case object: 사용자가 수강하는 과목의 정보를 보여준다.

Test Inputs: 사용자의 아이디, 과목명

Expected Results: 해당 과목을 수강하는 학생이 들어야 하는 잔여 강의 수와 제출해야 하는 잔여 과제 수를 보여준다.

**3.2.4. 과목 정보 test case – Negative**

Test case object: 해당 과목을 수강하는 학생의 정보를 찾을 수 없는 경우를 확인한다.

Test Inputs: 사용자의 아이디, 과목명

Expected Results: 현재 수강 중이지 않는 과목이라는 메시지를 띄운다.

**3.2.5. 페이지 이동 test case**

Test case object: 클릭한 과목 페이지로 올바르게 이동하는지 확인한다.

Test Inputs: 과목명

Expected Results: 클릭한 과목 페이지로 이동한다.

**3.2.6. 페이지 이동 test case – Negative**

Test case object: 이동하려는 과목 페이지가 존재하지 않는 경우를 확인한다.

Test Inputs: 과목명

Expected Results: 페이지 이동에 실패했다는 메시지를 띄운다.

**3.3. 강의 상세 페이지**

**3.3.1. 탭 메뉴 불러오기 test case**

Test case object: 각 강의 메뉴의 항목을 리스트로 불러올 수 있는지 확인한다.

Test Inputs: 과목명

Expected Results: 해당하는 과목에 설정된 탭 메뉴 항목을 리스트로 불러온다.

**3.3.2. 탭 메뉴 불러오기 test case – Negative**

Test case object: 각 강의 메뉴의 항목을 리스트로 불러오기를 실패한 경우를 확인한다.

Test Inputs: 과목명

Expected Results: 디폴트 값으로 설정된 탭 메뉴 리스트를 보여준다.

**3.3.3. 탭 메뉴 이동 test case**

Test case object: 공지, 수업자료실, 익명 게시판, 강의 컨텐츠, 과제, 시험 등으로 이루어진 탭 메뉴를 클릭했을 때 올바른 페이지로 이동하는지 확인한다.

Test Inputs: 과목명, 탭 메뉴의 이름

Expected Results: 클릭한 탭 메뉴의 페이지로 이동한다.

**3.3.4. 탭 메뉴 이동 test case – Negative**

Test case object: 탭 메뉴를 클릭했을 때 해당 페이지로 이동할 수 없는 경우를 확인한다.

Test Inputs: 과목명, 탭 메뉴의 이름

Expected Results: 페이지 이동에 실패했다는 메시지를 띄운다.

**3.4. 실시간 강의**

**3.4.1. 카메라 접근 권한 test case**

Test case object: 사용자가 카메라 접근 권한을 허용하고 카메라 기능을 이용하려 할 때 올바르게 작동하는지 확인한다.

Test Inputs: 접근 권한 정보

Expected Results: 연결 성공 메시지를 띄운다.

**3.4.2. 카메라 접근 권한 test case – Negative**

Test case object: 사용자가 카메라 접근 권한을 허용하지 않고 카메라 기능을 이용하려고 하는 경우를 확인한다.

Test Inputs: 접근 권한 정보

Expected Results: 연결 실패 메시지와 함께 카메라 접근 권한을 허용하라는 팝업창을 띄운다.

**3.4.3. 카메라 화면 공유 test case**

Test case object: 사용자의 카메라 화면이 올바르게 공유되고 있는지 확인한다.

Test Inputs: 카메라 화면

Expected Results: 공유되고 있는 사용자의 카메라 화면을 성공적으로 띄운다.

**3.4.4. 카메라 연결 test case – Negative**

Test case object: 사용자의 카메라 연결 상태에 문제가 생긴 경우를 확인한다.

Test Inputs: 카메라 화면

Expected Results: 문제가 발생했다는 메시지를 띄운다.

**3.4.5. 실시간 강의 자동 출석 test case**

Test case object: 수업 시작 시간 10분 이내로 강의룸에 접속하여 실시간 강의를 끝까지 수강한 사용자의 경우를 확인한다.

Test Inputs: 사용자의 접속 로그

Expected Results: 해당 사용자의 출석 현황을 출석으로 표시한다.

**3.4.6. 실시간 강의 자동 출석 test case – Negative**

Test case object: 실시간 수업 도중에 강의룸을 나오거나 강의룸에 입장하지 않은 사용자의 경우를 확인한다.

Test Inputs: 사용자의 접속 로그

Expected Results: 해당 사용자의 출석 현황을 결석으로 표시한다.

**3.4.7. 강의룸 채팅 test case**

Test case object: 사용자가 강의룸에 채팅창에 채팅을 작성한다.

Test Inputs: 사용자의 학번, 사용자의 이름, 채팅 작성 시간, 텍스트 내용

Expected Results: 채팅창에 사용자의 정보와 함께 입력한 텍스트 내용이 올라온다.

**3.4.8. 강의룸 채팅 test case – Negative**

Test case object: 사용자가 강의룸에 채팅 메시지를 작성하였으나 네트워크 연결 문제 등이 발생한다.

Test Inputs: 사용자의 학번, 사용자의 이름, 채팅 작성 시간, 텍스트 내용

Expected Results: 실패 메시지를 띄운다.

**3.5 익명게시판**

**3.5.1. 사용자가 게시판에 글을 업로드하려는 경우**

Test case object: 정상적인 사용자가 글을 작성한 내용이 서버에 업로드 후, 사용자에게 보여지는지 여부

Test Inputs: 정상적인 사용자의 작성 글

Expected Result: 작성된 글이 보여짐

**3.5.2. 비정상적 접근으로 글을 업로드하려는 경우**

Test case object: 비정상적인 사용자가 글을 작성하려 시도하는 경우 접근차단

Test Inputs: 비정상적인 사용자의 게시판 글 작성 시도

Expected Result: 접근 불가 메시지 현출

**3.5.3. 사용자가 게시판 글에 댓글을 업로드하려는 경우**

Test case object: 정상적인 사용자가 댓글을 작성한 내용이 서버에 업로드 후, 사용자에게 보여지는지 여부

Test Inputs: 정상적인 사용자의 작성 댓글

Expected Result: 작성된 댓글이 보여짐

**3.5.4. 비정상적 접근으로 댓글을 업로드하려는 경우**

Test case object: 비정상적인 사용자가 댓글을 작성하려 시도하는 경우 접근차단

Test Inputs: 비정상적인 사용자의 게시판 댓글 작성 시도

Expected Result: 접근 불가 메시지 현출

**3.6 채팅방**

**3.6.1. 사용자가 채팅방에 입장하려는 경우**

Test case object: 권한있는 사용자가 채팅방 입장을 허용함

Test Inputs: 권한있는 사용자가 채팅방을 클릭함

Expected Result: 수강생 목록과 채팅내역을 불러옴

**3.6.2. 사용자가 채팅방에 글을 업로드하려는 경우**

Test case object: 정상적인 사용자가 채팅방에 글을 작성하려 시도하는 경우

Test Inputs: 정상적인 사용자의 채팅방 글 작성 시도

Expected Result: 작성한 글이 보여짐

**3.6.3. 사용자가 채팅방 내에서 수강생목록을 확인하는 경우**

Test case object: 사용자가 채팅방에서 수강생 목록을 확인함

Test Inputs: 사용자가 수강생 목록을 클릭함

Expected Result: 익명화된 수강생 목록이 불려져옴

**3.6.4. 사용자가 푸쉬알람을 설정함**

Test case object: 푸쉬알람이 해제된 상태에서 설정상태로 변경함

Test Inputs: 사용자가 푸쉬버튼을 클릭함

Expected Result: 푸쉬알람이 설정됨

**3.6.5. 사용자가 푸쉬알람을 해제함**

Test case object: 푸쉬알람이 설정된 상태에서 해제상태로 변경함

Test Inputs: 사용자가 푸쉬버튼을 클릭함

Expected Result: 푸쉬알람이 해제됨

**3.7 할 일**

**3.7.1. 사용자가 관련 과목을 통해 할일 목록을 확인하려는 경우**

Test case object: 할일목록에서 관련과목 클릭시 해당 과목으로 이동

Test Inputs: 사용자가 할일목록에서 특정 할일을 클릭

Expected Result: 할일이 소속된 과목으로 이동

**3.7.2 할일을 완료하면 완료상태 표시**

Test case object: 사용자가 할일을 완료한 경우 완료상태로 변경 후 표시

Test Inputs: 사용자가 할일목록에서 완료 클릭

Expected Result: 해당 할일은 완료로 표시 후 할일 목록에서 삭제

**3.8 캘린더**

**3.8.1. 사용자가 캘린더를 확인하고 싶은 경우**

Test case object: 사용자가 캘린더 항목을 클릭하면 캘린더를 로드

Test Inputs: 사용자가 캘린더 항목을 클릭

Expected Result: 캘린더 로드

**3.8.2. 사용자가 일정을 추가하고 싶은 경우**

Test case object: 사용자가 캘린더에서 특정 날짜 클릭 시 일정 추가

Test Inputs: 사용자가 캘린더에서 특정 날짜 클릭

Expected Result: 일정 추가 양식 로드

**3.8.3. 사용자가 일정을 삭제하고 싶은 경우**

Test case object: 사용자가 캘린더에서 특정 날짜의 일정을 클릭하면 삭제 여부 확인

Test Inputs: 사용자가 캘린더에서 특정 날짜 일정 클릭

Expected Result: 삭제 여부 확인 메시지 표시

**3.8.4. 사용자가 할일의 날짜를 확인하고 싶은 경우**

Test case object: 사용자가 캘린더의 할일을 클릭하면 지정된 날짜가 표시

Test Inputs: 사용자가 캘린더에서 특정 일정을 클릭

Expected Result: 일정의 지정된 날짜가 표시

**3.9. 알림**

**3.9.1. 강의/공지/과제가 업로드 된 test case**

Test case object: 사용자가 해당 과목의 알림을 on으로 설정하고 새로운 강의/공지/과제가 과목 페이지에 업로드 되었을 때 올바르게 푸시 알림이 울리는지 확인한다.

Test Inputs: 과목명, 알림 내용, 사용자의 알림 설정 여부

Expected Result: 사용자는 알림 내용이 포함된 푸시 알림을 받는다.

**3.9.2. 강의/공지/과제가 업로드 된 test case – Negative**

Test case object: 사용자가 해당 과목의 알림을 off로 설정하고 새로운 강의/공지/과제가 과목 페이지에 업로드 된 경우를 확인한다.

Test Inputs: 과목명, 사용자의 알림 설정 여부

Expected Result: 사용자는 푸시 알림을 받지 못한다.

**4. Software Interface Test**

**4.1. Login Test**

**4.1.1. 로그인 승인 시간 Test Case**

Test case object: 사용자가 아이디와 비밀번호를 입력하여 로그인을 요청할 때 얼마나 빠르게 로그인을 승인할 수 있는지를 테스트한다.

Test Inputs: 사용자의 아이디와 비밀번호

Expected Results: 로그인이 1ms 안에 해결되어야 한다.

**4.1.2. 로그인 승인 시간 Test Case - Negative**

Test case object: 사용자가 아이디와 비밀번호를 입력하여 로그인을 요청할 때 얼마나 빠르게 잘못된 정보라고 알려줄 수 있는지를 테스트한다.

Test Inputs: 사용자의 아이디와 비밀번호

Expected Results: ‘잘못된 아이디와 비밀번호입니다.’ 라는 팝업창이 1ms 안에 띄워져야 한다.

**4.1.3. 로그인 유지 기능 Test Case**

Test case object: 사용자가 ‘로그인 유지’ 상태를 선택한 후 로그인하고 앱을 종료한다. 그 다음에 다시 접속했을 때 로그인을 생략하고 앱을 사용할 수 있는지를 테스트한다.

Test Inputs: 로그인 유지 상태 체크, 사용자의 아이디와 비밀번호

Expected Results: 로그인 유지

**4.1.4. 로그아웃 Test Case**

Test case object: 사용자가 로그아웃을 했을 때 로그아웃 속도를 측정한다. 또한, 로그인 유지 상태가 해제되는지 확인한다.

Test Inputs: 로그아웃

Expected Results: 로그아웃이 1ms 안으로 작동해야 한다. 로그인 유지 상태는 해제된다.

**4.2. Real-Time Streaming Lecture Connecting Test**

**4.2.1. 강의 전달 시간 Test Case**

Test case object: 강의자가 스트리밍하고 있는 강의가 수강자들에게 전달되는 시간을 테스트한다.

Test Inputs: 강의

Expected Results: 강의가 수강자들에게 1ms 안에 전달되어야 한다.

**4.2.2. 강의 채팅 속도 Test Case**

Test case object: 강의자와 수강자들이 사용하고 있는 채팅방의 채팅이 올라오는 속도를 측정한다.

Test Inputs: 채팅

Expected Results: 채팅을 친 후 1ms 모든 사용자들이 채팅을 볼 수 있다.

**4.2.3. 강의 전달 끊김 현상 Test Case**

Test case object: 음성이나 강의가 끊기지 않게 전달되는지를 테스트한다.

Test Inputs: 강의

Expected Results: 수강자들이 강의를 버퍼링(기다림)없이 들어야 한다.

**4.3. Data Delivery Test**

**4.3.1. 적절한 사용자 정보 전달 Test Case**

Test case object: 사용자가 로그인 시 사용자 정보에 맞는 데이터를 데이터베이스에서 불러올 수 있는지를 확인한다.

Test Inputs: 사용자의 아이디와 비밀번호

Expected Results: 사용자 정보에 맞는 데이터를 1ms 안에 불러와야 한다.

**4.3.2. 데이터 업로드 Test Case**

Test case object: 사용자가 데이터 업로드 시 알맞은 타입의 객체로 데이터베이스에 저장되며, 다른 사용자가 업로드된 데이터를 확인할 수 있는지를 테스트한다.

Test Inputs: 데이터 업로드 요청, 데이터 다운로드 요청

Expected Results: 데이터베이스에 데이터가 저장되고 사용자는 다운 받은 데이터를 확인할 수 있다.

**4.4. Alarm Test**

**4.4.1. 강의자 컨텐츠 업로드 알림 Test Case**

Test case object: 강의자가 컨텐츠 업로드 시 알림 시스템이 컨텐츠의 정보를 전달받고 사용자의 인터페이스에서 알림 기능이 정상적으로 작동하는지를 테스트한다. 단, 사용자가 알림 수신을 허용했을 경우에만이다.

Test Inputs: 데이터 업로드, 사용자 알림 수신 여부

Expected Results: 알림 시스템이 컨텐츠의 정보를 잘 나타내면서 사용자에게 알린다. 단, 사용자가 알림을 허용하지 않았을 때는 무시한다.

**4.4.2. 할 일&캘린더 알림 Test Case**

Test case object: 사용자가 할 일&캘린더 기능에서 설정한 내용을 알림 시스템이 알맞은 데이터 형태로 전달받고 알림 기능이 정상적으로 작동하는지를 테스트한다. 단, 사용자가 알림 수신을 허용했을 경우에만이다.

Test Inputs: 할 일&캘린더, 사용자 알림 수신 여부

Expected Results: 할 일&캘린더 데이터에 따라서 알림 시스템이 사용자에게 알린다. 단, 사용자가 알림을 허용하지 않았을 때는 무시한다.

**5. Supporting Information**

**5.1. Document History**

| **날짜** | **버전** | **설명** | **관련된 사람** |
| --- | --- | --- | --- |
| 2021/05/18 | 0.1 | 역할 분배 | - |
| 2020/05/25 | 0.3 | 3장 추가 | 박민지, 이태기 |
| 2020/05/28 | 0.9 | 2, 4장 추가 | 조건희, 김정안,  박성빈, 송재현 |
| 2020/05/28 | 0.9 | 2, 3, 4장 피드백 회의 | 팀 원 모두 |
| 2020/05/29 | 0.95 | 1, 5장 추가 | 박성빈, 조건희 |
| 2020/05/29 | 1.0 | 문서 정리 및 완성 | 조건희 |