성균:나누Re

성균관대 물품 대여 platform

소프트웨어공학개론

Team 13

손민규

송영욱

김영시

김예린

이현영

한승희

목 차

4 Software Interface Test

4.1 Login Test

4.1.1 Login approval time Test Case

4.1.2 Login approval time Test Case – Negative

4.1.3 Click time Test Case

4.1.4 Data transmit Test

4.1.4.1 Data request test

4.1.4.2 Data change test

1 Introduction

1.1 Purpose

본 문서는 성균관대 대여 플랫폼 나누 RE 서비스에 대한 STS(Software Test Specification)입니다. 나누 RE 는 성균관대학교 학생들이 물품 대여 및 나눔을 할 수 있도록 돕는 플랫폼이다. 따라서 다른 연관성이 없는 기능이 아닌 온전히 '물품 대여 및 나눔'에 집중된 플랫폼을 구현하고자 하였다. 사용자는 해당 플랫폼에서 물품 대여인 혹은 차용인을 구하고, 물품을 나눔을 받을 사람을 구할 수 있으며, 이를 통해 더욱 경제적이고 효율적인 삶을 추구할 수 있다. 또한 이 플랫폼의 기능은 대여 및 나눔과 관련이 없는 활동들을 억제하고 대여 및 나눔에 대한 약속을 올바르게 지키지 않은 사용자들에게 일부 대응할 수 있는 수단 또한 마련하였다.

본 문서에서는 테스트 방법과 툴을 소개한 후 유닛 테스트 방법과 인터페이스 테스트 방법을 별도로 소개한다. 각 테스트 방법은 의사 코드를 통해 작성되며, 이러한 테스트를 통해 서비스가 제대로 작동하는지 확인할 수 있다. 설계된 입력에 대해 예상 출력이 나오지 않는 경우 소프트웨어 수정이 필요하다. 우리는 회원가입, 로그인, 채팅, 게시물 등 세부 기능별로 테스트를 진행한다.

1.2 Scope

본 STS 문서에는 우선 나누 RE 에서 테스트를 진행하기 위해 필요한 정보들을 제공해 준다. 이 정보에는 테스트 실행에 필요한 명령어, 각 테스트의 목적과 체크리스트, 해당 테스트 시에 사용되는 입력 / 출력 값, 테스트의 작동 방식에 대한 수도 코드 등이 포함된다.

Unit Test 와 Interface Test 두가지의 테스트를 진행한다. 나누 RE 를 구성하는 시스템들을 기능이 가능한 최소한의 단위로 쪼개 각 기능에 대한 Unit Test 를 진행한다. Interface Test 는 시스템이 사용하고 있는 모든 외부적 요인들 (Response time, DB consistency) 에 대해 진행한다.

1.3 Definitions, Acronyms and Abbreviation

Table 1

용어	의미
채팅방	사용자들끼리 메시지를 주고받을 수 있는 공간
차단	게시글 / 채팅방을 보고 싶지 않은 유저에 대해 그 유저를 보지 않을 수 있도록 하는 기능
신고	부적절한 행위를 보이는 사용자에 대해 신고
ban	신고가 누적되거나 admin 에게 부적절한 행위가 적발된 유저는 플랫폼을 사용할 수 없도록 차단
대여 / 나눔	사용자들간 필요 물품을 대여 / 나눔
인증 메일	성균관대 학생임을 인증하기 위해 skku.edu / g.skku.edu 메일을 사용하여 인증
유저	공유경제 웹 플랫폼 나누 Re 의 사용자
관리자	나누 RE 의 작동 및 데이터베이스를 관리하고 제작하는 자
백엔드	서버와 데이터베이스 같이 유저에게 직접적으로 보이지 않는 어플리케이션의 부분
프론트엔드	어플리케이션에 보여지는 부분. 유저 인터페이스라고도 불린다.

1.4 References

Team7_STS

https://github.com/skkuse/2021spring_41class_team7

Test Plan Specification_Team 12

https://github.com/skkusal/SE_2022f

1.5 Overview

본 문서는 총 5 장으로 이루어져 있다.

두 번째 장에서는 개발하는 플랫폼에 대한 testing method 의 전체적인 구조에 대하여 설명한다. 나누 RE 의 전체 기능들을 세분화하고, 어떤 형식으로 testing 이 진행되는지, testing 과정에서 확인해야 하는 체크리스트는 무엇인지, test 진행에 대한 수도 코드 등을 서술한다. 그리고 이 테스팅을 하기위한 testing tool 을 서술한다.

세 번째 / 네 번째 장에서는 나누 RE 의 전체 기능들에 대해 test 를 실행한 결과를 상세하게 기술한다. 세 번째 장에서는 Unit test, 네 번째 장에서는 interface 를 다룬다. 각 기능들을 testing 하기 위해 사용된 Input 값, 해당 Input 에 대응하는 Output, 각 기능들에 사용된 test case 를 상세히 설명한다.

다섯 번째 장에서는 플랫폼에 대한 추가적인 정보들을 제공한다.

2 Approach

2.1 Test method

2.1.1 Software unit test methods

2.1.1.1 Register

- 모든 input 이 공란 없이 올바른 형식인지 확인
- 입력한 이메일로 '회원가입 인증 메일 [성균:나누 Re]' 메시지 전송되는지 확인
- 정상적으로 회원가입이 되어 DB 에 반영되는지 확인
- 이미 DB 상에 존재하는 중복된 ID, 이메일, 닉네임으로 회원가입 차단
- ID - 알파벳, 숫자, _로 구성/ length: 5~10

- 비밀번호 - 숫자로만 구성 불가/ length: 8 이상
- 이메일 - g.skku.edu 또는 @skku.edu
- 닉네임 - length: 10 이하

Table 2

```
# get user information(i)
if i
  if i == exist
      fail
  else
      i == valid
          Sending mail to skku.edu & get approval(a)
          if a == valid
              success Register()
          else
              fail
      else
          fail
  else
    fail
```

2.1.1.2 Login

- 등록된 사용자의 ID 와 비밀번호가 올바른 지 확인
- 운영자에게 ban 당한 계정으로 로그인을 시도할 때 차단

Table 3

```
# get user ID (i) , password(p)
if i and p == valid
  if user_ban
      fail
  else
      success
else
      fail
```

2.1.1.3 Logout

Logout 버튼이 눌리면 logout

Table 4

```
# logout button click listener (b)
if b
      success logout()
else
```

2.1.1.4 Profile

2.1.1.4.1 Change profile

- 닉네임 또는 위치(캠퍼스) 또는 프로필사진의 수정 사항을 올바르게 기입 시 수정한 회원정보가 DB 에 반영되는지 확인
- 각 항목에 올바르지 않은 내용을 입력 시, 정보 수정 요청 차단

Table 5

```
# get profile information(i)

if i == valid

        success

else

        fail
```

2.1.1.4.2 View profile

- 사용자의 프로필 정보(닉네임, 위치(캠퍼스), 프로필사진)를 열람할 수 있는지 확인
- 사용자의 차단 유저, 좋아요 한 게시글, 작성한 (대여원해요/대여합니다/나눔합니다) 게시글 리스트를 열람할 수 있는지 확인

Table 6

```
# profile button click listener (b)
```

```
if b

        success view_profile()

else
```

2.1.1.5 Notifications

2.1.1.5.1 Make notification

게시판에 글을 작성하면 글이 업로드 되는지 확인

Table 7

```
# get notification information(i) ,

if user == admin

    if i == valid

        success

else

        fail
```

2.1.1.5.2 Change notification

게시글의 수정된 제목, 내용이 올바르게 저장되는지 확인

Table 8

```
# get notification information(i)
```

```
if user == admin
    if i == valid
        success
else
    fail
```

2.1.1.5.3 View notification

사용자가 게시글을 눌렀을 때 공지사항 내용이 상세히 출력되는지 확인

Table 9

```
# notification button click listener (b)
if notification == exist
  if b
      success show_notification()
else
    fail
```

2.1.1.5.4 Delete notification

게시글의 삭제 버튼을 누를 때 게시글이 삭제되는지 확인

Table 10

```
# delete button click listener (b)
if user == admin
    if notification == exist
      if b
          success delete()
else
        fail
```

2.1.1.6 Post

2.1.1.6.1 Make post

- 인증된 사용자가 게시글을 등록했을 때 제목, 내용, 거래상태, 가격, 사진을 입력하면 등록되는지 확인
- 인증되지 않은 사용자가 게시글 등록 버튼을 눌렀을 때 차단
- 인증된 사용자가 게시글을 등록할 때, 제목, 내용, 거래상태, 가격, 사진을을 입력하지 않으면 등록 불가

Table 11

```
# get post information(i)
if user == valid
    if i == valid
            success
else
        fail
```

2.1.1.6.2 Change post

- 게시글을 작성한 사용자는 수정 가능
- 게시글을 작성하지 않은 사용자 또는 인증되지 않은 사용자는 수정 불가
- 게시글을 작성한 사용자가 수정한 제목, 내용, 거래상태, 가격, 사진이 올바르게 저장되는지 확인

Table 12

```
# get post information(i)
if user == valid
    if post == exist
        if i == valid
            success
else
    fail
```

2.1.1.6.3 View post

- 사용자가 메인 페이지 상단의 (대여원해요/대여합니다/나눔합니다) 게시판 버튼을 눌렀을 때 각 게시판의 게시글 목록을 열람할 수 있는지 확인
- 각 게시판에서 검색 내용에 대해 올바른 게시글 리스트가 나오는지 확인
- 게시판 페이지에서 게시글 들의 제목, 작성자 닉네임, 작성한 날짜 및 시간, 좋아요, 가격 정보가 나오는지 확인
- 게시판에 업로드 된 게시글 들이 작성한 날짜와 시간 순서대로 조회되는지 확인
- 게시글 제목을 눌렀을 때 상세 페이지로 들어가는지 확인
- 차단된 유저의 게시글이 보이지 않는지 확인

Table 13

```
if user == valid
```

```
            if post == exist
                if !block
                    success view_post()
            if click_like
                if click_like != exist
                    success post_like()
                else
                    fail
        else
            fail
```

2.1.1.6.4 Delete post

- 게시글을 작성한 사용자는 삭제 가능
- 게시글을 작성하지 않은 사용자 또는 인증되지 않은 사용자는 삭제 불가
- 게시글을 작성한 사용자가 삭제 버튼을 누를 때 게시글이 삭제되는지 확인

Table 14

```
        if user == valid
            if post == exist
                success
        else
            fail
```

2.1.1.7 Report

- 이미 신고한 유저는 신고 불가

- 10 번 초과로 신고 받은 유저는 ban

Table 15

```
# get Report information(i)
if user == valid
    if report != exist
        if i == valid
            success report()
        if report > 10
            success ban()
else
        fail
```

2.1.1.8 Block

2.1.1.8.1 User block

유저 차단을 하면 해당 유저의 게시글을 열람 불가한지 확인

Table 16

```
# get   information(i)
if user == valid
    if block != exist
            success block()
else
        fail
```

2.1.1.8.2 User unblock

유저 차단을 해제하면 해당 유저의 게시글을 열람 가능한지 확인

Table 17

```
# get   information(i)
if user == valid
    if block == exist
        success unblock()
else
    fail
```

2.1.1.9 Chatting

2.1.1.9.1 Make chat room

- 게시글을 올린 유저에게 정상적으로 채팅방이 생성되는지 확인
- 채팅방을 생성하거나 채팅방에 초대받는 유저가 인증되지 않은 사용자면 불가
- 채팅 하고자 하는 유저와 이미 채팅방이 생성되어 있으면 불가
- 채팅 하고자 하는 유저에게 차단되었다면 생성 불가

Table 18

```
if user == valid
    if chatroom != exist
        if !block
            success make_chat()
else
    fail
```

2.1.1.9.2 View chat room list

유저가 기존에 생성한 채팅방들이 누락없이 모두 리스트에 보이는지 확인

Table 19

```
if user == valid
    if chatroom == exist
        success view_chatroom_list()
else
    fail
```

2.1.1.9.3 Delete chat room

Table 20

```
# get post information(i)
if user == valid
    if chatroom == exist
        success delete_chat()
else
    fail
```

2.1.1.9.4 Enter chat room

채팅방 입장 시 기존에 주고 받은 대화 내용이 누락없이 모두 보이는지 확인

Table 21

```
if user == valid
```

```
            if chatroom == exist
                if message == exist
                    success view_chat()
        else
            fail
```

2.1.1.9.5 Sending message

채팅방에 일반 텍스트를 작성하고 메세지를 발신할 때 지정한 대상에게 정상적으로 해당 메세지가 출력되는지 확인

Table 22

```
        # get message (m)
        if user == valid
            if chatroom == exist
                if !block
                    if m == valid
                        success make_chat()
        else
            fail
```

2.1.2 Software interface test methods

2.1.2.1 Login Test

2.1.2.1.1 Login approval time Test Case

Login 시도를 했을 때 해당 ID, Password에 대해 DB와 로그인 승인에 대한 response를 주고받는 시간이 0.1초 이내여야 한다.

2.1.2.1.2 Login approval time Test Case - Negative

Login 시도를 했을 때 해당 ID, Password에 대해 DB와 로그인 거부에 대한

response를 주고받는 시간이 0.1초 이내여야 한다.

2.1.2.2 Logout Test

Logout 시도를 했을 때 해당 User 가 사이트에서 logout 을 하는 시간이 1 초 이내여야 한다.

2.1.2.3 Data transmit Test

DB 와 data 를 주고받을 때 해당 data 들이 consistency 를 유지해야 한다

2.1.2.3.1 Data request Test

Table 23

```
# get data request (m)
if user == valid
    if m == valid
        if data == exist and valid
            success request()
else
        fail
```

2.1.2.3.2 Data change Test

Table 24

```
# get data change request (m)
if user == valid
    if m == valid
        if data == exist and valid
            success data_change()
else
        fail
```

2.2 Test Assumptions

2.2.1 Key Assumption

- 각 테스트 전에 시스템에 필요한 데이터가 제공된다.
- 각 테스트는 최대 5 회를 수행하며, 3 회의 error rate 이 높다면 2 회의 테스팅을 추가로 수행한다.

2.2.2 General Assumptions

- 테스트는 Windows 11 환경에서 chrome 을 통해 수행된다.
- 테스트는 각 팀원의 인터넷 환경과 VPN 연결을 통해 진행된다.
- Error 가 발견되면 테스트를 중지하고 Error 를 수정한 후 테스트를 다시 수행한다.
- Testing 인원은 해당 test 에 대한 정보를 숙지하고 testing 을 진행한다.
- 팀장의 검토가 끝난 후 검토 결과에 따라 Testing 이 종료된다.

2.3 Testing Tool

2.3.1 Test Environment

Table 25

View	node	14.18.0
	react	18.2.0
	axios	1.4.0
	styled-components	5.3.10
Model	Python	3.10
	FastAPI	0.95.1
	SQLAlcehmy	2.0.12
	Websockets	11.0.3

2.3.2 Test Method

Table 26

Unit Test Method		
React	Jest	29.5
	React Testing Library	14.0.0
Python	Pytest	7.3.1

3 Unit test

3.1 Register

케이스 이름	테스트 시나리오	입력 및 사전조건	기대 출력
회원가입	모든 입력이 공란 없이 올바른 형식으로 입력되었을 때 DB 에 정상적으로 반영되는지 확인	DB 와 중복되지 않는 5~10 자의 ID 와 8 자 이상의 비밀번호, @g.skku.edu 나 skku.edu 의 이메일, 10 자 이하의 닉네임	이메일로 인증 링크가 전송되었다는 안내 화면 출력
	일부 입력이 공란인 상태로 회원 가입 시도 시 회원가입이 차단되는지 확인	일부 입력 공란	"모든 항목을 입력하십시오"라는 팝업창 표시
	이미 DB 에 존재하는 중복된 ID 로 회원가입 시도 시 회원가입 시도가 차단되는지 확인	중복된 ID 나머지는 맨 위 케이스와 동일	"Username Exist"라는 팝업 창 표시
	이미 DB 에 존재하는 중복된 닉네임으로 회원가입 시도 시 회원가입 시도가 차단되는지 확인	중복된 닉네임 나머지는 맨 위 케이스와 동일	"Nickname Exist"라는 팝업 창 표시

	비밀번호와 재입력 비밀번호가 일치하지 않은 상태로 회원가입 시도 시 회원가입 시도가 차단되는지 확인	비밀번호와 일치하지 않는 재입력 비밀번호	"비밀번호가 일치하지 않습니다"라는 팝업 창 표시
	올바르지 않은 형식의 ID 로 회원가입 시도 시 회원가입 시도가 차단되는지 확인	길이가 5 미만 또는 10 초과이며 알파벳, 숫자, _ 이외의 문자를 사용한 ID	"아이디는 5-20 자 이내의 영문, 숫자와 언더바만 사용 가능합니다"라는 팝업 창 표시
	올바르지 않은 형식의 비밀번호로 회원가입 시도 시 회원가입 시도가 차단되는지 확인	길이가 8 미만이거나 숫자로만 이루어진 비밀번호	"비밀번호는 최소 8 자 이상이어야 합니다." 혹은 "숫자로만 이루어진 비밀번호는 사용할 수 없습니다"라는 팝업 창 표시
	올바르지 않은 형식의 닉네임으로 회원가입 시도 시 회원가입 시도가 차단되는지 확인	10 자 초과의 닉네임	"nickname must be less than 10"라는 팝업 창 표시
	올바르지 않은 형식의 이메일로 회원가입 시도 시 회원가입 시도가 차단되는지 확인	@g.skku.edu. 나 @skku.edu 이외 형식의 이메일	"학교 이메일을 사용해주세요"라는 팝업 창 표시

3.2 Login

케이스 이름	테스트 시나리오	입력 및 사전조건	기대출력
로그인	모든 입력이 공란 없이 DB 에 존재하는 올바른 입력일 때	DB 에 존재하는 ID 와 비밀번호	정상적으로 로그인 되며 메인 화면으로 이동

	정상적으로 로그인이 되는 지 확인		
	DB 에 존재하는 ID 이나, 비밀번호가 옳지 않을 때 로그인이 차단되는지 확인	DB 에 존재하는 ID 와 등록된 비밀번호와는 다른 비밀번호	"Incorrect password"라는 팝업 창 표시
	DB 에 존재하지 않는 ID 로 로그인을 시도할 때 로그인이 차단되는지 확인	DB 에 존재하지 않는 ID	"Any user with that ID"라는 팝업 창 표시
	DB 에 존재하는 ID 와 올바른 비밀번호이지만 이메일 인증이 완료되지 않은 계정일 때 로그인이 차단되는지 확인	아직 이메일 인증을 마치지 않은 DB 에 존재하는 ID 와 등록된 비밀번호	"Verify email first!"라는 팝업 창 표시
	운영자에 의해 ban 된 계정으로 로그인을 시도할 때 차단되는지 확인	운영자에 의해 ban 된 DB 에 존재하는 ID 와 등록된 비밀번호	"User is Banned!"라는 팝업 창 표시

3.3 Logout

케이스 이름	테스트 시나리오	입력 및 사전조건	기대출력
로그아웃	로그인한 사용자의 세션으로부터 로그아웃을 시도, 정상적인 로그아웃이 완료되었는지 확인	로그인한 메인 페이지에서 로그아웃 버튼 입력	로그아웃이 되어 로그인 되어있지 않은 메인 페이지로 이동

3.4 Profile

3.4.1 Change profile

케이스 이름	테스트 시나리오	입력 및 사전조건	기대출력

프로필 수정	프로필 수정사항을 정확하게 입력할 시에 DB 에 수정사항이 잘 반영되는지 확인	10 자 이하의 닉네임을 입력하고 캠퍼스, 프로필 사진을 올바르게 등록함	DB 에 바뀌는 프로필 정보가 정확하게 입력됨.
	각 항목에 올바르지 않은 내용을 입력할 시, 정보 수정이 차단되는지 확인	닉네임을 10 자 초과로 등록, 캠퍼스 미 입력	각 입력 오류에 해당하는 오류 메시지 출력

3.4.2 View profile

케이스 이름	테스트 시나리오	입력 및 사전조건	기대출력
프로필 확인	본인의 프로필을 잘 확인할 수 있는지 확인	페이지 우측 상단의 '프로필' 탭 클릭	프로필 페이지로 넘어가며 유저 차단 목록, 좋아요 목록, 각 게시판 글 쓴 이력 확인 가능.
	타인의 프로필을 확인할 때 차단 유저 목록, 프로필 수정 항목이 안보이는지 확인	타인의 프로필 확인	차단목록, 프로필 수정 버튼을 제외한 나머지 프로필 목록을 확인할 수 있음

3.5 Notifications

케이스 이름	테스트 시나리오	입력 및 사전조건	기대출력
공지사항 생성	관리자가 공지사항을 작성함	관리자가 공지사항 글을 작성하고 POST 를 시도한다	공지사항이 메인 화면 상단에 노출되며 알맞게 작성된다.
공지사항 수정	관리자가 공지사항을 수정함	관리자가 공지사항에 수정사항을 가함	공지사항이 정상적으로 수정됨
공지사항 삭제	관리자가 공지사항을 삭제함	관리자가 공지사항을 삭제함	공지사항이 정상적으로 삭제됨

공지사항 열람	사용자가 공지사항을 열람함	사용자가 공지사항을 클릭해 열람함	공지사항 열람을 성공함
공지사항 생성 실패	관리자가 아닌 사용자가 공지사항 생성을 시도했을 때 요청이 거부되는지 확인	관리자가 아닌 사용자가 공지사항 생성을 시도	요청이 거부됨
공지사항 수정 실패	관리자가 아닌 사용자가 공지사항 수정을 시도했을 때 요청이 거부되는지 확인	관리자가 아닌 사용자가 공지사항 수정을 시도	요청이 거부됨
공지사항 삭제 실패	관리자가 아닌 사용자가 공지사항 삭제를 시도했을 때 요청이 거부되는지 확인	관리자가 아닌 사용자가 공지사항 삭제를 시도	요청이 거부됨

3.6 Post

케이스 이름	테스트 시나리오	입력 및 사전조건	기대출력
게시글 열람	사용자가 게시글을 열람할 수 있는지 확인	사용자가 읽고 싶은 게시글을 클릭한다	게시글의 내용이 상세하게 적힌 페이지로 넘어간다
게시글 등록	사용자가 게시글을 등록할 수 있는지 확인	사용자가 원하는 홍보 내용을 작성하여 게시글을 등록한다	게시글의 등록에 성공한다
게시글 수정	사용자가 게시글을 수정할 수 있는지 확인	사용자가 자신이 작성한 게시글을 수정 버튼을 통해 수정한다	게시글 수정 페이지로 접속되며 수정이 가능하다
타인 게시글 수정 실패	사용자가 본인이 작성하지 않은 게시글을 수정할 수 없는지 확인	사용자가 타인의 게시글에서 수정 버튼을 찾는다	수정 버튼이 보이지 않아 수정이 불가능하다

게시글 삭제	사용자가 게시글을 삭제할 수 있는지 확인	사용자가 자신이 작성한 게시 글에서 삭제 버튼을 누른다	게시글이 정상적으로 삭제된다
타인 게시글 삭제 실패	사용자가 본인이 작성하지 않은 게시글을 삭제할 수 없는지 확인	사용자가 타인의 게시 글에서 삭제 버튼을 찾는다	삭제 버튼이 보이지 않아 삭제가 불가능하다.

3.7 Report

케이스 이름	테스트 시나리오	입력 및 사전조건	기대출력
신고	사용자가 타인의 게시글을 신고할 수 있는지 확인	타인의 게시 글에서 신고 버튼을 통해 게시글을 신고한다	게시글이 정상적으로 신고되며 알림이 뜬다
신고 불가능-자기자신	사용자가 자신을 신고할 수 없는지 확인	자신의 게시 글에서 신고 버튼을 통해 게시글을 신고한다	자신의 게시 글에서는 신고 버튼을 찾을 수 없어 불가능하다
신고 불가능-재 신고	사용자가 한 게시글을 여러 번 신고할 수 없는지 확인	이미 신고한 게시 글에서 신고 버튼을 또 누른다	이미 신고한 게시글이라는 알림이 뜨며 요청이 거부된다
신고 누적 10 회	사용자가 누적 10 회 신고 당하면 자동으로 block 처리되는지 확인	10 명 이상의 사용자가 한 사용자를 신고한다	시스템에 의해 사용자가 자동으로 ban 된다.

3.8 Block

케이스 이름	테스트 시나리오	입력 및 사전조건	기대출력
유저 차단	사용자가 다른 사용자를 차단하면, 그 사용자의	사용자가 다른 사용자를 차단한다	차단한 사용자의 게시글이 보이지 않으며, 프로필의

	게시글이 보이지 않게 되는지 확인		'차단 목록'에 추가됨
유저 차단 해제	사용자가 차단한 유저의 차단을 해제하면 그 유저의 게시글이 다시 보이게 되는지 확인	사용자가 차단한 사용자의 차단을 해제한다	차단이 풀린 사용자의 게시글이 다시 보이게 되고, 프로필의 '차단 목록'에서 사라짐

3.9 Chatting

케이스 이름	테스트 시나리오	입력 및 사전조건	기대출력
채팅방 생성	게시글을 올린 유저와 소통을 위해 채팅방 생성이 잘 되는지 확인	게시 글에서 버튼을 눌러 채팅방을 생성한다	채팅방이 정상적으로 생성되며 채팅 리스트에 추가된다.
채팅방 리스트	지금까지 만든 채팅방들이 채팅방 리스트에 정상적으로 잘 등록되어 있는지 확인	홈페이지 우측 상단의 '채팅해요' 버튼 입력	채팅방 리스트들과 닉네임, 내용, 보낸 시간 등이 잘 출력된다.
채팅방 입장	채팅방을 눌러 직접 채팅을 하러 입장이 잘 되는지 확인.	채팅방 리스트의 채팅방 중 한 개를 클릭하여 입장	구체적인 채팅 내역이 잘 드러난다.
메시지 발신	채팅방에서 메시지가 발신되는지 확인	사용자가 메시지를 입력하고 보내기 버튼을 눌러 보낸다.	상대방에게 메시지가 잘 가고, 채팅방에 내역이 남는다.
메시지 수신	채팅방에서 메시지가 잘 수신되는지 확인	상대방 사용자가 메시지를 입력하고 보내기 버튼을 눌러 보낸다.	사용자에게 메시지가 잘 도착하고, 채팅방에 내역이 남는다.

채팅방 생성 실패- 이미 있는 채팅방	이미 한 사용자와 채팅방이 존재하는 상황에서 다시 채팅방 생성을 시도할 시 요청이 거부되는지 확인	사용자와 채팅방이 맺어진 상태에서 다시 채팅하기 버튼을 입력	채팅방이 이미 있다는 알림을 띄움

4 Software Interface Test

4.1 Login Test

4.1.1 Login approval time Test Case

Table 27

Object	사용자가 아이디와 비밀번호를 입력하여 로그인을 요청할 때 얼마나 빠르게 로그인을 승인할 수 있는지를 테스트한다.
Input	ID , Password
output	0.1 초 안에 로그인 승인

4.1.2 Login approval time Test Case – Negative

Table 28

Object	사용자가 아이디와 비밀번호를 입력하여 로그인을 요청할 때 얼마나 빠르게 로그인을 거부할 수 있는지를 테스트한다.
Input	ID , Password
output	0.1 초 안에 로그인 거부

4.1.3 Click time Test Case

Table 29

Object	사용자가 버튼을 눌러 어떠한 기능을 요청할 때 얼마나 빠르게 기능들을 실행할 수 있는지 확인
Input	Button click
output	1초안에 해당 요청 처리

4.1.4 Data transmit Test

DB와 data를 주고받을 때 해당 data들이 consistency를 유지해야 한다.

4.1.4.1 Data request Test

Table 30

Object	사용자가 어떠한 data를 요청할 때 얼마나 빠르고 정확하게 database와 정보를 주고 받을 수 있는지 확인
Input	Data_request(d)
output	Print(d)

4.1.4.2 Data change Test

Table 31

Object	사용자가 어떠한 data에 대한 수정을 요청할 때 얼마나 빠르고 정확하게 database와 정보를 주고받을 수 있는지 확인
Input	Data(d)
output	Print("${d} data changed")