

# 멘토-멘티 매칭 프로그램

## Software Test Plan Document



2021.05.30

**소프트웨어공학 TEAM2**

Team Leader 2015310783 서기용

Team Member 2015318579 김정훈

Team Member 2017310474 김시인

Team Member 2018312827 김민지

1.Introduction .....	5
1.1 Purpose .....	5
1.2 Scope .....	5
1.3 Definition, Acronyms, and Abbreviation .....	5
1.4 References .....	5
1.5 Overview .....	6
2. Approach .....	7
2.1 Test Method .....	7
2.1.1 Software unit test methods .....	7
2.1.1.1 프로필 .....	7
2.1.1.2 멘토 검색 .....	8
2.1.1.3 멘토 검색결과 .....	9
2.1.1.4 소모임 검색 .....	11
2.1.1.5 소모임 검색결과 .....	12
2.1.1.6 매칭정보 .....	14
2.1.1.7 채팅 .....	16
2.1.1.8 리뷰 .....	19
2.1.2 Software interface test methods .....	20
2.1.2.1 Firebase .....	20
2.1.2.2 Database .....	22
2.1.2.3 Application Programming Interface(API) .....	25
2.1.2.4 Apache .....	27
3. Software Unit Test .....	28
3.1 프로필 .....	28
3.1.1 SetProfile() test case – Positive .....	28
3.1.2 SetProfile() test case – Negative .....	28

3.1.3 ShowProfile() test case – Positive .....	29
3.1.4 ShowProfile() test case – Negative .....	29
3.1.5 ModifyProfile() test case – Positive .....	29
3.1.6 ModifyProfile() test case – Negative .....	29
3.2 멘토검색 .....	30
3.2.1 SearchMentor test() case – Positive .....	30
3.2.2 SearchMentor test() case – Negative .....	30
3.2.3 RecommendMentor() test case – Positive .....	30
3.2.4 RecommendMentor() test case – Negative .....	30
3.3 소모임검색 .....	31
3.3.1 Search 소모임() test case – Positive .....	31
3.3.2 Search 소모임() test case – Negative .....	31
3.3.3 Recommend 소모임() test case – Positive .....	31
3.3.4 Recommend 소모임() test case – Negative .....	31
3.4 멘토 검색결과 .....	31
3.4.1 RecommendMentor() test case – Positive .....	31
3.4.2 RecommendMentor() test case – Negative .....	32
3.4.3 ShowMentor() test case – Positive .....	32
3.4.4 ShowMentor() test case – Negative .....	32
3.4.5 ShowDetail() test case – Positive .....	32
3.4.6 ShowDetail() test case – Negative .....	32
3.5 소모임 검색결과 .....	33
3.5.1 소모임 추천 test case .....	33
3.5.2 소모임 리스트 출력 test case – Positive .....	33
3.5.3 소모임 리스트 출력 test case – Negative .....	33
3.5.4 소모임 세부정보 출력 test case .....	33

3.6 매칭 정보 .....	34
3.6.1 매칭 요청 test case – Positive .....	34
3.6.1 매칭 요청 test case – Negative .....	34
3.6.3 요청된 매칭 신청 수락 test case .....	34
3.6.4 요청된 매칭 신청 거절 test case .....	34
3.7 채팅 .....	35
3.7.1 채팅로그 불러오기 test case – Positive .....	35
3.7.2 채팅로그 불러오기 test case – Negative .....	35
3.7.3 메시지 전송 test case – Positive .....	35
3.7.4 메시지 전송 test case – Negative .....	35
3.7.5 메시지 답변 전송 test case – Positive .....	35
3.7.6 메시지 답변 전송 test case – Negative .....	36
3.8 리뷰 .....	36
3.8.1 리뷰 작성 test case – Positive .....	36
3.8.2 리뷰 작성 test case – Negative .....	36
3.8.3 프로필 리뷰 출력 test case – Positive .....	36
3.8.4 프로필 리뷰 출력 test case – Negative .....	36
3.8.5 별 점수 업데이트 test case – Positive .....	37
3.8.6 별 점수 업데이트 test case – Negative .....	37
4. Software Interface Test .....	37
4.1 Firebase .....	37
4.1.1 Connection Test .....	37
4.1.2 Android App Testing .....	37
4.1.3 Physical Devices Testing .....	38
4.1.4 Workflow Integration Testing .....	38
4.2 Database .....	38

4.2.1 Schema Test .....	38
4.2.2 Database Tables and Columns Test .....	39
4.2.3 Keys and Indexes Test .....	39
4.2.4 Database Server Validation Test .....	39
4.2.5 Trigger Test .....	39
4.2.6 Functional Database Test .....	39
4.2.7 Store Procedure Test .....	40
4.3 Application Programming Interface .....	40
4.3.1 Verifying Status Code .....	40
4.3.2 Verifying Payload .....	40
4.3.3 Verifying Response Header .....	40
4.3.4 Verifying Application State .....	41
4.4 Apache .....	41
4.4.1 Valid Response Test .....	41
4.4.2 Website Response Test .....	41
4.4.3 Connection Test .....	41
5. Supporting Information .....	42
5.1 Document History .....	42

# 1. Introduction

## 1.1 Purpose

이 테스트 계획 문서는 ‘멘토 멘티 추천 프로그램’의 Requirement Specification 에 기술된 Product Function 과 Software Interface 를 테스트하기 위한 방법과 테스트 케이스를 설명하기 위해 작성되었습니다. 테스트 방법은 테스트 툴과 각 테스트에 대한 pseudo code 를 포함하고, 각각의 테스트 케이스는 Test input 과 expected result 로 구성되어 있습니다.

## 1.2 Scope

이 문서는 크게 테스트를 진행하기 위한 test tool 과 각각의 test 에 대한 psuedo code 등의 test method 를 설명하는 Approach, 저희가 개발하려는 software 를 구성하는 product function 들에 대한 unit test, 그리고 외부 software 와의 interface 를 test 하는 software Interface test 에 대한 설명으로 구성되어 있습니다.

## 1.3 Definition, Acronyms, and Abbreviations

### 1.3.1 API

API 는 응용프로그램에서 사용할 수 있도록 운영체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있도록 만든 인터페이스 입니다. API 를 사용하면 구현 방식을 모르더라도 프로그램 간의 커뮤니케이션이 가능합니다.

### 1.3.2 Apache

아파치는 World Wide Web 서버용 소프트웨어 입니다. 추가 기능들이 다양하고 구축이 쉽게 때문에 제일 많이 사용되고 있습니다.

## 1.4 References

### 1.4.1 Software testing tool

<https://dzone.com/articles/top-10-automated-software-testing-tools>

### 1.4.2 DB

<https://www.guru99.com/data-testing.html#:~:text=Database%20Testing%20is%20a%20type,Database%20and%20check%20its%20responsiveness.>

### 1.4.3 API

<https://www.sisense.com/blog/rest-api-testing-strategy-what-exactly-should-you-test/>

<https://ko.wikipedia.org/wiki/API>

### 1.4.4 Apache

<https://www.dotcom-tools.com/web-servers-test>

[https://ko.wikipedia.org/wiki/%EC%95%84%ED%8C%8C%EC%B9%98\\_HTTP\\_%EC%84%9C%EB%B2%84](https://ko.wikipedia.org/wiki/%EC%95%84%ED%8C%8C%EC%B9%98_HTTP_%EC%84%9C%EB%B2%84)

## 1.5 Overview

이 문서에서는 5 개의 chapter 로 구성되어 있습니다. Chapter2 에서는 저희가 진행하는 프로젝트에 사용될 테스트 방법을 기술하고 있습니다. 저희가 진행할 테스트는 크게 Software unit test 와 Software interface test 로 구성이 됩니다. Software unit test method 에서는 저희가 개발하는 프로그램의 function 및 method 들을 테스트 하기 위한 Pseudo Code 를 보여줍니다. Software interface test method 에서는 저희 프로그램이 다른 시스템과 상호작용하기 위한 Interface 를 테스트하는 Pseudo Code 를 소개합니다.

Chapter3 에서는 Software unit test 에 사용될 test case 들을 기술하고 있습니다. 시스템을 구성하는 모든 function 과 method 를 테스트 할 수 있습니다. Test case 는 test 대상을 설명하는 test case object, test case 의 input, 그리고 input 에 대한 예상 결과로 구성됩니다. 하나의 method 는 실행에 성공했을 경우와 실패했을 경우를 가정한 Positive case 와 Negative case 를 쌍으로 가집니다.

Chapter4 에서는 다른 시스템들과의 상호작용을 위한 Interface 에서 발생하는 상황에 대한 test case 들로 구성됩니다. 저희 프로그램이 상호작용하는 외부 시스템은 Firebase, Database, API, 그리고 Apache 가 있습니다.

마지막으로 Chapter5 에서는 이 문서의 히스토리를 제공합니다.

## 2. Approach

### 2.1 Test Method

#### 2.1.1 Software unit test methods

##### 2.1.1.1 프로필

###### (1) SetProfile() Method

특정 ID 로 로그인 한 사용자에게 한하여 프로필 영역을 작성하는 method

```
Test Case set T
Success Case S
Failure Case F
for each case C in T do
    repeat
        SetProfile(C)
        if get success message from server then
            S U C
        else
            F U C
    until reach maximum iteration
end for
```

Fig1. Pseudo Code for SetProfile() Method

###### (2) ShowProfile() Method

```
Test Case set T
Success Case S
Failure Case F1
Failure Case F2
for each case C in T do
    repeat
        profile = ShowProfile(C)
        if profile != None then
            S U C
        elif profile == None then
            F1 U C
        else get any error message
            F2 U C
    until reach maximum iteration
end for
```

Fig2. Pseudo Code for ShowProfile() Method



## (3) ModifyProfile() Method

```

Test Case set T
Success Case S
Error Case E
Failure Case F
for each case C in T do
    Modified = ModifyProfile(C)
    if get any error message then
        E U C
    for values in Modified do
        if values != C[keywords] then
            F U C
            go to next iteration
        S U C
    until reach maximum iteration
end for

```

Fig3. Pseudo Code for ModifyProfile() Method

## 2.1.1.2 멘토 검색

## (1) SearchMentor() Method

```

Test Case set T
Success Case S
Error Case E

for each case C in T do
    keyword = C[keyword]
    result = SearchMentor(keyword)
    if get any error message then
        E U C
    elif result is success message then
        S U C
    until reach maximum iteration
end for

```

Fig4. Pseudo Code for SearchMentor() Method

## (2) Recommend\_Mentor() Method

```
Test Case set T
SendPreferenceinfo error case E1
Recommend error case E2
Success case S

for each case C in T do
    preference = SendPreferenceinfo()
    if get any error message then
        E1 U C

    result = Recommend_mentor(preference)

    if get any error message then
        E2 U C

    elif result is success message then
        S U C
    until reach maximum iteration
end for
```

Fig5. Pseudo Code for Recommend() Method

**2.1.1.3 멘토 검색 결과**

## (1) ShowMentor() Method

```
Test Case set T
Error Case set E
Failure Case set F

for each case C in T do
    List = ShowMentor()
    if get any error message then
        E U C

    elif List is empty then
        F U C

    else
        S U C
    until reach maximum iteration
end for
```

Fig6. Pseudo Code for ShowMentor() Method

## (2) ShowDetail\_mentor() Method

```

Test Case set T
ShowDetail error case SE
Failure Case F
Request error case RE
Success case S

for each case C in T do
    if request_detail then
        mentor_profile = ShowDetail_mentor()

        if get any error message then
            SE U C
            go to next iteration
        elif mentor_profile is None then
            F U C
            go to next iteration
        else S U C
            go to next iteration
    else
        RE U C

```

Fig7. Pseudo Code for ShowDetail() Method

## (3) Recommend\_mentor2() Method

```

Test Case set T
Request Result error case RE
Send Result error case SE
Recommend error case RCE
Success case S

for each case C in T do
    if request_result then
        recommend = Recommend_mentor2()
        if get any error message then
            RCE U C
            go to next iteration
    else then
        RE U C
        go to next iteration

    result = send_result(recommend)
    if get any error message then
        SE U C
        go to next iteration

    S U C
    until reach maximum iteration
end for

```

Fig7. Pseudo Code for ShowDetail() Method

### 2.1.1.4 소모임 검색

#### (1) SearchCommunity() Method

```

Test Case Set T
Error Case E
Success Case S

for each case C in T do
    keyword = C[keyword]
    result = SearchCommunity(keyword)

    if get any error message then
        E U C

    elif result is ok_message then
        S U C

    until reach maximum iteration
end for

```

Fig9. Pseudo Code for SearchCommunity() Method

#### (2) Recommend\_Community() Method

```

Test Case set T
SendPreferenceinfo error case E1
Recommend error case E2
Success case S

for each case C in T do
    preference = SendPreferenceinfo()
    if get any error message then
        E1 U C

    result = Recommend_community(preference)

    if get any error message then
        E2 U C

    elif result is success message then
        S U C

    until reach maximum iteration
end for

```

Fig10. Pseudo Code for Recommend\_community() Method

### 2.1.1.5 소모임 검색 결과

#### (1) ShowCommunity() Method

```

Test Case set T
Error Case set E
Failure Case set F

for each case C in T do
    List = ShowCommunity()
    if get any error message then
        E U C

    elif List is empty then
        F U C

    else
        S U C
    until reach maximum iteration
end for

```

Fig11. Pseudo Code for ShowCommunity() Method

#### (2) ShowDetail\_community() Method

```

Test Case set T
ShowDetail error case SE
Failure Case F
Request error case RE
Success case S

for each case C in T do
    if request_detail then
        community_profile = ShowDetail_community()

        if get any error message then
            SE U C

        elif community_profile is None then
            F U C

        else S U C
    else
        RE U C

```

Fig12. Pseudo Code for ShowDetail() Method

## (3) Recommend\_community2() Method

```
Test Case set T
Request Result error case RE
Send Result error case SE
Recommend error case RCE
Success case S

for each case C in T do
    if request_result then
        recommend = Recommend_community2()
        if get any error message then
            RCE U C
            go to next iteration
        else then
            RE U C
            go to next iteration

        result = send_result(recommend)
        if get any error message then
            SE U C
            go to next iteration

        S U C
    until reach maximum iteration
end for
```

Fig13. Pseudo Code for Recommend\_community2() Method

### 2.1.1.6 매칭정보

#### (1) RequestMatching() Method

```

Test Case set T
request_matching var error case RE
RequestMatching_function error case RME
Success case S

for each case C in T do
    if request_matching then
        result = RequestMatching()
        if get any error message then
            RME U C
            go to next iteration
        else then
            RE U C
            go to next iteration
    S U C
until reach maximum iteration
end for

```

Fig14. Pseudo Code for RequestMatching() Method

#### (2) Accept\_Reject() Method

```

Test Case set T
Error case E
Failure case F
Success case S

for each case C in T do
    result = Accept_Reject() : mentor user's decision
    if get any error message then
        E U C
        go to next iteration
    if result is None then
        F U C
        go to next iteration
    S U C
until reach maximum iteration
end for

```

Fig15. Pseudo Code for Accept\_Reject() Method

## (3) GetStatus() Method

```

Test Case set T
Getstatus error GE
Send_status error SE
Success case S

for each case C in T do
    click : click 'Status' by user

    command = GetStatus()
    if get any error message then
        GE U C
        go to next iteration

    status = SendStatus()
    : response to GetStatus from server
    if get any error message then
        SE U C
        go to next iteration
    S U C

until reach maximum iteration
end for

```

Fig16. Pseudo Code for GetStatus() Method

## (4) ShowStatus() Method

```

Test Case set T
Error case E
Failure case F
Success case S

for each case C in T do
    status : output of GetStatus()
    by server

    ShowStatus()
    if get any error message then
        E U C
        go to next iteration

    if nothing pops up then
        F U C
        go to next iteration

    S U C

until reach maximum iteration
end for

```

Fig17. Pseudo Code for ShowStatus() Method



### 2.1.1.7 채팅

#### (1) GetChattinglog() Method

```

Test Case set T
Error case E
Failure case F
Success case S

for each case in T do
    click : click chatroom by user
    get_log = GetChattinglog()
    if get any error message then
        E U C
        go to next iteration
    if get_log is None :
        F U C
        go to next iteration
    S U C
until reach maximum iteration
end for

```

Fig18. Pseudo Code for GetChattinglog() Method

#### (2) ShowChattinglog() Method

```

Test Case set T
Sendchattinglog error SE
Chattinglog send failure CLF
Showchattinglog error SCE
Success case S

for each Case in T do
    chattinglog : output of SendChattinglog()
    by server
    if get any error message then
        SE U C
        go to next iteration
    if chattinglog is None then
        CLF U C
        go to next iteration
    ShowChattinglog(chattinglog)
    if get any error message then
        SCE U C
        go to next iteration
    S U C
until reach maximum iteration
end for

```

Fig19. Pseudo Code for ShowChattinglog() Method

## (3) SendMessage() Method

```

Test case set T
Error case E
Failure case F
Success case S

for each case in T do
    message : message entered by user
    message = SendMessage(message)
    if get any error message then
        E U C
        go to next iteration
    if message is None then
        F U C
        go to next iteration
    S U C
    until reach maximum iteration
end for

```

Fig20. Pseudo Code for SendMessage() Method

## (4) ShowMessage() Method

```

Test case set T
Error case set E
Failure case set F
Success case set S

for each case C in T do
    message : output of SendMessage()
    ShowMessage(message)
    if get any error message then
        E U C
        go to next iteration
    elif message is None then
        F U C
        go to next iteration
    else S U C
    until reach maximum iteration
end for

```

Fig21. Pseudo Code for ShowMessage() Method

## (5) SendReply() Method

```
Test case set T
Error case set E
Failure case set F
Success case set S

for each case C in T do
    message : Entered Reply from mentor user
    reply = SendReply(message)
    if get any error message then
        E U C
        go to next iteration
    elif reply is None then
        F U C
    else then S U C
    until reach maximum iteration
end for
```

Fig22. Pseudo Code for SendReply() Method

## (6) ShowReply() Method

```
Test case set T
Error case set E
Failure case set F
Success case set S

for each case C in T do
    reply : Output of the SendReply()
    ShowReply()
    if get any error message then
        E U C
    elif if nothing pops up then
        F U C
    else then S U C
```

Fig23. Pseudo Code for ShowReply() Method

### 2.1.1.8 리뷰

#### (1) WriteReview() Method

```
Test case set T
Error case set E
Failure case set F
Success case set S

for each case C in T do
    review = WriteReview()
    if get any error message then
        E U C
    elif review is empty then
        F U C
    else S U C
    until reach maximum iteration
end for
```

Fig24. Pseudo Code for WriteReview() Method

#### (2) UploadReview() Method

```
Test case set T
Error case set E
Failure case set F
Success case set S

for each case C in T do
    review : output of WriteReview()
    score = UpdateReview(review)
    if get any error message then
        E U C
    elif score is None then
        F U C
    else S U C
    until reach maximum iteration
end for
```

Fig25. Pseudo Code for UploadReview() Method

## (3) UpdateScore() Method

```

Test case set T
Error case set E
Failure case set F
Success case set S

for each case C in T do
    score : output of UpdateReview()
    UpdateScore(score)
    if get any error message then
        E U C
    elif nothing changed from last review score then
        F U C
    else S U C
    until reach maximum iteration
end for

```

Fig26. Pseudo Code for UpdateScore() Method

**2.1.2 Software interface test methods****2.1.2.1 Firebase**

## (1) Connection Testing

```

Test case set T
Error case set E

for each case C in T do
    Input Ping Command to device
    Get Ping status
    if not Ping status then
        E U C
        go to next iteration
    Check Ping status
    until reach maximum iteration
end for

```

Fig27. Pseudo Code for Connection Testing

## (2) Android App Testing

```

Test case set T
Compatibility error case set E1
Execution error case set E2

for each case C in T do
    d : Android based hosting device
    check compatibility
    if get any error message then
        E1 U C
        go to next iteration

    check executability
    if get any error emssage then
        E2 U C
        go to next iteration

    else
        check compatibility status
        check execution status
        S U C
    until reach maximum iteration
end for

```

Fig28. Pseudo Code for Android App Testing

### (3) Physical Device Testing

```

Test case set T
Error case set E
Success case set S

for each case C in T do
    version command VI
    language command LI
    device command DI

    check if device work under
    each command VI, LI, DI

    if get any error message then
        E U C
        go to next iteration

    else
        check version status
        check language status
        check device status
        S U C
    until reach maximum iteration
end for

```

Fig28. Pseudo Code for Physical Device Testing

## (4) Workflow Integration Testing

```

Test case set T
Compatibility error case set CE
Integration error case set
Success case set S

for each case C in T do
    input compatibiliy command
    if get any error message then
        CE U C
        go to next iteration

    input integration command
    if get any error message then
        IE U C
        go to next iteration

    check compatibility status and
    integration status
    S U C
    until reach maximum iteration
end for

```

Fig29. Pseudo Code for Workflow Integration Testing

**2.1.2.2 Database**

## (1) Schema Test

```

Test case set T
Error case set E
Success case set S

for each case C in T do
    M1 : Database mapping format
    M2 : UI mapping format
    if Compatible(M1, M2) then
        S U C
    else
        E U C

    until reach maximum iteration
end for

```

Fig30. Pseudo Code for Schema Test

## (2) Database Tables and Columns Test

```

Test case set T
Naming convention error case set NE
Datatype error case set DE

for each case C in T do
    M1 = Backend database field mapping
    M2 = Frontend field mapping

    for each number in range Field lengths
        if naming convention of M1[number] == M2[number] then
            if datatype of M1[number] == M2[number] then
                S U C
            else DE U C
        else
            NE U C

    until reach maximum iteration
end for

```

Fig31. Pseudo Code for Database Tables and Columns Test

## (3) Keys and Indexes Test

```

Test case set T
Validation error set V
Datatype error set D
IndexingRule error set I
Size error set SE
Success set S

for each case C in T do
    if not valid(FK) then
        V U C
        go to next iteration

    if type(PK) != type(FK) then
        D U C
        go to next iteration

    if index not following rule then
        I U C
        go to next iteration

    if not checksize(index) then
        SE U C
        go to next iteration
    S U C
until reach maximum iteration
end for

```

Fig32. Pseudo Code for Keys and Indexes Test



## (4) Database Server Validation Test

```

Test case set T
Validation error set V
Authorization error A
Transaction error set T
Success set S

for each case C in T do
    if Database_server_command is not valid then
        V U C
        go to next iteration

    if not Authorized_user_action_command then
        A U C
        go to next iteration

    if maximum_user < Transaction then
        T U C
        go to next iteration
    S U C
    until reach maximum iteration
end for

```

Fig33. Pseudo Code for Database Server Validation Test

## (5) Trigger Test

```

Test case set T
Update error set UE
Insert error set I
Delete error set D
Success case set S

for each case C in T do
    if update error then UE U C
        go to next iteration

    if insert error then I UC
        go to next iteration

    if delete error then D U C
        go to next iteration
    S U C
    until reach maximum iteration
end for

```

Fig34. Pseudo Code for Trigger Test

## (6) Store Procedure Test

```
Test case set T
Error case set E

for each case C in T do
    for each application in C do
        for each module in C do
            check storing procedure
            if error E U (application, module)
            else S U (application, module)
        until reach maximum iteration
    end for
```

Fig35. Pseudo Code for Store Procedure Test

**2.1.2.3 Application Programming Interface(API)**

## (1) Verifying Status Code

```
Test case set T
wrong HTTP code case set N
correct HTTP code case set P

for each case C in T do
    HTTP_status = Verify(HTTP_status code)
    if wrong then
        N U C
    else
        P U C
    until reach maximum iteration
end for
```

Fig36. Pseudo Code for Verifying Status Code

## (2) Verifying Payload

```
Test case set T
for each case in T do

    check JSON body
    check Field names
    check types
    check values

    until reach maximum iteration
end for
```

Fig37. Pseudo Code for Verifying Payload

## (3) Verifying Response Header

```
Test case set T
Security and performance error case set E1
Ping error case set E2
Success case set S

for each case C in T do

    Input Security and performance command
    if get error message then
        E1 U C
        go to next iteration

    else
        check Security and performance status

    Input Ping Command
    if get error message then
        E2 U C
        go to next iteration

    else
        check Ping status
        S U C
    until reach maximum iteration
end for
```

Fig38. Pseudo Code for Verifying Response Header

#### (4) Verifying Application State

```

Test case set T
UI error case set E1
Interface case set E2
Success case set S

for each case C in T do
    Input UI command
    if get error message then
        E1 U C
        go to next iteration
    else
        check UI status

    Input Interface Command
    if get error message then
        E2 U C
        go to next iteration
    else
        check Interface status
    S U C
until reach maximum iteration
end for

```

Fig39. Pseudo Code for Verifying Application State

#### 2.1.2.4 Apache

##### (1) Valid Response Test

```

Test case set T
Response error case set E
Invalid response case set I
Success case set S

for each case C in T do
    Wait for response from webserver
    if no response then
        E U C
    else
        if Webserver Return Status is valid
            S U C
            go to next iteration
        else
            I U C
    until reach maximum iteration
end for

```

Fig40. Pseudo Code for Verifying Response Test

## (2) Website Response Test

```

Test case set T
for each case C in T do
    get reference_time
    input Respond Command
    check (real_time - reference_time)
    until reach maximum iteration
end for

```

Fig41. Pseudo Code for Website Response Test

## (3) Connection Test

```

Test case set T
for each case C in T do
    Input Ping command
    Check Ping status
    until reach maximum iteration
end for

```

Fig42. Pseudo Code for Connection Test

# 3. Software Unit Test

## 3.1 프로필

### 3.1.1 SetProfile() test case – Positive

Test case object: 사용자, 멘티 사용자 프로필, 멘토 사용자 프로필, 서버 기반으로 프로필이 정상적으로 설정되는 경우 평가

Test Inputs: ID Email (@skku.edu 형식의 성균관대학교 웹메일 주소), 비밀번호, 닉네임, 멘토/멘티 역할, 매칭 희망 분야, 소모임 관심 분야

Expected Results: 프로필 설정 Success message

### 3.1.2 SetProfile() test case – Negative

Test case object: 사용자, 멘티 사용자 프로필, 멘토 사용자 프로필, 서버 기반으로 프로필이 정상적으로 설정되지 않는 경우 평가

Test Inputs: @skku.edu 가 아닌 타 형식 (ex. @naver.com, @gmail.com) 이메일 주소, ID Email 이외비밀번호, 닉네임, 멘토/멘티 역할, 매칭 희망 분야, 소모임 관심 분야는 positive test case 와 동일

Expected Results: 프로필 설정 Failed message

### 3.1.3 ShowProfile() test case – Positive

Test case object: 사용자, 멘티 사용자 프로필, 멘토 사용자 프로필, 네트워크, 서버 기반으로 사용자의 프로필이 정상적으로 확인되는지 평가

Test Inputs: 프로필 확인 요청 입력, 네트워크 연결 안정 상태

Expected Results: 프로필 확인 Success message

### 3.1.4 ShowProfile() test case – Negative

Test case object: 사용자, 멘티 사용자 프로필, 멘토 사용자 프로필, 네트워크, 서버 기반으로 사용자의 프로필이 확인되지 않는 경우 평가

Test Inputs: 프로필 확인 요청 입력, 네트워크 연결 불안정 상태

Expected Results: 프로필 확인 Failed message

### 3.1.5 ModifyProfile() test case – Positive

Test case object: 사용자, 멘티 사용자 프로필, 멘토 사용자 프로필, 서버 기반으로 프로필 수정이 정상적으로 동작하는지 평가

Test Inputs: 수정하려는 새 비밀번호 영어 소문자/숫자 8 자리 이상

Expected Results: 프로필 수정 Success message

### 3.1.6 ModifyProfile() test case – Negative

Test case object: 사용자, 멘티 사용자 프로필, 멘토 사용자 프로필, 서버 기반으로 프로필 수정이 정상적으로 동작하지 않는 경우 평가

Test Inputs: 새 비밀번호 영어 소문자/숫자 8 자리 미만, 숫자 미포함 영어 소문자 8 자리 이상, 영어 소문자 미포함 숫자 8 자리 이상

Expected Results: 프로필 수정 Failed message

## 3.2 멘토 검색

### 3.2.1 SearchMentor() test case – Positive

Test case object: 사용자, 멘토 검색창, 서버, 검색 로그, 추천시스템 기반으로 멘토 사용자가 정상적으로 검색되는지 평가

Test Inputs: 데이터베이스에 존재하는 멘토 사용자 닉네임 또는 매칭 희망분야 검색어

Expected Results: 멘토 검색 Success message

### 3.2.2 SearchMentor() test case – Negative

Test case object: 사용자, 멘토 검색창, 서버, 검색 로그, 추천시스템 기반으로 멘토 사용자가 정상적으로 검색되지 않는 경우 평가

Test Inputs: 데이터베이스에 존재하지 않는 멘토 사용자 닉네임 또는 매칭 희망분야 검색어

Expected Results: 멘토 검색 Failed message

### 3.2.3 RecommendMentor() test case – Positive

Test case object: 멘티 사용자 프로필, 서버, 검색 로그, 추천시스템 기반으로 정상적 추천시스템 동작 평가

Test Inputs: 입력된 올바른 검색어, 멘티 사용자 프로필 매칭 희망분야 속성

Expected Results: 멘토 추천 Success message

### 3.2.4 RecommendMentor() test case – Negative

Test case object: 멘티 사용자 프로필, 서버, 검색 로그, 추천시스템 기반으로 정상이지 않은 추천시스템 동작 평가

Test Inputs: 멘토 검색 Failed message, 멘티 사용자 프로필 매칭 희망분야 속성

Expected Results: 멘토 추천 Failed message

## 3.3 소모임 검색

### 3.3.1 Search 소모임() test case – Positive

Test case object: 사용자, 소모임 검색창, 서버, 소모임 검색 로그, 추천시스템 기반으로 소모임이 정상적으로 검색되는지 평가

Test Inputs: 데이터베이스에 존재하는 소모임명 또는 소모임 관심분야 검색어

Expected Results: 소모임 검색 Success message

### 3.3.2 Search 소모임() test case – Negative

Test case object: 사용자, 소모임 검색창, 서버, 소모임 검색 로그, 추천시스템 기반으로 소모임이 정상적으로 검색되지 않는 경우 평가

Test Inputs: 데이터베이스에 존재하지 않는 소모임명 또는 소모임 관심분야 검색어

Expected Results: 소모임 검색 Failed message

### 3.3.3 Recommend 소모임() test case – Positive

Test case object: 사용자 프로필, 서버, 소모임 검색 로그, 추천시스템 기반으로 정상적 추천시스템 동작 평가

Test Inputs: 입력된 올바른 검색어, 사용자 프로필 소모임 관심분야 속성

Expected Results: 소모임 추천 Success message



### 3.3.4 Recommend 소모임() test case – Negative

Test case object: 사용자 프로필, 서버, 소모임 검색 로그, 추천시스템 기반으로 비정상적 추천시스템 동작 평가

Test Inputs: 소모임 검색 Failed message, 사용자 프로필 소모임 관심분야 속성

Expected Results: 소모임 추천 Failed message

## 3.4 멘토 검색결과

### 3.4.1 RecommendMentor() test case – Positive

Test case object: 멘티 사용자 프로필, 서버, 검색 로그, 추천시스템 기반으로 정상적 추천시스템 동작 평가

Test Inputs: 입력된 올바른 검색어, 멘티 사용자 프로필 매칭 희망분야 속성

Expected Results: 멘토 추천 Success message

### 3.4.2 RecommendMentor() test case – Negative

Test case object: 멘티 사용자 프로필, 서버, 검색 로그, 추천시스템 기반으로 비정상적 추천시스템 동작 평가

Test Inputs: 멘토 검색 Failed message, 멘티 사용자 프로필 매칭 희망분야 속성

Expected Results: 멘토 추천 Failed message

### 3.4.3 ShowMentor() test case – Positive

Test case object: 추천시스템, 멘토 검색결과창, 사용자 기반으로 정상적 멘토 검색결과 동작 평가

Test Inputs: 추천시스템 기반 검색어 관련 멘토 사용자 닉네임 리스트

Expected Results: 멘토 검색결과 Success message

### 3.4.4 ShowMentor() test case – Negative

Test case object: 추천시스템, 멘토 검색결과창, 사용자 기반으로 비정상적 멘토 검색결과 동작 평가

Test Inputs: 검색어 관련 유효한 값이 없는 빈 리스트

Expected Results: 멘토 검색결과 Failed message

### 3.4.5 ShowDetail() test case – Positive

Test case object: 멘토 검색결과창, 멘토 사용자 프로필, 사용자, 네트워크, 서버 기반으로 멘토 상세정보가 정상적으로 제시되는지 평가

Test Inputs: 특정 멘토 사용자 프로필 상세정보 요청, 사용자가 클릭한 특정 멘토 사용자 프로필 닉네임, 매칭 희망분야, 소모임 관심분야, 매칭 이력, 별점, 후기 속성, 네트워크 연결 안정 상태

Expected Results: 멘토 상세정보 Success message

### 3.4.6 ShowDetail() test case – Negative

Test case object: 멘토 검색결과창, 멘토 사용자 프로필, 사용자, 네트워크, 서버 기반으로 멘토 상세정보가 정상적으로 제시되지 않는 경우 평가

Test Inputs: 특정 멘토 사용자 프로필 상세정보 요청, 네트워크 연결 불안정 상태

Expected Results: 멘토 상세정보 Failed message

## 3.5 소모임 검색 결과

### 3.5.1 소모임 추천 테스트 케이스

Test Case Object : 사용자의 정보가 입력되었을 때 이와 적합한 소모임이 추천 되는지 평가

Test Inputs : 사용자의 관심 정보

Expected Results : 사용자의 관심 정보와 관련 있는 소모임 리스트 연산

### 3.5.2 소모임 리스트 출력 테스트 케이스 – Positive

Test Case Object : 적절한 검색어가 입력되었을 때 검색 연산 함수가 어떻게 작동하는지 평가

Test Inputs : 임의의 검색어

Expected Results : 검색 성공 메시지와 관련 소모임 리스트 출력

### 3.5.3 소모임 리스트 출력 테스트 케이스 – Negative

Test Case Object : 적절하지 않은 검색어가 입력되었을 때 검색 연산 함수가 어떻게 작동하는지 평가

Test Inputs : 적절하지 않은 임의의 검색어

Expected Results : 검색 실패 메시지 출력

### 3.5.4 소모임 세부정보 출력 테스트 케이스

Test Case Object : 소모임 리스트에서 한 소모임을 선택했을 때 세부정보가 잘 출력되는지 평가

Test Inputs : 특정 소모임 선택

Expected Results : 소모임 소개와 관심분야, 소모임 가입 신청 버튼과 소모임장 정보

## 3.6 매칭 정보

### 3.6.1 매칭 요청 테스트 케이스 - Positive

Test Case Object : 유효한 멘토 사용자에게 매칭 요청 신청이 접수되는지 평가

Test Inputs : 유효한 사용자에게 매칭 요청 신청

Expected Results : 매칭 요청 성공 메시지 출력

### 3.6.2 매칭 요청 테스트 케이스 - Negative

Test Case Object : 유효하지 않은 멘토 사용자에게 매칭 요청 신청이 접수되지 않는지 평가

Test Inputs : 유효하지 않은 사용자에게 매칭 요청 신청

Expected Results : 매칭 요청 실패 메시지 출력

### 3.6.3 요청된 매칭 신청 수락 테스트 케이스

Test Case Object : 멘토에게 전송된 매칭 요청이 수락되는지 평가

Test Inputs : 임의의 사용자에게서 매칭 요청

Expected Results : 매칭 요청 수락 메시지 출력

### 3.6.4 요청된 매칭 신청 거절 테스트 케이스

Test Case Object : 멘토에게 전송된 매칭 요청이 거절되는지 평가

Test Inputs : 임의의 사용자에게서 매칭 요청

Expected Results : 매칭 요청 거절 메시지 출력

### 3.6.5 매칭 정보 출력 테스트 케이스

Test Case Object : 사용자 개인 프로필에 매칭 정보가 정확히 출력되는지 평가

Test Inputs : 사용자의 프로필 세부사항 클릭

Expected Results : 사용자의 현재 매칭 정보 출력 – 매칭 없음, 매칭 멘토 유저 아이디

## 3.7 채팅

### 3.7.1 채팅로그 불러오기 테스트 케이스 - Positive

Test Case Object : 이전에 나눴던 채팅을 데이터베이스에서 불러오는지 평가

Test Inputs : 과거 채팅 이력이 있는 채팅방에 참여

Expected Results : 이전에 나눈 채팅 데이터 출력

### 3.7.2 채팅로그 불러오기 테스트 케이스 - Negative

Test Case Object : 이전에 나뉘던 채팅을 데이터베이스에서 불러올 수 없는 경우 평가

Test Inputs : 과거 채팅 이력이 있는 채팅방에 참여

Expected Results : 채팅 로그 출력 실패 메시지 출력

### 3.7.3 메시지 전송 테스트 케이스 - Positive

Test Case Object : 작성한 메시지가 대화 상대방에게 전송되는지 평가

Test Inputs : 임의의 문구 메시지

Expected Results : 임의의 메시지가 채팅방에 출력됨

### 3.7.4 메시지 전송 테스트 케이스 - Negative

Test Case Object : 작성한 메시지가 대화 상대방에게 전송되지 않은 경우 평가

Test Inputs : 임의의 문구 메시지

Expected Results : 메시지 전송 실패 알림과 재전송 메시지 출력

### 3.7.5 메시지 답변 전송 테스트 케이스 - Positive

Test Case Object : 작성한 답변 메시지가 대화 상대방에게 전송되는지 평가

Test Inputs : 임의의 문구 메시지

Expected Results : 임의의 메시지가 채팅방에 출력됨

### 3.7.6 메시지 전송 테스트 케이스 - Negative

Test Case Object : 작성한 답변 메시지가 대화 상대방에게 전송되지 않은 경우 평가

Test Inputs : 임의의 문구 메시지

Expected Results : 답변 메시지 전송 실패 알림과 재전송 메시지 출력

## 3.8 리뷰

### 3.8.1 리뷰 작성 테스트 케이스 - Positive

Test Case Object : 매칭 상대 리뷰 작성 테스트 평가

Test Inputs : 매칭 상대에 대한 리뷰 메시지

Expected Results : 리뷰 작성 완료 메시지 출력

### 3.8.2 리뷰 작성 테스트 케이스 - Negative

Test Case Object : 매칭 상대 리뷰 작성 테스트 실패 평가

Test Inputs : 매칭 상대에 대한 리뷰 메시지

Expected Results : 리뷰 작성 실패 메시지 출력

### 3.8.3 프로필 리뷰 출력 테스트 케이스 - Positive

Test Case Object : 입력된 리뷰 프로필 출력 테스트 평가

Test Inputs : 데이터베이스에 저장된 타 사용자의 리뷰 메시지

Expected Results : 프로필에 리뷰 메시지 출력

### 3.8.4 프로필 리뷰 출력 테스트 케이스 - Negative

Test Case Object : 입력된 리뷰 프로필 출력 실패 테스트 평가

Test Inputs : 데이터베이스에 저장된 타 사용자의 리뷰 메시지

Expected Results : 프로필에 리뷰 업데이트 실패 메시지 출력

### 3.8.5 별 점수 업데이트 테스트 케이스 - Positive

Test Case Object : 입력된 별 점수 계산 후 출력 평가

Test Inputs : 데이터베이스에 저장된 타 사용자의 별 점수

Expected Results : 프로필에 기존의 별 점수와 함께 계산된 현재 사용자의 별 점수 출력

### 3.8.6 별 점수 업데이트 테스트 케이스 - Negative

Test Case Object : 입력된 별 점수 계산 후 출력 실패 테스트 평가

Test Inputs : 데이터베이스에 저장된 타 사용자의 별 점수

Expected Results : 프로필에 별 점수 업데이트 실패 메시지 출력

## 4. Software Interface Test

### 4.1. Firebase

#### 4.1.1. Connection Test

Test case object: Firebase 의 외부 소프트웨어 인터페이스를 확인한다.

Test Input: Ping Command

Expected Result: Ping Status

#### 4.1.2. Android App Testing

Test case object: Firebase 호스팅하는 다양한 Android 기기에 테스트를 합니다.

Test Input: Compatibility and Execution Command

Expected Result: Compatibility status and Execution status

#### 4.1.3. Physical Devices Testing

Test case object: 설치되어 실행되는 기기로 앱을 테스트하므로 특정 기기 및 구성에서만 발생하는 문제를 찾는 테스트를 합니다.

Test Input: Version command, language command, and device command

Expected Result: Version Status, Language Status, and Device Status

#### 4.1.4. Workflow Integration Testing

Test case object: 시스템들에 호환성 과 통합 실행이 테스트를 합니다.

Test Input: Compatibility Command and Integration Command

Expected Result: Compatibility Status and Integration Status

## 4.2. Database

### 4.2.1. Schema Test

Test case object: 데이터베이스와 관련된 다양한 스키마 형식을 검증 하고 tables/ views/ columns 의 매핑 형식이 사용자 인터페이스의 매핑 형식과 호환되는지 여부를 검증하고 테스트를 합니다.

Test Input: Compatibility Command, Mapping Formats, and User interface

Expected Result: Compatibility Status, Mapping Formats status, and User interface status

### 4.2.2. Database Tables and Columns Test

Test case object: 백엔드의 데이터베이스 필드 및 열 매핑이 프론트엔드의 매핑과 호환되는지 여부를 테스트를 합니다.

Test Input: Length and Naming Convention Command, Datatype Command, and Field Lengths Command

Expected Result: Length and Naming Convention Status, Datatype Status, and Field Lengths Status

### 4.2.3. Keys and Indexes Test

Test case object: 외부 키에 대한 참조가 유효한지 확인하고 기본 키와 foreign key 의 데이터 유형이 두 테이블에서 동일한지 그리고 모든 키 및 인덱스가 규칙을 따랐는지 필수 필드 및 인덱스의 크기와 길이를 확인하고 테스트를 합니다.

Test Input: Primary key Command, Foreign key Command, Clustered Indexes Command, and Non-Clustered Indexes Command



Expected Result: Primary key Status, Foreign key Status, Clustered Indexes Status, and Non-Clustered Indexes Status

#### 4.2.4. Database Server Validation Test

Test case object: 요구 사항에 지정된 재로 데이터베이스 서버 구성을 확인하고 응용 프로그램에 필요한 작업이 수행하는지 그리고 사용자의 권한 및 지정된 최대 허용 사용자 트랜잭션 수의 요구 사항을 충족할 수 있는지 확인을 하고 테스트를 합니다.

Test Input: Transaction Command, Authorized user actions Command, and Database Server Command

Expected Result: Transaction Status, Authorized user actions Status, and Database Server Status

#### 4.2.5. Trigger Test

Test case object: 필요한 업데이트/삽입/ 삭제를 확인 하고 테스트 대상 응용 프로그램 영역에서 기능을 트리거 테스트를 합니다.

Test Input: Update Command, Insert Command, Delete Command

Expected Result: Update Status, Insert Status, Delete Status

#### 4.2.6. Functional Database Test

Test case object: 데이터베이스와 관련된 end-user 가 수행하는 트랜잭션 및 작업이 예상대로 작동하는지 테스트를 합니다.

Test Input: Null Command, Field Length Command, and Similar Field Command

Expected Result: Null Status, Field Length Status, and Similar Field Status

#### 4.2.7. Stored Procedure Test

Test case object: 모든 모듈의 저장 프로시저를 테스트 대상 응용 프로그램에 대한 테스트를 합니다.

Test Input: Module Command, and Application Command

Expected Result: Module Status, and Application Command Status

## 4.3. Application Programming Interface (API)

### 4.3.1. Verifying Status Code

Test case object: HTTP status code 가 정확하게 작성이 되어 있는지 테스트를 합니다.

Test Input: HTTP Status Code Command

Expected Result: HTTP Status Code Status

### 4.3.2. Verifying Payload

Test case object: 응답 페이로드를 확인을 하고 테스트를 합니다.

Test Input: JSON body Command, Filed names, types, and values Command

Expected Result: JSON body Status, Filed names, types, and values Status

### 4.3.3. Verifying Response Header

Test case object: 응답 헤더를 확인을 하고 테스트를 합니다.

Test Input: Security and Performance Command, Ping Command

Expected Result: Security and Performance Status, Ping Status

### 4.3.4. Verifying Application State

Test case object: Manual 테스트, UI 테스트, 다른 인터페이스들을 쉽게 검사할 수 있습니다.

Test Input: UI Command, Interface Command

Expected Result: UI Status, Interface Status

## **4.4. Apache**

### **4.4.1. Valid Response Test**

Test case object: 웹 서버가 응답이 유효한지 확인을 하면서 테스트를 합니다.

Test Input: Web Server Return Command

Expected Result: Web Server Return Status

### **4.4.2. Website Response Test**

Test case object: 유저들 한테 얼마나 빠르게 응답이 되는지를 테스트를 합니다.

Test Input: Respond Command

Expected Result: Respond Status

### **4.4.3. Connection Test**

Test case object: Apache 의 연결 상태를 테스트를 합니다.

Test Input: Ping Command

Expected Result: Ping Status

## 5. Supporting Information

### 5.1 Document History

Date	Version	Description	Writer
5/20	1.0	1	김정훈
5/21	1.1	3.1, 3.2	김시인
5/22	1.2	3.5,3.6	김민지
5/23	1.3	4	서기용
5/25	1.4	2	김정훈
5/26	1.5	3.3, 3.4	김시인
5/27	1.6	3.7, 3.8	김민지
5/28	1.7	4 수정/추가	서기용
5/29	1.8	취합/수정/최종	김정훈