



SKKU GROUP BUY Application

Software Test Specification

2022.11.13.

Introduction to Software Engineering

TEAM 7 (SKKU GROUP BUY Application)

Team Leader	Kyungmin Min
Team Member	Jaesung Lee
Team Member	Jeongjae Park
Team Member	Jongyoon Kim
Team Member	Seongyun Lee
Team Member	Yiqin Wei

1. Introduction	7
1.1. Purpose	7
1.2. Scope	7
1.3. Definitions, Acronyms, and Abbreviation	7
1.4. References	8
1.5. Overview	9
2. Approach	9
2.1. Test method	9
2.1.1. Software unit test methods	9
2.1.2. Software interface test methods	12
2.2. Test Assumptions	13
2.2.1. Key Assumptions	13
2.2.2. General Assumptions	13
2.3. Testing tools	13
3. Unit-Test	13
3.1. Login page	14
3.1.1. Register	14
3.1.2. Login	15
3.2. Main page	19
3.3. Tab Navigator	22
3.4. Chat room page	24
3.5. Host page	27
3.6. My page	28
3.7. Post detail page	31
3.8. Review page	34
3.9. Create page	35
4. Software Interface Test	36
4.1. Objectives	36
4.2. Functional Test for API server	36
4.3. Test for API server to prevent system failure	38
4.4. Evaluate API Server	39

4.5. Software Interface Test Method	40
5. Supporting Information	41
5.1. Test Environment	41
5.2. Staffing	41
5.3. Document History	41

LIST OF FIGURES

[Figure 1] Unit test pseudo-code for Register	10
[Figure 2] Unit test pseudo-code for Login	10
[Figure 3] Unit test pseudo-code for Join a group	10
[Figure 4] Unit test pseudo-code for Search	11
[Figure 5] Unit test pseudo-code for Create new item	11
[Figure 6] Unit test pseudo-code for Review	11
[Figure 7] Unit test pseudo-code for Chat room	12
[Figure 8] Unit test pseudo-code for Profile	12

LIST OF TABLES

[Table 1] Test case for Login page 1	14
[Table 2] Test case for Login page 2	15
[Table 3] Test case for Login page 3	15
[Table 4] Test case for Login page 4	16
[Table 5] Test case for Login page 5	16
[Table 6] Test case for Login page 6	17
[Table 7] Test case for Login page 7	18
[Table 8] Test case for Main page 1	19
[Table 9] Test case for Main page 2	20
[Table 10] Test case for Main page 3	20
[Table 11] Test case for Main page 4	21
[Table 12] Test case for Main page 5	21
[Table 13] Test case for Main page 6	22
[Table 14] Test case for Tab navigator 1	22
[Table 15] Test case for Tab navigator 2	23
[Table 16] Test case for Tab navigator 3	23
[Table 17] Test case for Tab navigator 4	23
[Table 18] Test case for Chat room page 1	24
[Table 19] Test case for Chat room page 2	25
[Table 20] Test case for Chat room page 3	25
[Table 21] Test case for Chat room page 4	26
[Table 22] Test case for Chat room page 5	27
[Table 23] Test case for Host page 1	27
[Table 24] Test case for Host page 2	28
[Table 25] Test case for My page 1	29

[Table 26] Test case for My page 2	29
[Table 27] Test case for My page 3	30
[Table 28] Test case for My page 4	31
[Table 29] Test case for My page 5	31
[Table 30] Test case for Post detail page 1	32
[Table 31] Test case for Post detail page 2	32
[Table 32] Test case for Post detail page 3	33
[Table 33] Test case for Post detail page 4	33
[Table 34] Test case for Review page 1	34
[Table 35] Test case for Create page 1	35
[Table 36] Functional test for Register user	37
[Table 37] Functional test for Search items	37
[Table 38] Functional test for Join a group buying	37
[Table 39] Functional test for Post a suggestion	37
[Table 40] Functional test for Change user profile image	38
[Table 41] System failure test for Abnormally repeated requests	38
[Table 42] System failure test for Invalid request payload (parameter)	38
[Table 43] System failure test for Accessing user data abnormally	39
[Table 44] System failure test for Excessive traffic	39
[Table 45] Evaluation test for API server response time	39
[Table 46] Evaluation test for Traffic capacity	40
[Table 47] Testing Environment	41
[Table 48] Test Staffing	41
[Table 49] Document History	41

1. Introduction

Introduction section will describe the purpose of this Test Plan Specification document, the scope of the document, definitions, acronyms, and abbreviation of the words, references, and the overall contents of the document.

1.1. Purpose

This document is a test plan for the “SKKU GROUP BUY APP”. The purpose of the document is to describe a detailed description of testing methods, so that the team members can conduct the testing consistently and conduct more accurate communication between team members.

1.2. Scope

This document provides “SKKU GROUP BUY APP” testing information. The information contains pseudo code of test case, purpose of the test, input/output of the test and test methods.

There are two kinds of testing. One is a unit test and the other is an interface test. Unit test is testing the basic elements that make up the system, unit, operating normally. Interface test is conducted for all external interfaces used by the system. Also, Interface test will check the external interface operating normally.

1.3. Definitions, Acronyms, and Abbreviation

API	Application Program Interface
-----	-------------------------------

Unit	A unit is the most basic testable part of a system
------	--

Interface	The part that connects two systems or components to each other
-----------	--

Component	Component is a functional unit consisting of a combination of units
-----------	---

User	Someone who uses a system
Host	User who hosts the group buying
Participant	User who participates the group buying
System administrator	Someone who checks the overall process of registration, login and manages the system
Client (user device)	A user device/user that connected to server
Server	A computer or computer program which manages access to a centralized resource or service in a network
Software	The programs and other operating information used by a computer
Query	A request to display specific data from a database
Database	A set of data that is integrated and managed for the purpose of being shared and used by multiple people

1.4. References

- 1 “Software Test Specification”. SKKU 2021 Introduction to Software Engineering Team7.

https://github.com/skkuse/2021spring_41class_team7/blob/main/doc/Team7_STS.pdf

- 1 “Software Test Specification”. SKKU 2021 Introduction to Software Engineering Team12.

https://github.com/skkuse/2021spring_41class_team12/blob/main/Test%20Plan%20Specification_Team%2012.docx

1 IEEE Std 830-1998

1.5. Overview

The document consists of 5 sections. Section 1 is introduction. Introduction section describes the overall contents of the document. Section 2 is about an approach on how to conduct the test. Also, section 2 shows testing methods, test assumptions and testing tools. Section 3 and 4 describe the Unit Test and Interface Test, respectively. Section 3 describes the procedure of the test case and input/output of the test case for each function testing. Section 4 describes the test plan for external systems for this mobile application including the API server. Section 5 is about supporting information. Section 5 describes test environment, staffing and document history.

2. Approach

This section describes testing approaches. It briefly introduces software unit testing and software interface testing. Also, an introduction to pre-test assumptions and testing tools is described. This part is a summary of the tests since more detailed tests will be included in the following documentation. In the Methods section, pseudocode is used to abstractly outline a rough idea about the testing.

2.1. Test method

2.1.1. Software unit test methods

This part uses pseudo-code expressing the contents of the software unit test.

[Figure 1] Unit test pseudo-code for Register

algorithm 1 Register

input: *a, i* ▷ get user email address and information

```

1: if a == new then
2:   if a == valid then ▷ if it is a university email
3:     success;
4:   else
5:     fail;
6:   end if
7: else ▷ if it is a used email
8:   fail;
9: end if
10:
11: if i == valid then
12:   success;
13: else
14:   fail;
15: end if

```

[Figure 2] Unit test pseudo-code for Login

algorithm 2 Login

input: *id, psw* ▷ get user account information

```

1: if id == valid then
2:   if psw == valid then ▷ if password is correct
3:     success;
4:   else ▷ if password is wrong
5:     fail;
6:   end if
7: else ▷ if it is an unregistered ID
8:   fail;
9: end if

```

[Figure 3] Unit test pseudo-code for Join a group

algorithm 3 Join

test1 = Different users join at the same time

test2 = Join success

input: *p* ▷ press the join button

```

1: if p == valid then
2:   success;
3: else
4:   fail;
5: end if

```

[Figure 4] Unit test pseudo-code for Search

algorithm 4 Search

test1 = Search results delivery time
test2 = search precision
test3 = test inputting key word

input: *key* ▷ get the input key word

1: **while** *title, contents == key* **do**
2: *display*; ▷ If any matches
3: **end while**

[Figure 5] Unit test pseudo-code for Create new item

algorithm 5 Create

input: *i* ▷ get item information

1: **if** *i==valid* **then**
2: *success*;
3: **else** ▷ If there is any information blank or illegal
4: *fail*;
5: **end if**

[Figure 6] Unit test pseudo-code for Review

algorithm 6 Review

input: *len* ▷ length of a review

1: **if** $0 < len \leq 200$ **then**
2: *success*;
3: **end if**
4:
5: **if** $len > 200$ **then**
6: **if** $len \leftarrow 200$ **then**
7: *success*;
8: **else**
9: *fail*;
10: **end if**
11: **end if**

[Figure 7] Unit test pseudo-code for Chat room

algorithm 7 Chat Room

Test1 = Chat Room Enter
 Test2 = Chat Room exit
 Test3 = Send Message

input: *len* ▷ message length

```

1: if  $0 < len \leq 200$  then
2:   success;
3: else ▷ messagelength > 200
4:   if warning then
5:     success;
6:   else
7:     fail;
8:   end if
9: end if
  
```

[Figure 8] Unit test pseudo-code for Profile

algorithm 8 Profile

Test1 = Profile open timing(both user and host)
 Test2 = Profile viewing(both user and host)
 Test3 = Set Profile(user)

input: *i* ▷ get user information

```

1: if  $i == \text{valid}$  then
2:   success;
3: else
4:   fail;
5: end if
  
```

2.1.2. Software interface test methods

Software interface testing is part of integration testing, so tests will be done on external systems, including the API server on our application. Normal and abnormal use cases are covered when testing. Databases and servers that support multi-threaded situations will be tested for concurrency errors with emphasis. Detailed test cases and methods will be introduced in Chapter 4.

2.2. Test Assumptions

2.2.1. Key Assumptions

Assume that software can be adapted to different environments provided and data is defaulted to be valid unless testing is required.

2.2.2. General Assumptions

1. Assuming all test results will be recorded by taking screenshots.
2. Assuming that testing will be stopped immediately as soon as a defect occurs. After fixing defects, the test should be started from scratch.
3. Assuming the test environment consists of mobile devices and emulators.
4. Assuming the network environment includes wifi, hotspots, and mobile networks.
5. Assuming each team member is familiar with all the test information.
6. Assuming all the team members should share any shortcomings and changes in the test plan regardless of their own part.
7. Assuming every personnel internal and external conducts testing should have the ability to read the requirements and test plan documents.

2.3. Testing tools

Unit tests will be generated by Android Studio automatically, and also will be executed by Android Studio. Tests of interfaces will be done on mobile devices by team members manually.

3. Unit-Test

In this chapter, we will show the scope of the unit test based on each page. We have categorized the functions to be tested on each page. There are several functions on each page, and we designed validation testing and defect testing for each function to test them.

From now on, we will show functions in the categorized page and its validation testing and defect testing.

3.1. Login page

3.1.1. Register

In the case of a user who has already registered, a new register should not be possible.

1. Click “회원 가입하기” button
2. Fill the user information text boxes with various cases of input
3. Click “회원 가입하기” button
4. Wait for the output message

<Test case>

[Table 1] Test case for Login page 1

(Success case)	
Input	Non-existent user information
Output message	“Success to register”
(Failure case)	
Input	Existing user information
Output message	“Existing user. Go to the log-in page.”

An error message is sent for an invalid email verification.

1. Click “회원 가입하기” button
2. Enter a student email
3. Click “인증 번호 받기” button
4. Enter a verification code
5. Wait for the output message

<Test case>

[Table 2] Test case for Login page 2

(Success case)	
Input	Student email, verification code
Output message	“Verification was successful”
(Failure case)	
Input	Student email, verification code
Output message	“Invalid verification. Try again with valid student email”
(Failure case)	
Input	Student email, verification code
Output message	“Invalid verification. Please enter valid verification code”

An error message is sent if a user input wrong password verification.

1. Click “회원 가입하기” button
2. Fill “비밀번호 입력” text box
3. Fill “비밀번호 확인” text box
4. Wait for the output message

<Test case>

[Table 3] Test case for Login page 3

(Success case)	
Input	a1b2c3, a1b2c3
Output message	“Password verification success”
(Failure case)	
Input	a1b2c3, sdfkdjs
Output message	“Password verification fail”

3.1.2. Login

Minimum (0) and maximum (20) lengths should be set for all the text boxes.

1. Fill “이메일” text box
2. Fill “비밀번호” text box
3. Click “로그인” button
4. Check for the textbox result

<Test case>

[Table 4] Test case for Login page 4

(Success case)	
User email input (keyboard)	thisisuseremail@skku.edu
Textbox result	thisisuseremail@skku.edu
(Failure case)	
User email input (keyboard)	thisisuseremail@skku.edu
Textbox result	sdfsdfsfssa!ddsd.sds

Blank should not be set for all the text boxes.

1. Fill “이메일” text box
2. Fill “비밀번호” text box
3. Click “로그인” button
4. Check for the textbox result

<Test case>

[Table 5] Test case for Login page 5

(Success case)	
User email input (keyboard)	thisisuseremail@skku.edu

Textbox result	thisisuseremail@skku.edu
(Failure case)	
User email input (keyboard)	this is user email@skku.edu
Textbox result	thisisuseremail@skku.edu

Passwords should be displayed in masked format.

1. Fill “비밀번호” text box
2. Click “로그인” button
3. Check for the textbox result

<Test case>

[Table 6] Test case for Login page 6

(Success case)	
User password input (keyboard)	thisispass
Textbox result	●●●●●●●●●●
(Failure case)	
User password input (keyboard)	thisispass
Textbox result	thisispass

Validation message should be shown when invalid user email and/or password is entered, and the fields are left blank.

1. Fill “이메일” text box
2. Fill “비밀번호” text box
3. Click “로그인” button
4. Check for the output message

<Test case>

[Table 7] Test case for Login page 7

(Success case)			
Real user email	thisisuseremail@s kku.edu	Real user password	realpass
Input user email	thisisuseremail@s kku.edu	Input user password	realpass
Output message		“Success to access the account”	
(Failure case)			
Real user email	thisisuseremail@s kku.ed	Real user password	realpass
Input user email	thisisuseremail@s kku.ed	Input user password	thisispass
User email textbox result		(blank)	
User password textbox result		(blank)	
Output message		“Invalid user email or password. Try again!”	
(Failure case)			
Real user email	thisisuseremail@s kku.ed	Real user password	realpass
Input user email	useremail	Input user password	realpass
User email textbox result		(blank)	
User password textbox result		(blank)	
Output message		“Invalid user email or password. Try again!”	
(Failure case)			
Real user email	thisisuseremail@s kku.ed	Real user password	realpass
Input user email	useremail	Input user password	thisispass
User email textbox result		(blank)	

User password textbox result	(blank)
Output message	“Invalid user email or password. Try again!”

3.2. Main page

Search results should be displayed when the user clicks the search button.

1. Click “검색어 입력” textbox
2. Write search keyword in the textbox
3. Click “검색” button
4. Wait for the result

<Test case>

[Table 8] Test case for Main page 1

(Success case)	
Motion	Write “생수” and click “검색” button
Search result	Group Buying lists which contains “생수”
(Failure case)	
Motion	Write “생수” and click “검색” button
Search result	There are no items which contains “생수” keyword

Search result displayed should be relevant to the search keyword.

1. Click “검색어 입력” textbox
2. Write search keyword in the textbox
3. Click “검색” button
4. Wait for the search result

<Test case>

[Table 9] Test case for Main page 2

(Success case)	
Motion	Write the keyword in the textbox.
Input content	“생수”
Search result	Group Buying lists which contains “생수”
(Failure case)	
Motion	Write the keyword in the textbox.
Input content	“생수”
Search result	Group Buying lists which contains “과자”

Each time a user enters the “Main page”, the group buying items stored on the server must be listed.

1. Enter the “Main page”
2. Check for the result

<Test case>

[Table 10] Test case for Main page 3

(Success case)	
Motion	Enter the main page.
Input content	User button click
Expected result	Show Group Buying lists from Server
(Failure case)	
Motion	Enter the main page.
Input content	User button click, Lose network connection
Expected result	Show default

When importing group buying items, items with expired periods should not be displayed on the main page.

<Test case>

[Table 11] Test case for Main page 4

(Success case)	
Motion	Enter the main page.
Input content	User button click, Items that have expired
Expected result	It does not appear on the main page normally.
(Failure case)	
Motion	Enter the main page.
Input content	User button click, Items that have expired
Expected result	It appears on the main page.

When importing group buying items, items which are full of target people should not be displayed on the main page.

<Test case>

[Table 12] Test case for Main page 5

(Success case)	
Motion	Enter the main page.
Input content	User button click, Items which are full of target people
Expected result	It does not appear on the main page normally.
(Failure case)	
Motion	Enter the main page.
Input content	User button click, Items which are full of target people
Expected result	It appears on the main page.

The number of participants in group buying items and the progress bar indicating them must be continuously updated.

<Test case>

[Table 13] Test case for Main page 6

(Success case)	
Motion	Click the “참여하기” button in test-case item.
Input content	User button click
Item result	“3명 남았어요” becomes “2명 남았어요”
Progress bar result	Progress bar becomes closer to the target
(Failure case)	
Motion	Click the “참여하기” button in test-case item.
Input content	User button click
Item result	“3명 남았어요” becomes “3명 남았어요”
Progress bar result	No change in progress bar

3.3. Tab Navigator

User can enter “Main page” by clicking “홈” button in tab navigator.

<Test case>

[Table 14] Test case for Tab navigator 1

(Success case)	
Input	Click “홈” button
Output Page	Enter the corresponding “Main page”
(Failure case)	
Input	Click “홈” button
Output Page	Any other pages rather than the corresponding “Main page”

User can enter “Chat room page” by clicking “채팅” button in tab navigator.

<Test case>

[Table 15] Test case for Tab navigator 2

(Success case)	
Input	Click “채팅” button
Output Page	Enter the corresponding “Chat room page”
(Failure case)	
Input	Click “채팅” button
Output Page	Any other pages rather than the corresponding “Chat room page”

User can enter “My page” by clicking “마이 페이지” button in tab navigator.

<Test case>

[Table 16] Test case for Tab navigator 3

(Success case)	
Input	Click “마이 페이지” button
Output Page	Enter the corresponding “My page”
(Failure case)	
Input	Click “마이 페이지” button
Output Page	Any other pages rather than the corresponding “My page”

The tab navigator should appear at the bottom of the screen only on “Main page”.

<Test case>

[Table 17] Test case for Tab navigator 4

(Success case)	
Input	Enter another page except “Main page”

Output Result	Hide Tab Navigator
(Success case)	
Input	Enter “Main page”
Output Result	Show Tab Navigator
(Failure case)	
Input	Enter another page except “Main page”
Output Result	Show Tab Navigator
(Failure case)	
Input	Enter “Main page”
Output Result	Hide Tab Navigator

3.4. Chat room page

The length of the chat should be less than or equal to 200.

1. Enter the “Chat room page”
2. Write some text in “메시지 보내기” field
3. Click the send button

<Test case>

[Table 18] Test case for Chat room page 1

(Success case)	
Input	String, which has length ≤ 200
Result	String is sent to the chat
(Failure case)	
Input	String, which has length > 200
Result	String is not sent to the chat
Output Message	“채팅은 200자 이하로 제한됩니다”

User can enter “Post detail page” by clicking “제품 안내 페이지로 이동” on the top of the “Chat room page”.

1. Enter the “Chat room page”
2. Click “제품 안내 페이지로 이동” button

<Test case>

[Table 19] Test case for Chat room page 2

(Success case)	
Input	Click “제품 안내 페이지로 이동” button
Output Page	Enter the corresponding “Post detail page”
(Failure case)	
Input	Click “제품 안내 페이지로 이동” button
Output Page	Any other pages rather than the corresponding “Post detail page”

User can set a notice for a chat room by toggling (clicking) the loudspeaker button and sending a message.

1. Enter the “Chat room page”
2. Click the loudspeaker button
3. Write some text in “메시지 보내기” field
4. Click the send button

<Test case>

[Table 20] Test case for Chat room page 3

(Success case)	
Input	String, which has length ≤ 200
Result	1. Existing notice is not a notice anymore 2. Sent message becomes a new notice
(Failure case)	
Input	String, which has length > 200

Result	String is not sent to the chat
Output Message	“채팅은 200자 이하로 제한됩니다”
(Failure case)	
Input	String, which has length ≤ 200
Result	New notice is not registered
(Failure case)	
Input	String, which has length ≤ 200
Result	Existing notice remains

If user receives the product, by clicking “제품 수령 완료” button, user can exit the chat room and write a review for the host user.

<Test Case>

[Table 21] Test case for Chat room page 4

(Success case)	
Input	Participant user clicks “제품 수령 완료” button
Output Page	Participant user enters “Review page”
(Success case)	
Input	Host user clicks “제품 수령 완료” button
Output Page	Host user enters “Main page”
(Failure case)	
Input	Participant user clicks “제품 수령 완료” button
Output Page	Participant user does not enter “Review page”
(Failure case)	
Input	Host user clicks “제품 수령 완료” button
Output Page	Host user enters “Review page”

If the network connection is lost, user typed text will not be sent to the chat room, and the application will notify the user that the network connection is lost.

<Test case>

[Table 22] Test case for Chat room page 5

(Success case)	
Input	String, which has length ≤ 200
Condition	Network connection lost
Result	The text is not sent to the chat room
Output Message	“네트워크 연결 상태가 좋지 않습니다”
(Failure case)	
Input	String, which has length ≤ 200
Condition	Network connection lost
Result	The text is not sent to the chat room
Output Message	No notification
(Failure case)	
Input	String, which has length ≤ 200
Condition	Network connection lost
Result	The text is sent to the chat room

3.5. Host page

The “Host page” should print out the average score of the host user.

<Test case>

[Table 23] Test case for Host page 1

(Success case)	
Input	Participant user leaves a review score (Not in “Host page”)
Result	Review score is reflected in the host user’s score as average

(Failure case)	
Input	Participant user leaves a review score (Not in “Host page”)
Result	Host user’s score is not changed
(Failure case)	
Input	Participant user leaves a review score (Not in “Host page”)
Result	Host user’s score is changed, but it is not the average

By clicking “참여하기” button, the user can participate the corresponding group buying.

<Test case>

[Table 24] Test case for Host page 2

(Success case)	
Input	Click “참여하기” button
Output Page	Enter the corresponding “Post detail page”
(Failure case)	
Input	Click “참여하기” button
Output Page	Any other pages rather than corresponding “Post detail page”

The list of the created group buying is updated in real time.

3.6. My page

User can change the profile image by clicking “프로필 사진 변경” button and uploading png file. The file size is up to 1 MB.

<Test case>

[Table 25] Test case for My page 1

(Success case)	
Input	.png file, file size <= 1 MB
Result	User profile image changes
(Failure case)	
Input	.png file, file size > 1 MB
Result	User profile image does not change
Output Message	“파일 크기는 1 MB를 초과할 수 없습니다”
(Failure case)	
Input	.jpg file, or any file with other extension
Result	User profile image does not change
Output Message	“프로필 이미지는 png 파일이어야 합니다”

User can change their nickname and password in “My page”. The nickname cannot be duplicated between two or more different users. The length of the nickname should be in this range: $2 \leq \text{length} \leq 20$

<Test case>

[Table 26] Test case for My page 2

(Success case)	
Input	“newnickname”
Condition	There is no other user who uses “newnickname” as nickname
Result	User nickname changes
(Failure case)	
Input	“newnickname”
Condition	There is other user who uses “newnickname” as nickname
Result	User nickname does not change
Output Message	“해당 닉네임은 이미 사용중입니다”

(Failure case)	
Input	“A”
Result	User nickname does not change
Output Message	“닉네임은 2자 이상 20자 이하여야 합니다”
(Failure case)	
Input	“12345678901234567890123”
Result	User nickname does not change
Output Message	“닉네임은 2자 이상 20자 이하여야 합니다”

User can change their password in “My page”. To guarantee the minimum security, the password length should be in this range: $3 \leq \text{length} \leq 20$

<Test case>

[Table 27] Test case for My page 3

(Success case)	
Input	“newpassword”
Result	User password changes
(Failure case)	
Input	“A”
Result	User password does not change
Output Message	“비밀번호는 3자 이상 20자 이하여야 합니다”
(Failure case)	
Input	“12345678901234567890123”
Result	User password does not change
Output Message	“비밀번호는 3자 이상 20자 이하여야 합니다”

User can enter “Post detail page” by clicking “바로가기” button in “참여 중인 공동 구매” list.

<Test case>

[Table 28] Test case for My page 4

(Success case)	
Input	Click “바로가기” button
Output Page	Enter corresponding “Post detail page”
(Failure case)	
Input	Click “바로가기” button
Output Page	Does not enter corresponding “Post detail page”

User can deactivate their account and leave the service. If so, every data about the user should be deleted from the server.

<Test case>

[Table 29] Test case for My page 5

(Success case)	
Input	Click “회원 탈퇴하기” button
Output Page	Enter “Start page”
Result	Every data about the user is deleted
(Failure case)	
Input	Click “회원 탈퇴하기” button
Output Page	Does not enter “Start page”
(Failure case)	
Input	Click “회원 탈퇴하기” button
Result	Some data remains

3.7. Post detail page

Users who are not hosts can participate in the group buying by clicking “참여하기” button.

<Test case>

[Table 30] Test case for Post detail page 1

(Success case)	
Input	Click “참여하기” button
Result	1. “참여하기” button disappears and “채팅방으로 이동” button and “공동 구매 탈퇴” button are created 2. The number of people participating in this group buying increases by 1
Output Message	“공동 구매에 참여했습니다”
(Failure case)	
Input	Click “참여하기” button
Result	Display an error message
Output Message	“마감 기한이 지난 공동 구매는 참여할 수 없습니다”

Users can move to the “Chat room page” of their group buying by clicking the “채팅방으로 이동” button.

<Test case>

[Table 31] Test case for Post detail page 2

(Success case)	
Input	Click “채팅방으로 이동” button
Output Page	Enter the corresponding “Chat room page”
(Failure case)	
Input	Click “채팅방으로 이동” button
Output Page	Any other pages rather than the corresponding “Chat room page”

Users can cancel the group buying you belong to by clicking the “공동 구매 탈퇴” button.

<Test case>

[Table 32] Test case for Post detail page 3

(Success case)	
Input	Click “공동 구매 탈퇴” button
Result	1. “참여하기” button is created and “채팅방으로 이동” button and “공동 구매 탈퇴” button disappear 2. The number of people participating in this group buying decreases by 1
Output message	“공동 구매를 탈퇴하였습니다”
(Failure case)	
Input	Click “공동 구매 탈퇴” button
Result	Display an error message
Output message	“공동 구매 탈퇴에 실패하였습니다”

Host user can cancel the group buying you belong to by clicking “공동 구매 탈퇴” button.

<Test case>

[Table 33] Test case for Post detail page 4

(Success case)	
Input	Click “공동 구매 탈퇴” button
Result	Group buying hosted by the host themselves are gone
Output message	“공동 구매를 삭제하였습니다”
(Failure case)	
Input	Click “공동 구매 탈퇴” button

Result	Display an error message
Output message	“공동 구매 탈퇴에 실패하였습니다”

3.8. Review page

Users who successfully completed a group buying leave a review for the host through text and stars.

<Test case>

[Table 34] Test case for Review page 1

(Success case)	
Input	Click “리뷰 등록하기” button
Condition	Review text is also written and star rating is checked
Result	Host reviews are registered normally
Output message	“리뷰를 등록하였습니다”
(Success case)	
Input	Click “리뷰 등록하기” button
Condition	Review text is not written but star rating is checked
Result	Host reviews are registered normally
Output message	“리뷰를 등록하였습니다”
(Failure case)	
Input	Click “리뷰 등록하기” button
Condition	Review text is written but star rating is not checked
Result	Host reviews are not registered
Output message	“별점을 선택해주십시오”
(Failure case)	
Input	Click “리뷰 등록하기” button
Condition	Review text is not written and star rating is not checked
Result	Host reviews are not registered

Output message	“별점을 선택해주십시오”
----------------	---------------

3.9. Create page

Host users who wish to host a group buying create a group buying.

<Test case>

[Table 35] Test case for Create page 1

(Success case)	
Input	Click “공동 구매 생성하기” button
Condition	Upload an image and “제목”, “목표 인원”, “마감 날짜”, “물품 수령 장소”, “게시글 내용” are written
Result	Group buying is successfully created
Output message	“공동 구매가 생성되었습니다”
(Failure case)	
Input	Click “공동 구매 생성하기” button
Condition	“제목” is not written
Result	Group buying is not created
Output message	“제목을 입력해주십시오”
(Failure case)	
Input	Click “공동 구매 생성하기” button
Condition	“목표 인원” is not written
Result	Group buying is not created
Output message	“목표 인원을 적어주십시오”
(Failure case)	
Input	Click “공동 구매 생성하기” button
Condition	“마감 날짜” is not checked
Result	Group buying is not created
Output message	“마감 날짜를 설정해주십시오”

(Failure case)	
Input	Click “공동 구매 생성하기” button
Condition	“물품 수령 장소” is not written
Result	Group buying is not created
Output message	“물품 수령 장소를 입력해주시요”
(Failure case)	
Input	Click “공동 구매 생성하기” button
Condition	“게시글 내용” is not written
Result	Group buying is not created
Output message	“게시글 내용을 적어주시요”

4. Software Interface Test

4.1. Objectives

In this chapter, we describe the test plan for external systems for this mobile application including the API server. The test plan covers various test cases that may occur in normal use, also including abnormal cases for malicious usage. The server and database are based on multi-threads system, so the errors might occur in concurrency situations. The test plan includes some situations in which concurrency problems might occur such as accessing some database resources from different users(thread) at the same time. In the testing, the performance of external systems also would be measured numerically.

4.2. Functional Test for API server

This section covers a test case for a feature with a variety of contextual actions among the main features.

[Table 36] Functional test for Register user

Test Plan	Send various requests with Email Address
-----------	--

Excepted Result	<p>[Case 1] New Email Address Response with success message, and a verifying code have been sent.</p> <p>[Case 2] Email Address Already Exist Response with failure message.</p>
-----------------	--

[Table 37] Functional test for Search items

Test Plan	Send various requests with search keyword
Excepted Result	Response with JSON data which contains exact results for search query.

[Table 38] Functional test for Join a group buying

Test Plan	Send various requests with Group Buying ID, user token.
Excepted Result	<p>[Case 1] Joining Success Response with success message, and a user joined the group successfully</p> <p>[Case 2] The Group Buying Gathering Finished Response with “Gathering finished” message,</p> <p>[Case 3] Joined Already Response with “Joined Already” message.</p>

[Table 39] Functional test for Post a suggestion

Test Plan	Send various requests with Images and text.
Excepted Result	<p>[Case 1] Posting Success Response with success message, and a suggestion is posted successfully</p> <p>[Case 2] Posting Failure Response with “fail” message,</p>

[Table 40] Functional test for Change user profile image

Test Plan	Send various requests with an image file.
Excepted Result	<p>[Case 1] Changing Success Response with success message, and the profile image changed successfully</p> <p>[Case 2] Changing Failure - Network Error Response with “fail” message,</p>

4.3. Test for API server to prevent system failure

In this section, we list up the test cases for each endpoint of the api server. The features which are critical and security-required are subjected to more sophisticated testing.

[Table 41] System failure test for Abnormally repeated requests

Situation	A user sends an unusually repeated request
Test Plan	Send requests 100 times in 1 minute.
Possible dangers	If the DDOS attack happens, the server goes down.
Handling	Block the user for curtain periods

[Table 42] System failure test for Invalid request payload (parameter)

Situation	A malicious user can send a request with invalid data manually.
Test Plan	Send requests with invalid data.
A possible danger	The code could raise Runtime errors.
Handling	Check all request data and handle the exceptional cases.

[Table 43] System failure test for Accessing user data abnormally

Situation	A malicious user tries to access other user's private data.
Test Plan	Send a request that requests other user's private data.
A possible danger	Personal Information extrusion occurs.
Handling	Deny the unauthorized user to get private data.

[Table 44] System failure test for Excessive traffic

Situation	The number of requests is more excessive than expected.
Test Plan	Send 100 requests at a time.
A possible danger	Response time would be increased, finally the server goes down.
Handling	Set timeout for request.

4.4. Evaluate API Server

This section describes how to evaluate the server performance to figure out if the system satisfies the requirements.

[Table 45] Evaluation test for API server response time

Test Plan	Send a certain amount of requests at a time and measure the average response time of the requests.
Excepted Result	The average response time must be less than the value required.

[Table 46] Evaluation test for Traffic capacity

Test Plan	Increase the amount of requests at a time and find out the amount
-----------	---

	which the server works normally.
Excepted Result	The number of requests that can be processed simultaneously must be greater than the value required.

4.5. Software Interface Test Method

We use Python to execute the test cases above. Several test cases will be created at the same time as the server is developed. We will not send each test request manually, rather python code will execute such test cases automatically. We will use Python Framework for testing “unittest”.

5. Supporting Information

5.1. Test Environment

[Table 47] Testing Environment

Testing Environment	
Multiple Android Device	<ul style="list-style-type: none"> • Android API Level 33 : Galaxy S22 • Android API Level 32 : Google Pixel XL • Android API Level 30: Galaxy S20 • Android API Level 30: Galaxy Z Flip 3

5.2. Staffing

[Table 48] Test Staffing

Responsibilities	Team member
Test Plan	Jaesung Lee, Kyungmin Min
Test Case	Seongyun Lee, Jeongjae Park
Manual Testing	Yiqin Wei, Jongyoon Kim

5.3. Document History

[Table 49] Document History

Date	Version	Description	Writer
2022.11.06	0.0.1	Format documents	Kyungmin Min
2022.11.07	0.1.0	Addition of 1	Kyungmin Min
2022.11.07	0.1.1	Addition of 3	Jaesung Lee
2022.11.07	0.1.2	Addition of 2	Yiqin Wei
2022.11.07	0.1.3	Addition of 4, 5	Seongyun Lee

Date	Version	Description	Writer
2022.11.08	0.1.4	Addition of 2.1, 2.2	Yiqin Wei
2022.11.08	0.1.5	Addition of 3.1, 3.2	Jaesung Lee
2022.11.09	0.1.6	Addition of 3.3	Jaesung Lee
2022.11.09	0.1.7	Addition of 3.4 - 3.6	Jongyoon Kim
2022.11.09	0.1.8	Addition of 4.1, 4.2	Seongyun Lee
2022.11.10	0.1.9	Revision of 2	Yiqin Wei
2022.11.10	0.1.10	Addition of 2.3	Yiqin Wei
2022.11.11	0.1.11	Revision of 3.1 - 3.3	Jaesung Lee
2022.11.12	0.1.12	Revision of 3.4 - 3.6	Jongyoon Kim
2022.11.12	0.1.13	Addition of 3.7 - 3.9	Jeongjae Park
2022.11.13	0.1.14	Revision of 1.3, 1.5	Kyungmin Min
2022.11.13	1.0.0	Integration documents	Jongyoon Kim