



SKKU Online Reading Room

Test Plan

2021.05.30.

TEAM 8

Team Leader	장영재
Team Member	박윤진
Team Member	김지수
Team Member	박세연

CONTENTS

1. Introduction	4
1.1 Purpose	4
1.2 Scope	4
1.2.1 Test Object	4
1.2.2 Test Items	4
1.3 Definitions, Acronyms, and Abbreviation	4
1.3.1 Unit test	4
1.3.2 Interface test	4
1.3.3 API	4
1.3.4 DB	5
1.4 References	5
1.5 Overview	5
2. Approach	5
2.1 Test method	5
2.1.1 Software unit test methods	5
2.1.2 Software interface test methods	8
2.2 Test tool	8
2.2.1 Android Studio	8
2.2.2 Firebase	8
2.2.3 Ostorlab	8
3. Software Unit Test	8
3.1 Sign-up	9
3.1.1 Application sign-up test case	9
3.1.2. Application sign-up test case – Negative	9
3.2. Sign-in	9
3.2.1. Application sign-in test case	9
3.2.2. Application sign-in test case - Negative	9
3.3. Check information in the database	10
3.3.1. Correct user data and database	10
3.3.2. Incorrect user data and database	10
3.4. Participate study room	10
3.4.1. Participate study room test case	10
3.4.2. Participate study room test case - Negative	10
3.5. Create New study room	11
3.5.1. Create study room	11
3.5.2. Create study room - Negative	11
3.6. Chat service	11
3.6.1. Send correct message in chat service	11
3.6.2. Send incorrect message in chat service	11
4. Software Interface test	12
4.1. Database Interface	12
4.1.1 Connection test	12
4.1.2. Null Pointer Testing	13
4.1.3. Component Failure Testing	13
4.1.4. Stress Testing	13
4.1.5. Shared Memory Testing	14
4.2. Jitsi Video API Interface	14
4.2.1. Connection Test	14

4.2.2. Null Pointer Testing	14
4.2.3. Component Failure Testing	15
4.2.4. Stress Testing	15
4.3. Chat System Interface	15
4.3.1. Connection Test	15
4.3.2. Null Pointer Testing	16
4.3.3. Component Failure Testing	16
4.3.4. Stress Testing	16
4.3.5. Shared Memory Testing	17
5. Supporting Information	17
5.1 Document History	17

LIST OF TABLES

[Table 1] Table of Connection test	12
[Table 2] Table of Null Pointer Testing	13
[Table 3] Table of Component Failure Testing	13
[Table 4] Table of Stress Testing	13
[Table 5] Table of Shared Memory Testing	14
[Table 6] Table of Connection Test	14
[Table 7] Table of Null Pointer Testing	14
[Table 8] Table of Component Failure Testing	15
[Table 9] Table of Stress Testing	15
[Table 10] Table of Connection Test	15
[Table 11] Table of Null Pointer Testing	16
[Table 12] Table of Component Failure Testing	16
[Table 13] Table of Stress Testing	16
[Table 14] Table of Shared Memory Testing	17
[Table 15] Table of Document History	17

1. Introduction

1.1 Purpose

성균관대학교 온라인 독서실 시스템은 성균관대학교 학생들을 대상으로 카메라를 통한 온라인 독서실 서비스를 제공하고 수업별 오픈 채팅 서비스를 제공한다. 이 시스템은 성균관대학교로부터 수업 정보를 불러오고 서비스에 등록된 학생들의 정보를 데이터베이스에 저장하여 학생들에게 유용한 서비스를 무료로 제공하는 것을 목적으로 한다.

1.2 Scope

1.2.1 Test Object

SKKU Online Reading Room 애플리케이션을 대상한 테스트 계획 문서이다.

1.2.2 Test Items

1) Item to be tested

- 기능의 원활한 동작에 대한 기대결과 충족
- 요구사항에 명시된 성능 충족 확인
- 사용자 평가 원칙에 준하여 사용성 평가
- 실사용 환경과 유사한 테스트 환경에서의 정상적인 기능 보장

2) Item not to be tested

- 타 어플리케이션과의 연계성
- 다양한 입력 데이터의 허용성
- OS 별 환경에서의 호환

1.3 Definitions, Acronyms, and Abbreviation

1.3.1 Unit test

유닛 테스트(unit test)는 컴퓨터 프로그래밍에서 소스 코드의 특정 모듈이 의도된 대로 정확히 작동하는지 검증하는 절차다. 즉, 모든 함수와 메소드에 대한 테스트 케이스(Test case)를 작성하는 절차를 말한다. 이를 통해서 언제라도 코드 변경으로 인해 문제가 발생할 경우, 단시간 내에 이를 파악하고 바로 잡을 수 있도록 해준다. 이상적으로, 각 테스트 케이스는 서로 분리되어야 한다. 이를 위해 가짜 객체(Mock object)를 생성하는 것도 좋은 방법이다. 유닛 테스트는 (일반적인 테스트와 달리) 개발자(developer) 뿐만 아니라 보다 더 심도 있는 테스트를 위해 테스터(tester)에 의해 수행되기도 한다.

1.3.2 Interface test

인터페이스는 API, 웹 서비스 등이 될 수 있다. 소프트웨어 또는 애플리케이션 또는 웹 사이트의 여러 구성 요소 간의 통신은 전체 성능에 영향을 미칠 수 있으므로 통신, 즉 인터페이스도 테스트하고 확인해야 한다. 인터페이스 기능을 확인하기 위해 수행되는 테스트를 인터페이스 테스트라고 한다.

1.3.3 API

API는 Application Programming Interface의 약어로 응용 프로그램에서 사용할 수 있도록, 운영 체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스를 뜻한다. 주로 파일 제어, 창

제어, 화상 처리, 문자 제어 등을 위한 인터페이스를 제공한다.

1.3.4 DB

DB는 데이터베이스의 약어로 여러 사람이 공유하여 사용할 목적으로 체계화해 통합, 관리하는 데이터의 집합이다. 작성된 목록으로써 여러 응용 시스템들의 통합된 정보들을 저장하여 운영할 수 있는 공용 데이터들의 묶음이다.

1.4 References

- Team1, Software Design Document, SKKU. (2020)
- 이은석 교수님, System Modeling, SKKU 소프트웨어공학개론 강의자료. (2021)

1.5 Overview

해당 테스트 계획 문서는 테스트 전략, 목적, 테스트, 테스트 예상에 대한 내용을 요약한 문서이다. 테스트 계획을 바탕으로 관리자는 소프트웨어 테스트를 정의된 프로세스에 따라 수행할 수 있다. 이는 관리자 뿐만 아니라 외부 이해관계자들에게도 테스트의 세부사항을 이해하는 데에 도움을 준다. 테스트 방법으로는 Unit test, interface test 두 가지를 선택해 테스트를 진행한다. 테스트 결과, 대상 애플리케이션이 정상적으로 작동하는지 확인한다. Unit test에서는 대상 애플리케이션의 각 기능별로 나눠 부분적인 작동을 확인한다. Interface test에서는 DB 또는 API와 같은 외부 인터페이스와 정상적으로 연결되고 작동되는지 테스트한다.

2. Approach

이 장에서는 시스템에 대한 테스트를 구현하기 위해 사용되는 tool과 의사코드를 개략적으로 설명한다.

2.1 Test method

2.1.1 Software unit test methods

Software unit test는 각 unit에 대하여 소프트웨어의 결함여부를 확인하고 요구사항에 맞게 기능하는지 검증한다. 이 장에서는 unit의 테스트를 위해 사용되는 의사코드를 제공한다. 각 테스트케이스에 대한 세부적인 내용은 3장에서 확인할 수 있다.

2.1.1.1. Register

```

Function register (id, pw, name, email)
if (id가 영어와 숫자의 조합인 경우 and id의 길이>=6 and id의 길이<=20 and id가 중복되지 않는 경우)
    print("사용가능한 ID입니다.")
else
    print("사용할 수 없는 ID입니다.")

if (pw가 영어와 숫자의 조합인 경우 and pw의 길이>=8 and pw의 길이<=20)
    print("사용가능한 password입니다.")
else
    print("사용할 수 없는 password입니다.")

if (email이 올바른 형식인 경우)
    print("이메일을 전송하였습니다.")
else
    print("이메일 형식이 올바르지 않습니다.")

```

2.1.1.2. Sign in

```

Function signIn (ID, Password)
if (ID와 password가 등록된 정보와 일치하는 경우)
    로그인 완료
else
    오류 메시지 출력 "가입하지 않은 아이디이거나, 잘못된 비밀번호입니다."

```

2.1.1.3. Profile modification

- 전공 변경

```

Function modification (최근 수정일)
if (최근 수정일이 한 학기 이내인 경우)
    오류 메시지 출력 "전공 변경은 한 학기에 한 번만 가능합니다."
else
    변경 완료

```

2.1.1.4. Chat list

```
Function chatList (사용자 정보)
if (로그인이 되어있지 않은 경우)
    오류 메시지 출력 “로그인 후 이용할 수 있습니다.”
else if (사용자의 수업 정보가 없는 경우)
    오류 메시지 출력 “채팅 가능한 상대가 없습니다.”
else if (네트워크에 연결되어 있지 않은 경우)
    오류 메시지 출력 “네트워크 연결이 필요합니다.”
else
    chat list를 불러옴
```

2.1.1.5. Send message

```
Function send (message)
if (네트워크에 연결되어 있지 않은 경우)
    오류 메시지 출력 “네트워크 연결이 필요합니다.”
else
    message 전송
```

2.1.1.6. Enter study room

```
Function enter (사용자의 전공)
if (로그인 되어있지 않은 경우)
    오류 메시지 출력 “로그인 후 이용할 수 있습니다.”
else if (스터디룸 인원 = 정원)
    오류 메시지 출력 “정원이 초과되었습니다.”
else if (스터디룸의 전공 != 사용자의 전공)
    오류 메시지 출력 “전공이 다릅니다.”
else if (네트워크에 연결되어 있지 않은 경우)
    오류 메시지 출력 “네트워크 연결이 필요합니다.”
else if (사용가능한 카메라 장치가 존재하지 않는 경우)
    오류 메시지 출력 “카메라 장치 허용이 필요합니다.”
else
    스터디룸 입장
```

2.1.1.7. Make study room

```

Function makeRoom (room_name)
if (로그인 되어있지 않은 경우)
    오류 메시지 출력 “로그인 후 이용할 수 있습니다.”
else if (room_name에 특수문자가 포함됨 or room_name 길이>12)
    오류 메시지 출력 “올바른 방 이름이 아닙니다.”
else if (네트워크에 연결되어 있지 않은 경우)
    오류 메시지 출력 “네트워크 연결이 필요합니다.”
else if (사용가능한 카메라 장치가 존재하지 않는 경우)
    오류 메시지 출력 “카메라 장치 허용이 필요합니다.”
else
    스터디룸 입장
  
```

2.1.2 Software interface test methods

프로젝트에서 사용되는 소프트웨어 인터페이스가 요구사항에 맞게 기능하는지 검증한다. 인터페이스를 특성에 따라 Parameter interfaces, Procedural interfaces, Message passing interfaces, Shared memory interfaces로 나누어 발생할 수 있는 오류를 테스트한다. Software interface 테스트에 대한 세부적인 내용은 4장에서 확인할 수 있다.

2.2 Test tool

2.2.1 Android Studio

Android 스튜디오는 Android 앱 개발을 위한 공식 통합 개발환경(IDE)이다. IntelliJ의 강력한 코드 편집기와 개발자 도구를 제공하고 결과를 확인할 수 있는 에뮬레이터를 제공한다.

2.2.2 Firebase

Firebase는 모바일, 웹 애플리케이션 개발 플랫폼으로 Backend 기능을 클라우드 서비스 형태로 제공한다. Firebase에서 제공하는 Firebase Test Lab은 클라우드 기반 앱 테스트 인프라로 한 번의 작업으로 다양한 기기에서 안드로이드 앱을 테스트하고 결과를 확인할 수 있다.

2.2.3 Ostorlab

모바일 앱을 자동으로 진단해주는 서비스이다. 앱에서 발생할 수 있는 잠재적인 취약점을 자동으로 찾아주며, 취약점 발생 지점을 보여준다. 애플리케이션의 security를 높이기 위해 사용한다.

3. Software Unit Test

이 장에서는 개발한 소프트웨어의 특정 메소드, 컴포넌트가 의도된 대로 정확히 작동하는지 검증하는 단위 테스트에 대해 서술한다. 어플리케이션의 전체 코드 중에서 아래에 나오는 몇 가지 기능 부분의 코드에 대한 테스트케이스를 작성한다.

3.1 Sign-up

3.1.1 Application sign-up test case

Test case object

서비스에 가입하는 함수(메서드)에 올바른 ID와 PW 형식을 입력했을 경우를 테스트한다.

Test Input

- 1) 영문과 숫자 조합으로 이루어진 ID와 PW를 입력창에 입력한다.
- 2) 중복되지 않은 ID와 PW 조합을 입력한다.

Expected Results

해당 ID와 PW를 사용할 수 있다는 메시지를 출력하고, 입력한 ID와 PW를 데이터 베이스에 저장하는 함수의 인자로 전달한다.

3.1.2. Application sign-up test case – Negative

Test case object

서비스에 가입하는 함수(메서드)에 올바르지 않은 ID와 PW 형식을 입력했을 경우를 테스트한다.

Test Input

- 1) 영문과 숫자 이외에 다른 문자가 포함된 ID와 PW를 입력한다.
- 2) 이미 데이터베이스에 존재하는 ID를 입력한다.

Expected Results

해당 ID와 PW를 사용할 수 없다는 메시지를 출력한다.

3.2. Sign-in

3.2.1. Application sign-in test case

Test case object

서비스 이용을 위해 가입 시 입력한 ID와 PW를 입력하여 로그인 하는 경우를 테스트한다.

Test Input

데이터베이스에 존재하는 ID와 PW 조합을 입력한다.

Expected Results

입력한 ID에 해당하는 사용자의 정보를 출력하는 메인 화면으로 이동한다.

3.2.2. Application sign-in test case - Negative

Test case object

로그인 화면에서 잘못된 ID와 PW를 입력하는 경우를 테스트한다.

Test Input

- 1) 데이터베이스에 존재하지 않는 ID를 입력하여 로그인을 시도한다.
- 2) 데이터베이스에 존재하는 ID에 대응하지 않는 PW를 입력한다.

Expected Results

- 1) 해당 ID는 존재하지 않는 ID라는 메시지를 출력한다.
- 2) PW가 일치하지 않는다는 메시지를 출력한다.

3.3. Check information in the database

3.3.1. Correct user data and database

Test case object

로그인한 사용자의 정보와 일치하는 데이터베이스의 정보를 화면에 올바르게 출력한다.

Test Input

데이터베이스에서 정보를 불러오는 함수에 로그인한 사용자의 정보를 인자로 전달한다.

Expected Results

올바른 사용자의 정보(전공, 이름, 학번, 이미지)를 화면에 출력한다.

3.3.2. Incorrect user data and database

Test case object

로그인한 사용자의 정보와 일치하는 데이터베이스의 정보가 올바르지 않은 정보일 경우를 테스트한다.

Test Input

데이터베이스에 정보를 불러오는 함수에 로그인한 사용자의 정보에서 오류가 포함된 내용을 전달한다.

Expected Results

시스템 관리자에게 데이터베이스에 잘못된 접근이라는 알림을 보낸다. 사용자에게는 잘못된 접근이라는 메시지를 출력하고 해당 사용자를 로그아웃 시킨다.

3.4. Participate study room

3.4.1. Participate study room test case

Test case object

스터디 목록에서 원하는 스터디를 선택할 때 입장할 수 있는 스터디 룸을 선택하는 경우를 테스트한다.

Test Input

입장 가능 인원이 남은 스터디 룸을 클릭한다.

Expected Results

해당 스터디 룸에 사용자가 참가할 수 있도록 하며, 해당 스터디 룸의 참여자 인원을 1 증가시킨다. 해당 스터디 룸에 참여 중인 사용자들의 정보를 불러온다.

3.4.2. Participate study room test case - Negative

Test case object

스터디 목록에서 원하는 스터디를 선택할 때 입장할 수 없는 스터디 룸을 선택하는 경우를 테스트한다.

Test Input

입장 가능 인원이 남지 않은 스터디 룸을 클릭한다.

Expected Results

해당 스터디 룸은 참여할 수 없다는 메시지를 출력한다.

3.5. Create New study room

3.5.1. Create study room

Test case object

스터디 룸을 생성할 수 있는 사용자가 스터디 생성 페이지에서 Create 버튼을 클릭하는 경우를 테스트한다.

Test Input

- 1) 현재 생성한 스터디 룸이 없는 사용자이다.
- 2) 현재 소속된 전공에 대한 스터디를 올바르게 생성한다.

Expected Results

새로운 스터디에 대한 정보를 시스템 데이터베이스에 등록하고, 전체 스터디 목록 리스트에 해당 스터디를 다른 사용자에게 보일 수 있도록 리스트를 초기화한다.

3.5.2. Create study room - Negative

Test case object

스터디 룸을 생성할 수 없는 사용자가 스터디 생성 페이지에서 Create 버튼을 클릭하는 경우를 테스트한다.

Test Input

- 1) 현재 생성한 스터디 룸이 있는 사용자이다.
- 2) 현재 소속된 전공과 일치하지 않는 스터디 룸을 생성한다.

Expected Results

- 1) 이미 생성한 스터디 룸이 있다는 메시지를 출력한다.
- 2) 시스템 관리자에게 데이터베이스에 잘못된 접근이라는 알림을 보내고, 사용자에게는 잘못된 접근이라는 메시지를 출력한다.

3.6. Chat service

3.6.1. Send correct message in chat service

Test case object

오픈 채팅 서비스에서 올바른 메시지를 입력하는 경우를 테스트한다.

Test Input

사용자가 채팅방에서 메시지를 입력하고 전송 버튼을 클릭한다.

Expected Results

사용자가 입력창에 입력한 메시지 내용을 시스템 데이터베이스에 저장하고, 해당 채팅방의 모든 사용자에게 메시지를 전송한다.

3.6.2. Send incorrect message in chat service

Test case object

오픈 채팅 서비스에서 올바르지 않은 형식의 메시지를 입력하는 경우를 테스트한다.

Test Input

사용자가 채팅방에서 메시지를 입력하지 않은 상태에서 전송 버튼을 클릭한다.

Expected Results

내용을 입력하라는 메시지를 출력한다.

4. Software Interface test

소프트웨어 인터페이스 테스트는 시스템의 허점과, 인터페이스에 대한 올바르지 않은 이해가 초래하는 버그들을 찾는 테스트이다. 크게는 네 가지 인터페이스의 종류가 있으며 다음과 같다.

- 파라미터 인터페이스: 한 컴포넌트에서 다른 컴포넌트로 데이터를 옮기는 인터페이스.
- 프로시저 인터페이스: 한 서브시스템에 속한 프로시저들을 다른 서브시스템에서 실행하는 인터페이스.
- 메시지 전달 인터페이스: 서브시스템에서 다른 서브시스템들로 서비스 요청을 보내는 인터페이스.
- 공유 메모리 인터페이스: 공유 메모리를 사용하는 인터페이스.

인터페이스를 테스트하기 위해서 기본적으로 5가지 가이드라인이 있으며, 다음과 같다:

- 가능한 값들 중 가장 큰 값과 가장 작은 값으로 프로시저의 파라미터를 테스트한다.
- 널 포인터를 파라미터로 입력해서 예외처리 여부를 확인한다.
- 컴포넌트에 오류가 나도록 테스트를 작성해본다.
- 메시지 전달 시스템에서 메시지의 양을 극한으로 올려서 잘 견디는지 확인한다.
- 공유 메모리를 사용하는 컴포넌트들의 프로시저의 순서를 바꿔가며 실행시켜본다.

4.1. Database Interface

데이터베이스 시스템에서는 firebase에서 데이터의 인풋과 아웃풋을 담당하는 API 인터페이스를 사용하게 된다.

4.1.1 Connection test

[Table 1] Table of Connection test

Test Case Object	Firestore의 데이터베이스 API 인터페이스 연결 테스트.
Test Input	Ping command
Expected Results	Ping status

4.1.2. Null Pointer Testing

[Table 2] Table of Null Pointer Testing

Test Case Object	Firestore의 데이터베이스 API 인터페이스 널 포인터 테스트.
Test Input	Null값을 read하라는 명령을 보낸다.
Expected Results	NullPointerException에 알맞은 Response를 반환한다.

4.1.3. Component Failure Testing

[Table 3] Table of Component Failure Testing

Test Case Object	Firestore의 데이터베이스 API 인터페이스 오류 테스트.
Test Input	존재하지 않는 데이터를 읽으라는 명령을 보낸다.
Expected Results	존재하지 않는 데이터이기 때문에, 그에 알맞은 Response를 반환한다.

4.1.4. Stress Testing

[Table 4] Table of Stress Testing

Test Case Object	Firestore의 데이터베이스 API 인터페이스 성능 테스트.
Test Input	300개의 Read() 요청을 한번에 보낸다.
Expected Results	한번에 처리가 되지 않는 것들은 큐로 들어가서 천천히 다 처리한다.

4.1.5. Shared Memory Testing

[Table 5] Table of Shared Memory Testing

Test Case Object	Firestore의 데이터베이스 API 인터페이스 공유 메모리 테스트.
Test Input	같은 데이터에 대해서 read와 write를 바꿔가며 입력한다.
Expected Results	Lock이 정상적으로 작동해서, race condition이 생기지 않는다. 그러므로 데이터가 순서에 맞지 않게 업데이트 되는 일이 없다.

4.2. Jitsi Video API Interface

Jitsi는 오픈소스 화상회의 라이브러리로, 화상회의 서버를 만들 수 있다. 이 서버는 API를 사용해서 접근해서 웹 어플리케이션의 형태로 사용할 수 있게 된다.

4.2.1. Connection Test

[Table 6] Table of Connection Test

Test Case Object	Jitsi Meet Server API 인터페이스 연결 테스트.
Test Input	Ping command
Expected Results	Ping status

4.2.2. Null Pointer Testing

[Table 7] Table of Null Pointer Testing

Test Case Object	Jitsi Meet Server API 인터페이스 널 포인터 테스트.
Test Input	온라인 독서실에 참가할 때, 방 정보로 Null값을 보낸다.
Expected Results	NullPointerException과 함께 방 정보를 다시 입력하라는 로그로 남긴다.

4.2.3. Component Failure Testing

[Table 8] Table of Component Failure Testing

Test Case Object	Jitsi Meet Server API 인터페이스 오류 테스트.
Test Input	온라인 독서실에 참가할 때, 방 정보로 존재하지 않는 방 정보를 보낸다.
Expected Results	존재하지 않는 방 정보이기 때문에, Exception과 함께 다시 올바른 방을 입력하라는 메시지를 로그로 남긴다.

4.2.4. Stress Testing

[Table 9] Table of Stress Testing

Test Case Object	Jitsi Meet Server API 인터페이스 성능 테스트.
Test Input	여러 아이디로 같은 방에 대하여 300개의 참여 요청을 한 번에 보낸다.
Expected Results	같은 아이디로 여러 번 참여할 경우 처음 외에 다른 요청들은 무시되기 때문에, 아이디의 개수만큼만 참여 요청이 처리된다.

4.3. Chat System Interface

채팅 방 내에서 유저들이 주고받는 메시지를 처리하는 시스템으로, Firebase Cloud Messaging을 통해서 상대방에게 직접 메시지를 전달하며, DB에도 저장한다.

4.3.1. Connection Test

[Table 10] Table of Connection Test

Test Case Object	Firebase Cloud Messaging API 인터페이스 연결 테스트
Test Input	Ping command
Expected Results	Ping status

4.3.2. Null Pointer Testing

[Table 11] Table of Null Pointer Testing

Test Case Object	Firebase Cloud Messaging API 인터페이스 연결 테스트
Test Input	메시지를 보낼 때, 널 값을 보낸다.
Expected Results	아무런 메시지가 전달되지 않으며, 시스템에서 NullPointerException을 로그로 남긴다.

4.3.3. Component Failure Testing

[Table 12] Table of Component Failure Testing

Test Case Object	Firebase Cloud Messaging API 인터페이스 오류 테스트
Test Input	존재하지 않는 유저에게 메시지를 보낸다.
Expected Results	아무런 메시지가 전달되지 않으며, 시스템에서 InvalidUserException을 로그로 남긴다.

4.3.4. Stress Testing

[Table 13] Table of Stress Testing

Test Case Object	Firebase Cloud Messaging API 인터페이스 성능 테스트
Test Input	여러 유저가 한 유저에게 200개에 메시지를 동시에 보낸다.
Expected Results	메시지를 정상적으로 받게 된다. Firebase는 scalable하기 때문이다.

4.3.5. Shared Memory Testing

[Table 14] Table of Shared Memory Testing

Test Case Object	Firestore Cloud Messaging API 인터페이스 공유 메모리 테스트
Test Input	두 유저가 상대방에게 순서를 바꿔가며 상대방에게 메시지를 남긴다.
Expected Results	Firestore Cloud Messaging 시스템 자체적으로락을 관리해서 race condition이 발생하지 않을 것이다.

5. Supporting Information

5.1 Document History

[Table 15] Table of Document History

Date	Version	Description	Writer
21/05/28	0.1	Section 1, 5 추가	김지수
21/05/28	0.2	Section 2 추가	박윤진
21/05/28	0.3	Section 3 추가	박세연
21/05/28	0.4	Section 4 추가	장영재
21/05/28	1.0	전체 문서 취합 및 정리	김지수