전국 맛집 후기 공유 사이트

**맛** 있을 **지도**

소프트웨어 테스트 보고서

팀장: 20153063 윤 혁

팀원: 20153096 허원영

20173046 김소원

20193065 손영훈

# 문서 개요

이 문서는 개발 과정에서 시스템이 사전에 작성된 요구사항에 충족하여 개발되었는지를 확인하고 소프트웨어의 결함을 발견하기위한 것이며 소프트웨어의 여러 측면을 테스트하여 결과 보고서를 작성한것이다.

## 구성 항목

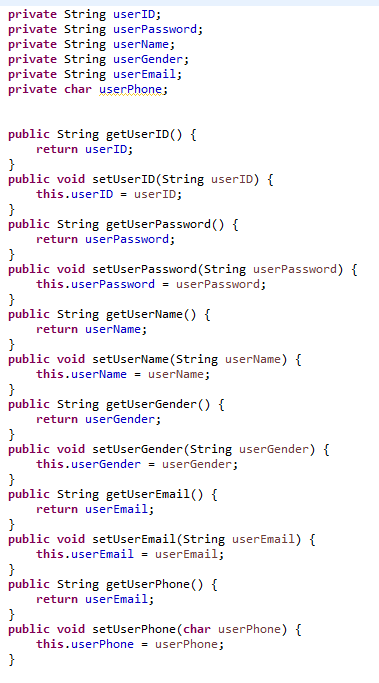
본 문서의 주요 구성 항목은 다음과 같다.

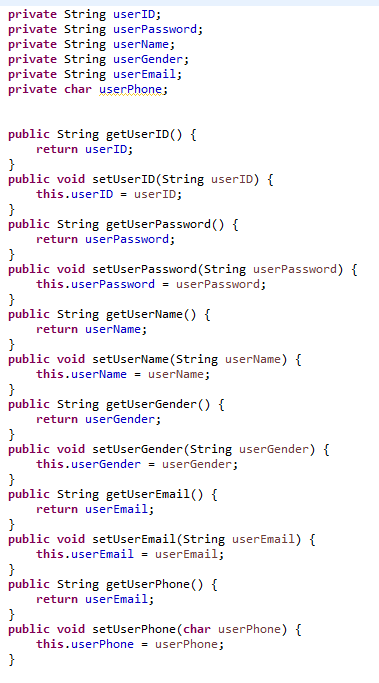
* 테스트 개요: 맛있을지도 시스템에 대해서 간략하게 소개하고 테스트의 종류 및 목적, 테스트 시나리오를 명시한다.
* Unit Testing: 실제 구현한 클래스 다이어그램을 명시하여 실제 구현한 클래스와 디자인했던 클래스의 차이를 확인하고 소스코드와 결과 화면을 명시하여 예상한 입력에 대해 어떻게 동작하는지 확인한다.
* System Testing: 소프트웨어 요구사항 명세서의 사용자의 요구사항으로부터 데이터 모델 및 기능, 품질들에 대한 테스트를 진행하고 결과를 명시한다
* References: 보고서를 작성하는데 활용한 참고자료를 첨부한다.

# Unit Testing

## User

|  |  |
| --- | --- |
| DRW00004c586d13 |  |
| <계획한 user클래스> | <실제로 사용한 user클래스> |

* user클래스 코드



## UserDAO

|  |  |
| --- | --- |
| DRW00004c586d13 |  |
| <계획했던 UserDAO클래스> | <실제로 구현된 UserDAO 클래스> |

개발에서 유저정보 수정 기능을 구현하지 못하여 Update() 함수는 제외하였다.

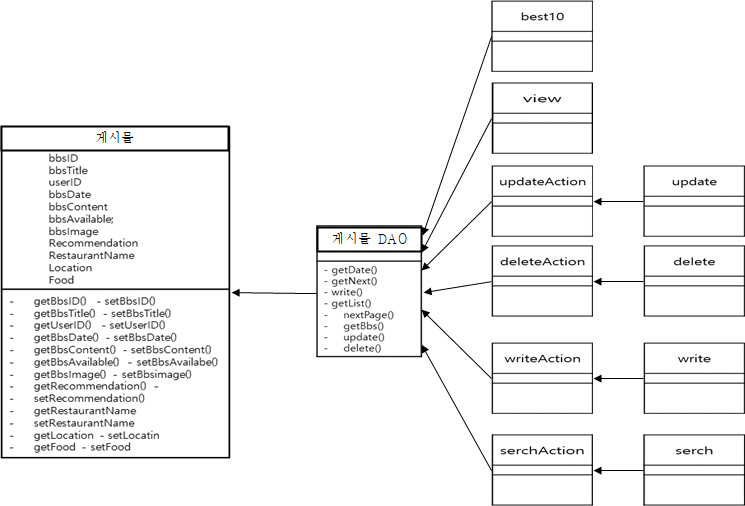
* UserDAO 클래스 코드

## Bbs

Bbs는 게시물, 게시판과 종합적으로 관련된 클래스이다.

이전 소프트웨어 설계서에서 첨부한 Bbs클래스의 캡쳐가 user클래스의 캡쳐와 동일한 것으로 잘못 첨부되어 있어 수정이 필요하다

.

기존에 계획한 게시물 클래스의 전체적인 그림은 아래와 같다.

|  |  |
| --- | --- |
|  |  |
| <계획했던 Bbs 클래스> | <실제로 구현된 Bbs 클래스> |

이미지 업로드 기능을 구현하지 못하여 관련 메소드들은 삭제하였다.

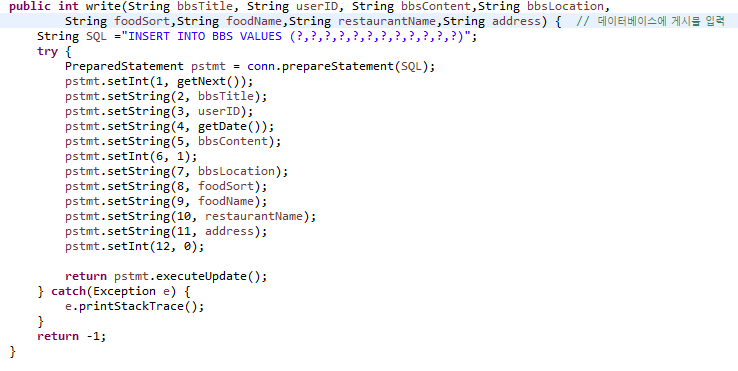
## BbsDAO

|  |  |
| --- | --- |
|  |  |
| <계획했던 BbsDAO 클래스> | <실제로 구현된 BbsDAO 클래스> |

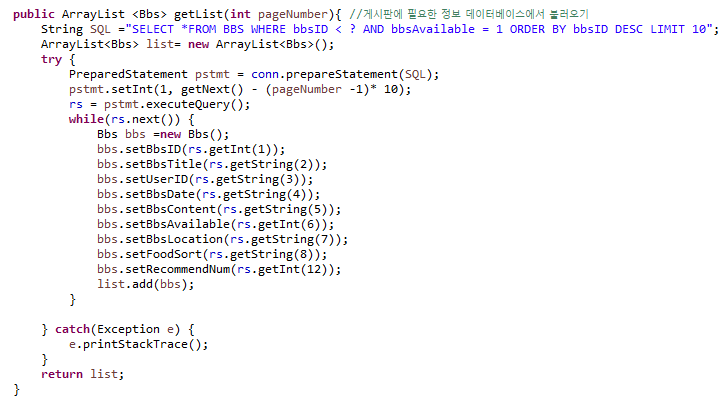
BbsDAO 클래스는 기존 계획했던 클래스에서 BbsDAO(), getBest(), recom(int bbsID)가 추가 되었다.

각 기능별 코드는 아래와 같다.

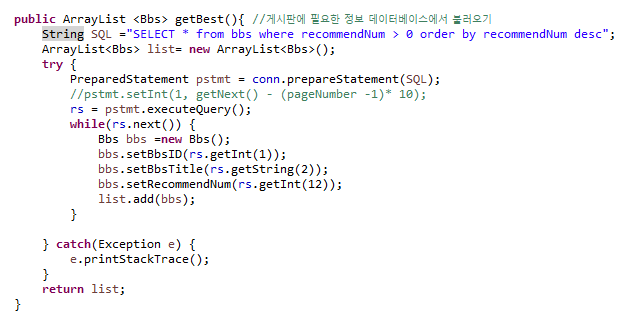
* 게시글 작성



* 게시글 목록출력



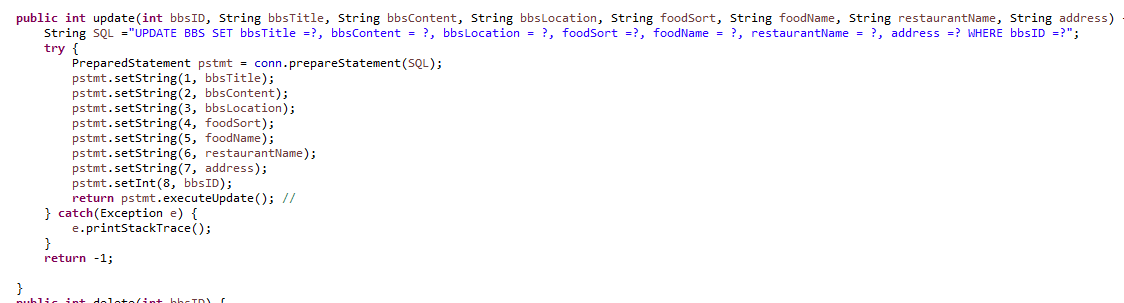
* Best10 출력



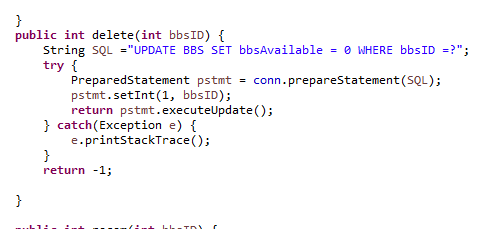
* 게시글 출력



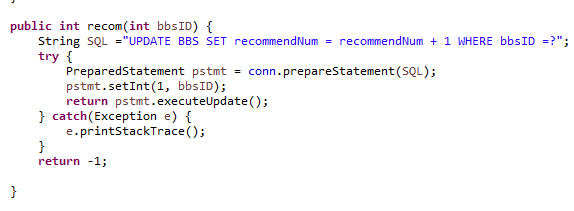
* 게시글 수정



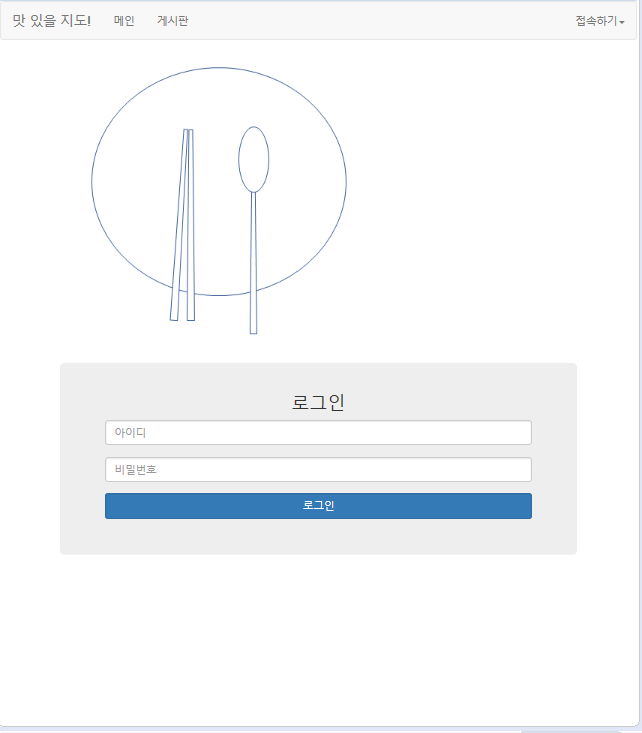
* 게시글 삭제



* 게시글 추천



## 로그인



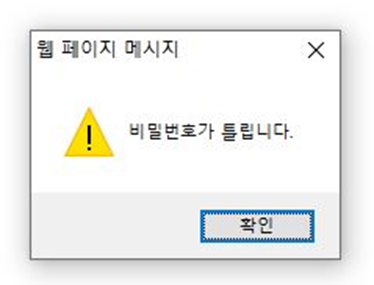
### 로그인 성공

올바르게 로그인을 시도한 경우 UserDAO클래스의 login 메소드에서 return값으로 1을 반환하게 되며 로그인에 성공한다.

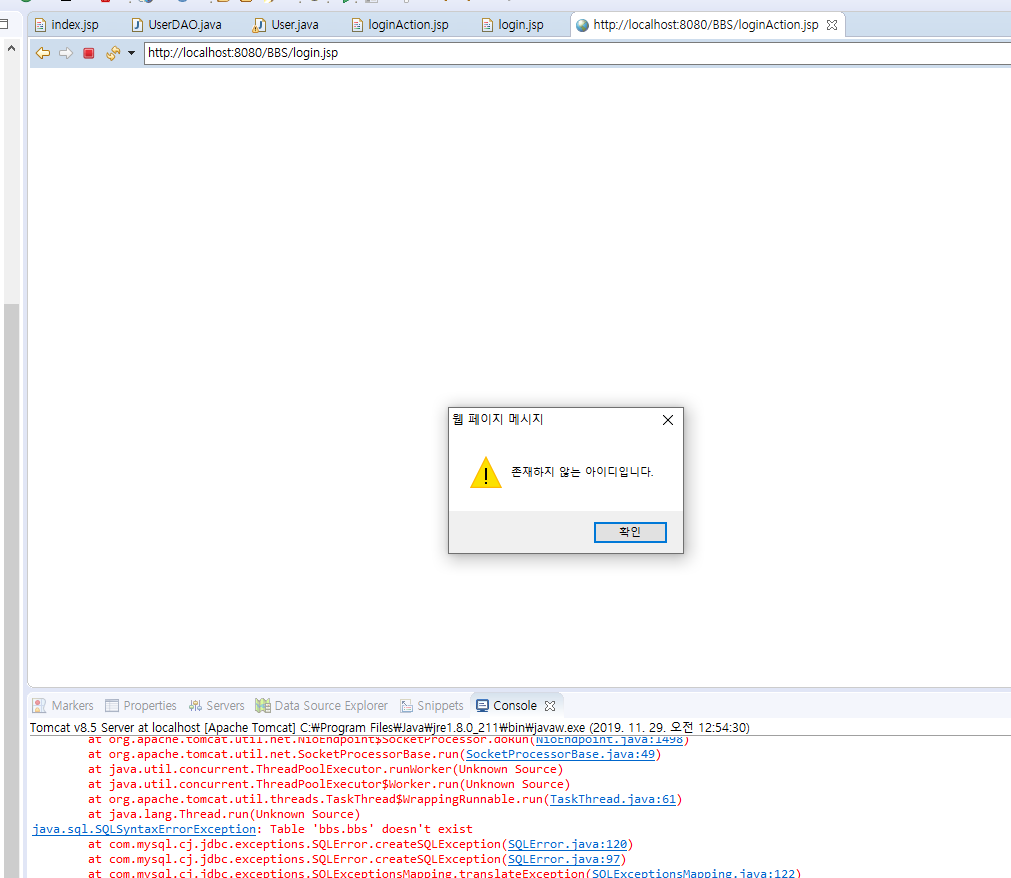
로그인에 성공한 사용자는 main화면으로 돌아가게된다.



### 로그인 실패

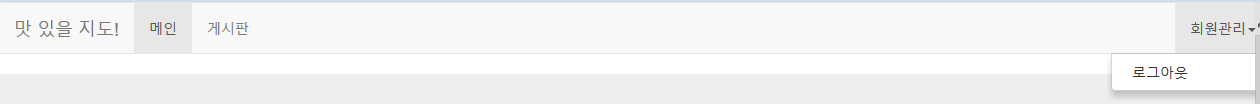
* 로그인 시도에서 잘못된 비밀번호를 입력한 경우 ‘비밀번호가 틀립니다’라는 경고문을 출력하고 UserDAO클래스의 login함수에서 return값으로 0을 반환하고 로그인 페이지로 다시 돌아간다.



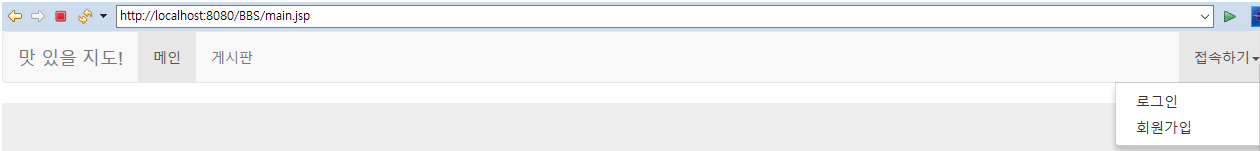
* 로그인 시도에서 잘못된 아이디를 입력한 경구 ‘존재하지 않는 아이디입니다.’라는경고문을 출력하고 UserDAO클래스의 login함수에서 return값으로 -1값을 반환하고 로그인 페이지로 다시 돌아간다.
* mySQL과의 데이터베이스 연동 오류가 발생하면 ‘데이터베이스 오류가 발생했습니다’라는 경고문을 출력하고 UserDAO클래스의 login함수에서 return값으로 -2를 반환하고 로그인 화면으로 돌아간다.



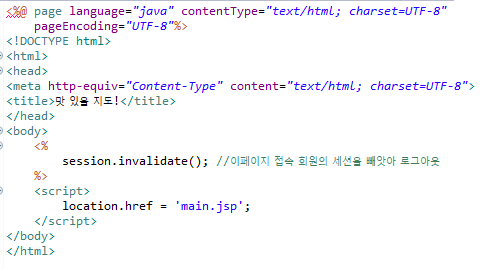
## 로그아웃



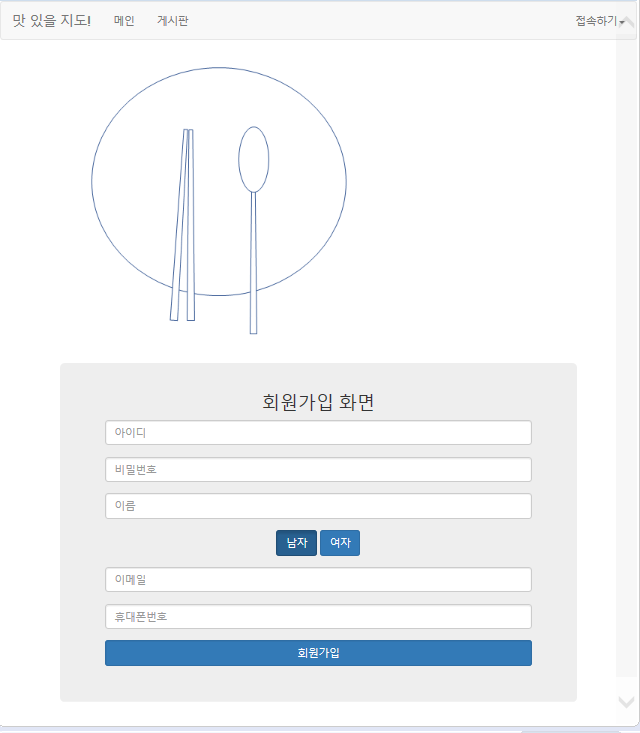
로그인 되어있는 상태에서 사용자가 회원관리에서 로그아웃버튼을 클릭하면 logoutAction클래스를 불러와 아래와 같이 로그아웃된다.



* logoutAction 코드



## 회원가입



### 회원가입 성공

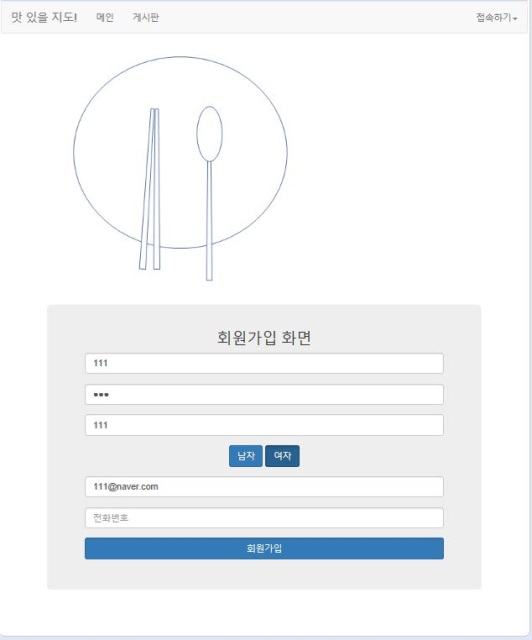
올바르게 회원가입을 시도한 경우 UserDAO클래스의 join함수에서 유저 정보를 DB로 저장하게되고 return값으로 -1이외의 값을 반환한다.

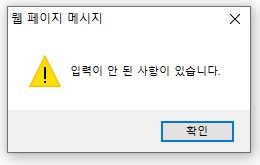
회원가입에 성공한 사용자는 main화면으로 돌아간다.

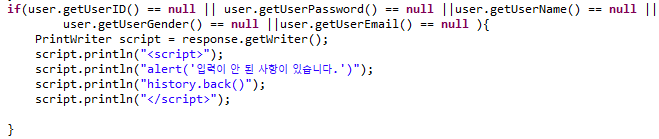


### 회원가입 실패

* 필수 입력사항중 하나라도 입력하지 않았을 경우 ‘입력이 안 된 사항이 있습니다’라는 경고메시지를 출력하고 회원가입 화면으로 돌아간다.

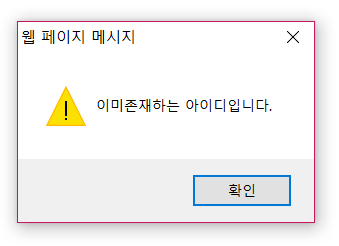






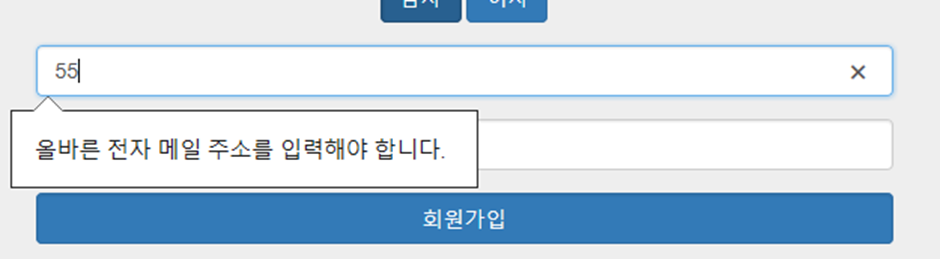
* 필수 입력사항을 모두 입력하였으나 userID가 이미 DB에 저장되어있는 항목이라면 ‘이미 존재하는 아이디입니다’라는 경고메시지를 출력하고 회원가입 화면으로돌아간다.





* 올바르지 않은 이메일 형식을 사용하여 회원가입을 시도한 경우

‘올바른 전자 메일 주소를 입력해야합니다’라는 경고메세지 출력하고 회원가입을 진행하던 화면으로 돌아간다.

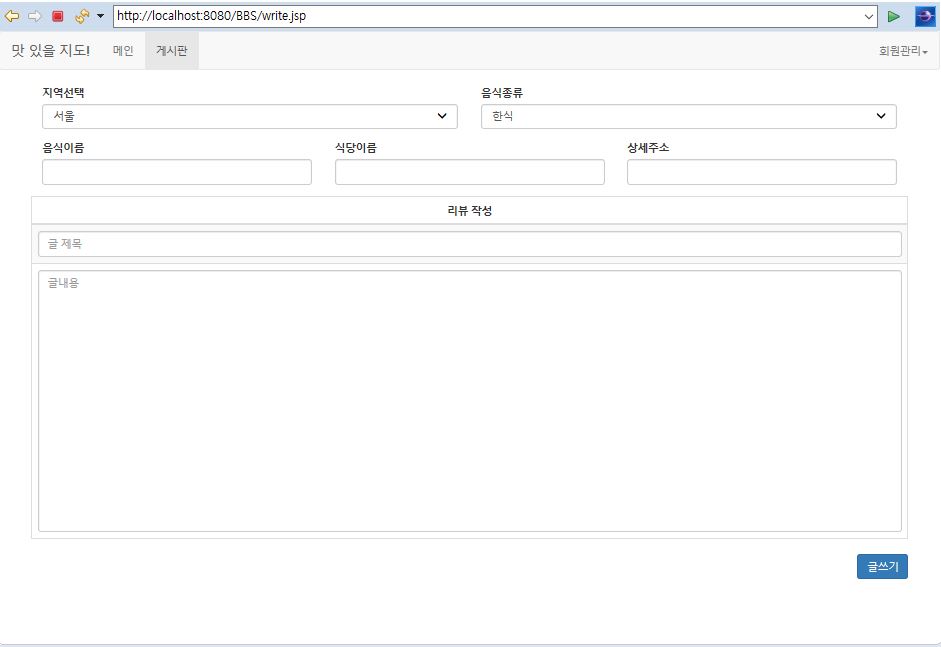


## 게시물

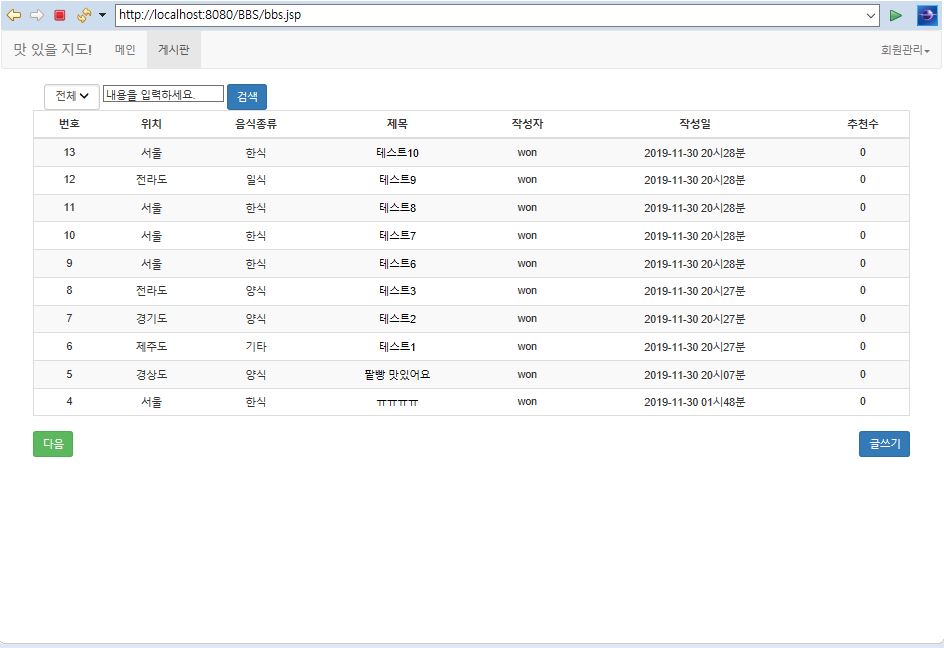
### 게시물 작성

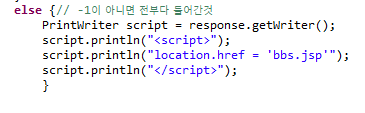
게시물 작성은 로그인된 사용자만 가능하다.

게시물 작성 버튼을 클릭하면 아래와 같은 화면이 나타난다.

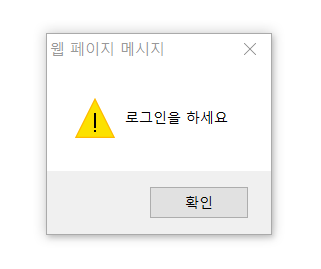
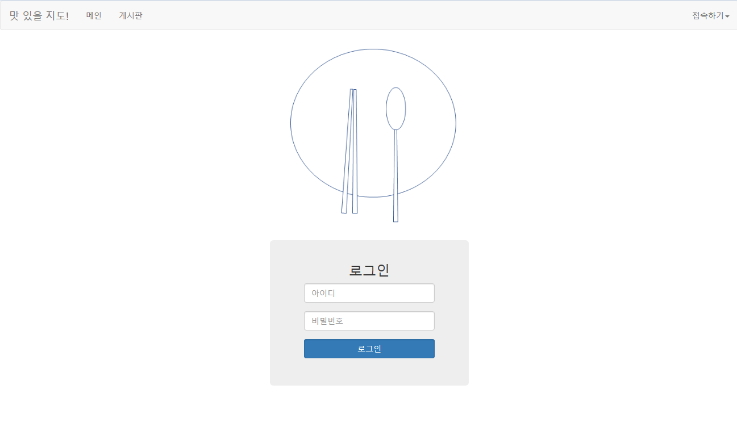


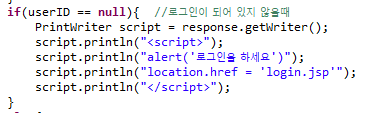
* 로그인 되어있는 사용자가 미기입항목 없이 글쓰기를 완료했을 경우 사용자가 작성한 글이 게시글 목록에 즉시 반영된다.



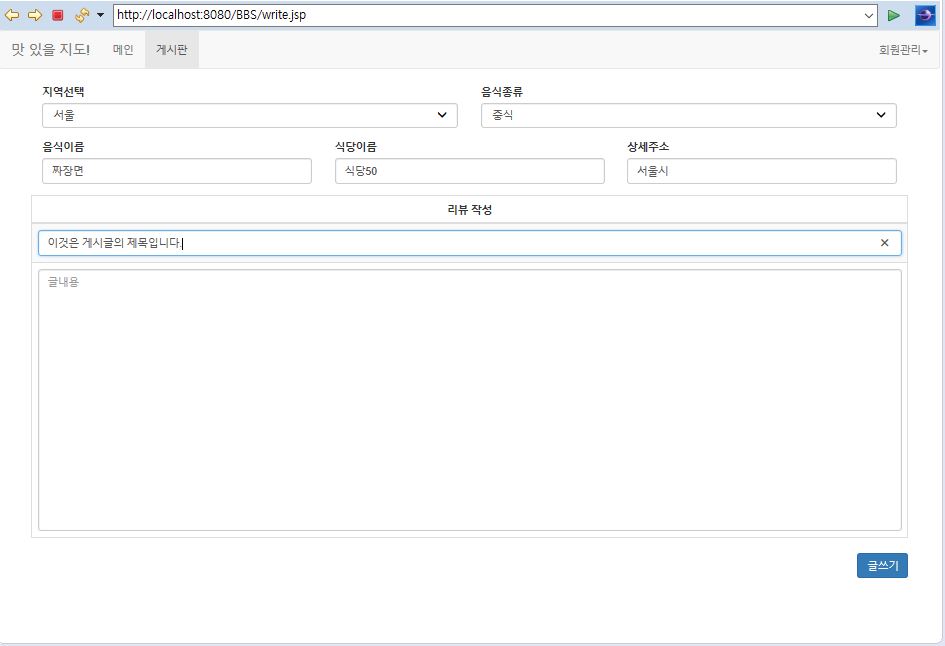


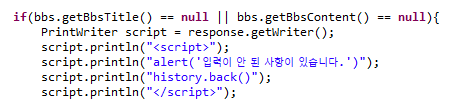
* 로그인하지 않은 사용자가 게시글 작성을 시도하면 다음과 같은 경고메시지가 나타나며 로그인 화면으로 이동한다.

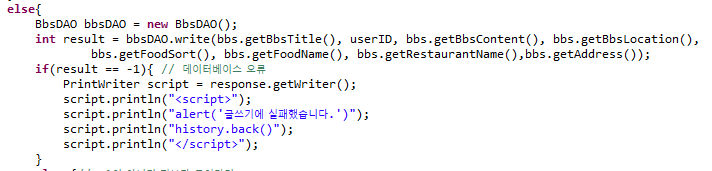
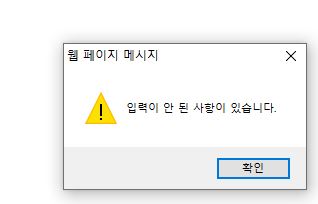




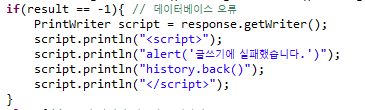
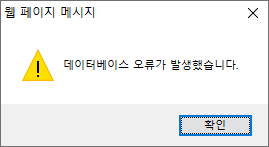
* 게시글 작성에 미기입항목이 있는경우







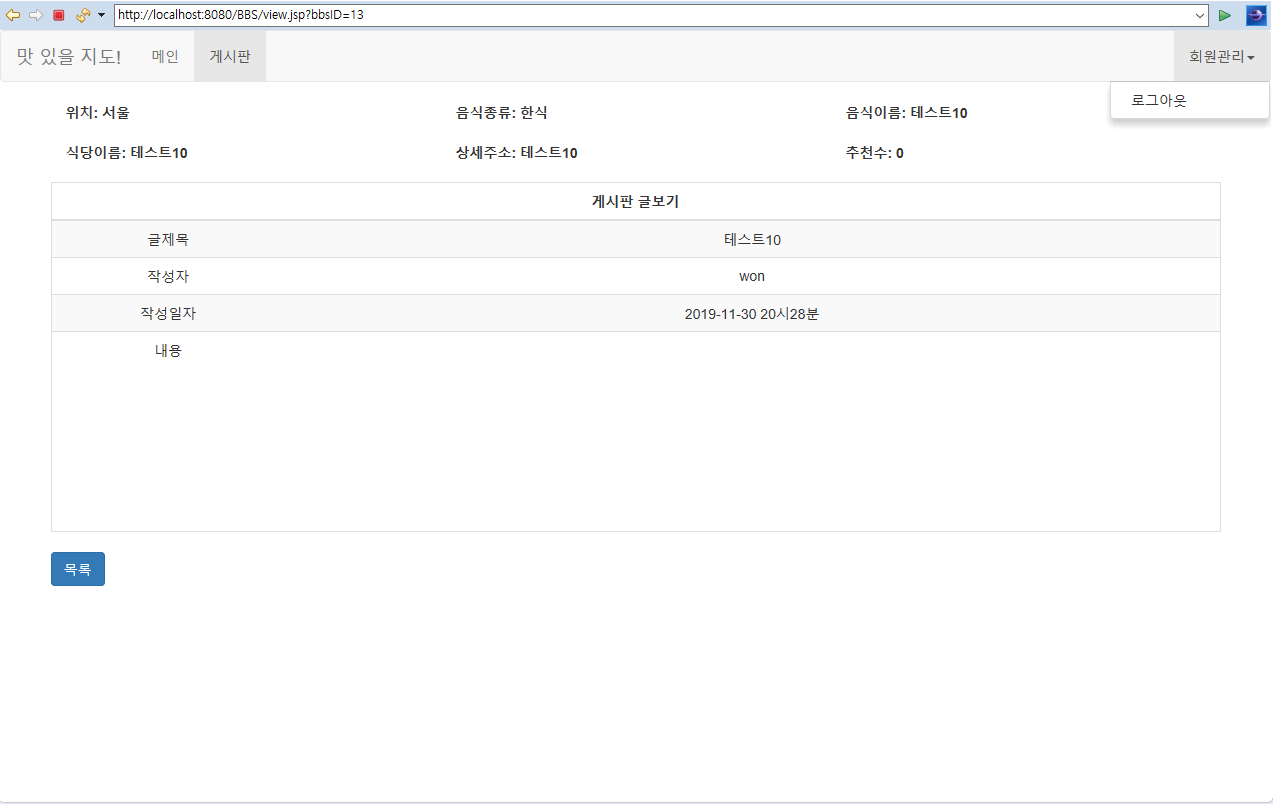
* 데이터베이스 오류가 발생한 경우



### 게시물 수정/삭제/추천

* 게시물 수정/삭제는 작성자 본인만 가능하다.

권한이 없는 사용자에게는 게시물 목록으로 돌아가는 버튼만 화면에 표시되고 게시물 수정/삭제 버튼은 화면에 표시되지 않는다.



* 게시글 수정

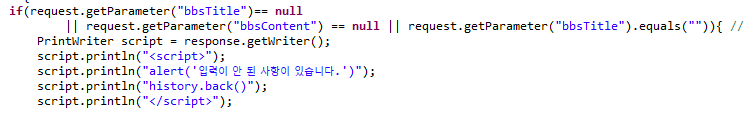
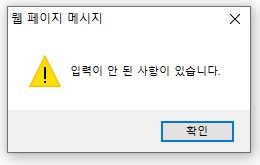
로그인 정보를 통해 게시글 작성자의 본인여부가 확인되면 사용자는 게시글

수정이 가능하다.

작성하지 않은 항목이 존재하는 상태로 글수정을 누르면 ‘입력이 안 된 사항이 있습니다’라는 경고메시지를 출력하고 게시글 수정 화면으로 돌아간다.



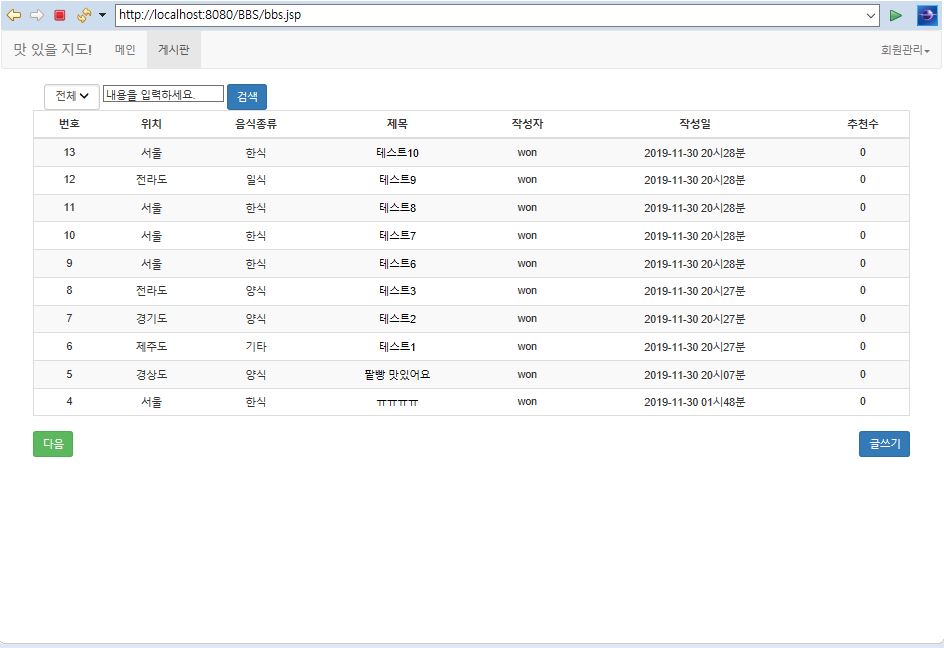




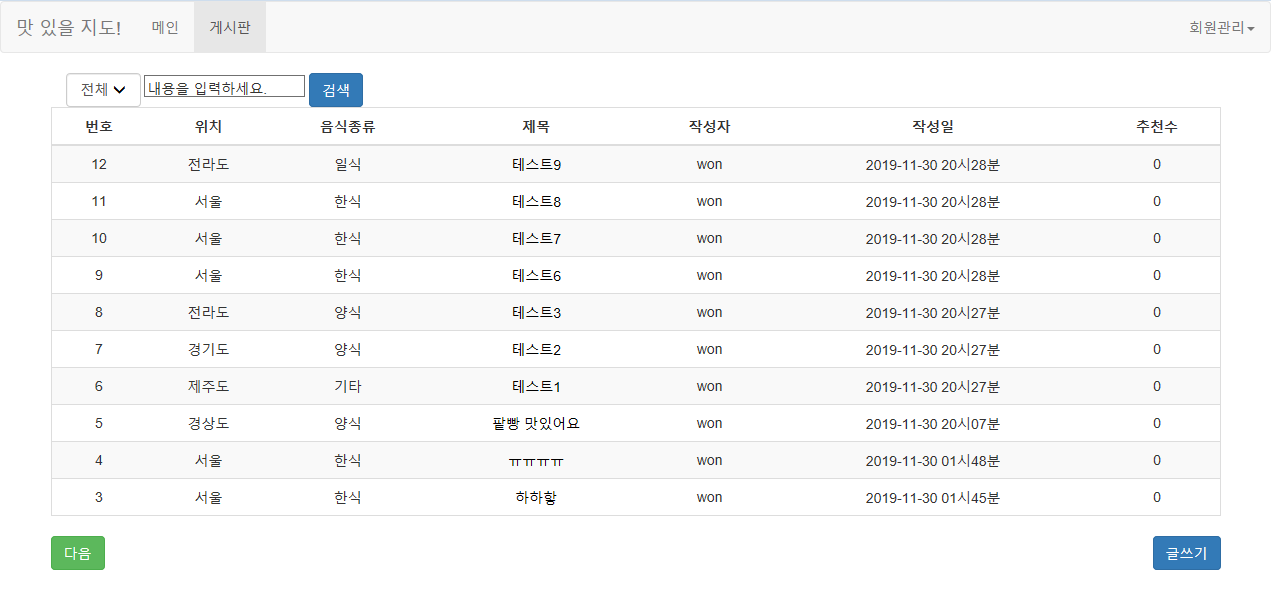
* 게시물 삭제

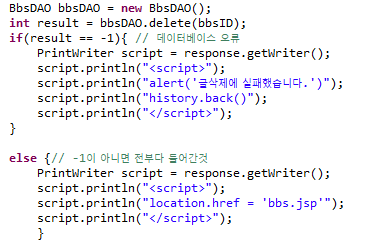
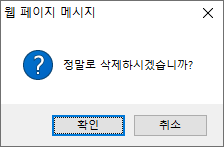
게시물 삭제 버튼을 누르면 ‘정말로 삭제하시겠습니까?’라는 메시지가 나오고 확인을 누르면 게시물 목록으로 이동한다.

이동한 게시물 목록에서는 게시글 삭제가 즉시 반영되어 나타난다.



<게시글이 삭제되기 전 게시물 목록>



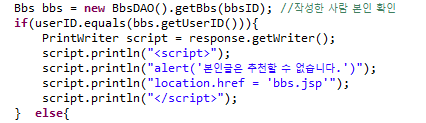
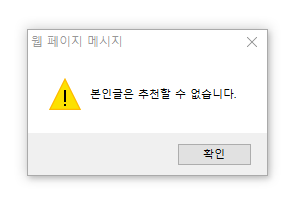
<13번째 게시글이 삭제된 게시글 목록>

게시글 삭제 도중 데이터베이스 연동 오류가 발생하면 ‘글 삭제에 실패했습니다’라는

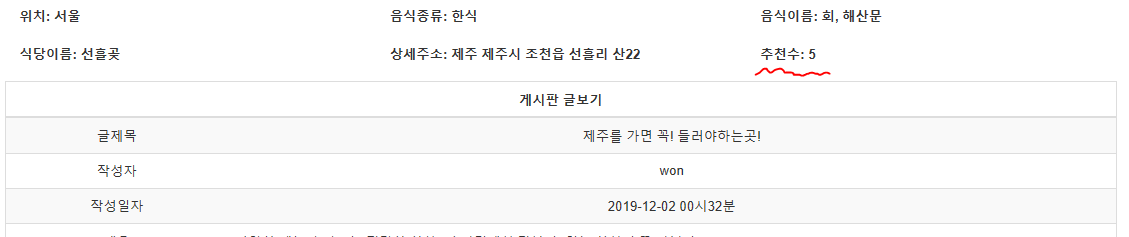
경고메시지를 출력하고 이전화면으로 돌아간다.

* 게시물 추천
* 게시물 추천은 작성자본인은 할수 없다.

게시물 작성을 본인이 했을때의 실행화면과 소스코드는 이다.

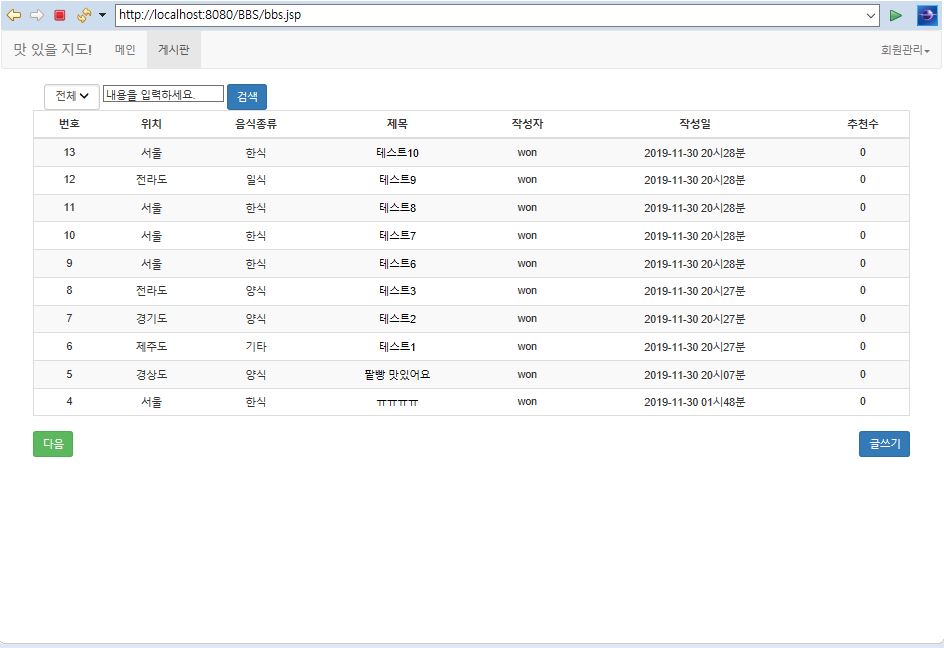


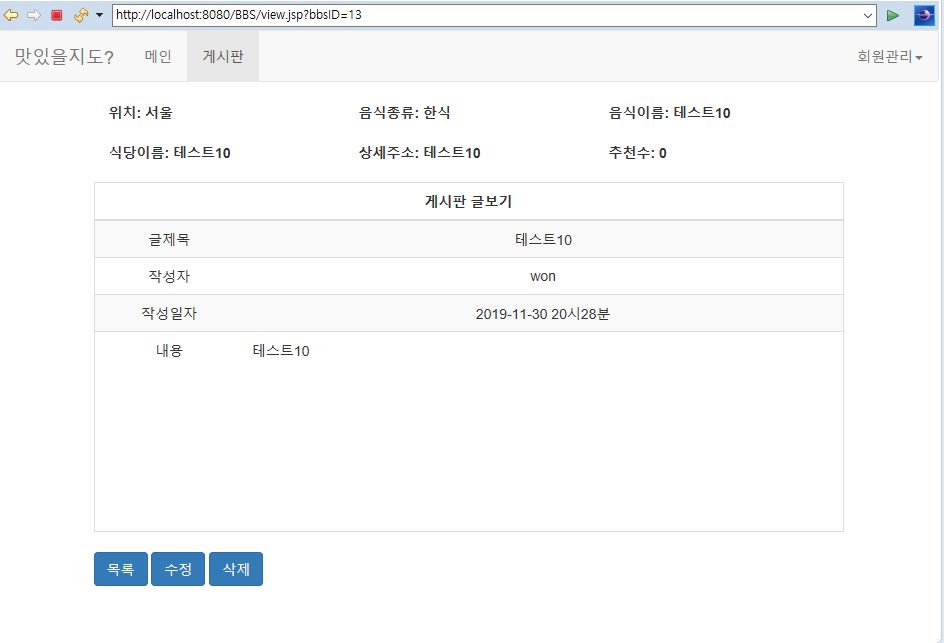
* 로그인 된 사용자가 추천하기를 누르면 게시글의 추천수가 증가한다.



### 게시물 목록 보기

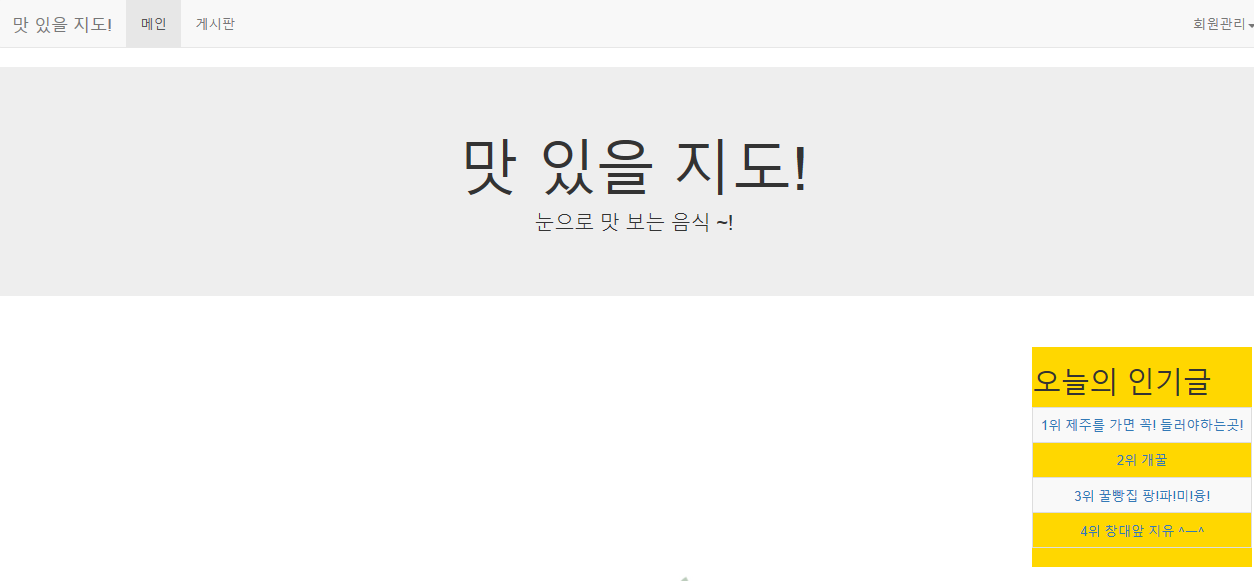
사용자가 게시글을 클릭하면 Bbs.DAO에서 데이터 테이블을 불러와 화면에 출력한다.

아래그림은 게시글을 클릭하면 게시글 상세 내용이 화면에 출력되는 것을 보여주는 것이다.



### 2.8.4 Best10 출력

메인화면에 추천수가 높은 순으로 10가지를 보여준다. 추천수가 1보다 낮은 게시물은 출력하지 않는다.



# SYSTEM TESTING

## 시스템 테스트 시나리오

원영은 맛있을지도!라는 웹사이트에서 맛집이 많은 지역으로 여행을 가려고 한다.

웹 사이트에 들어가자 메인 화면에 표시된 Best10 게시물 중 5개가 여수에 위치한 음식점 게시물이었다.

원영은 여수에 대한 더 많은 게시글을 보기위해 지역별 검색(전라도)을 했다.

최상단에 올라온 게시물인 ‘어서오게장 식당’에 관심이 생긴 원영은 게시물을 클릭하여 확인한다.

글이 마음에 들어 추천버튼을 누르려 했으나 비회원은 이용할 수 없어 원영이 해당 게시글을 추천하기 위해 회원가입을 한 뒤 게시글 추천기능을 이용했다.

여행에서 돌아온 원영은 리뷰를 달기 위해 로그인을 하여 리뷰를 작성하였다.

그후 작성한 리뷰를 다시 읽어보다 오타를 발견한 원영은 게시글 수정을 하였다.

6개월 뒤 원영은 다시 방문한 게장집의 맛이 달라져 실망해 자신이 작성했던 리뷰를

삭제했다.

원영은 로그아웃을 했다.

* 테스트 시나리오 순서에 따른 테스팅 항목

|  |  |  |
| --- | --- | --- |
| 번호 | 항목 | 결과 |
| FR-010 | BEST10 | 성공 |
| FR-004 | 지역별 후기 검색 | 실패 |
| FR-008 | 게시물 출력 | 성공 |
| FR-001 | 회원가입 | 성공 |
| FR-002 | 로그인 | 성공 |
| FR-009 | 게시물 추천 | 성공 |
| FR-005 | 게시물 작성 | 성공 |
| FR-006 | 게시물 수정 | 성공 |
| FR-007 | 게시물 삭제 | 성공 |
| FR-003 | 로그아웃 | 성공 |

## Functional requirement

### FR-001-회원가입

|  |  |  |
| --- | --- | --- |
| 개요 | 사용자는 부분적인 시스템 사용을 위해 회원가입을 해야 한다. |  |
| 선행  조건 | * 사용자는 다른 사용자와 중복되는 ID를 사용할 수 없다. * 사용자는 회원가입 버튼을 누른다. * 사용자는 정해진 형식에 맞게 모든 정보를 입력하여야 한다. * 로그인 한 상태에서는 회원가입을 할 수 없다. |  |
| 후행  조건 | * 다른 사용자와 중복되는 ID 입력 시 경고메시지를 출력한다. * 사용자가 정해진 형식을 지키지 않았을 시 경고메시지를 출력한다. * 사용자가 모든 정보를 입력하지 않았을 시 경고메시지를 출력한다. * 조건에 맞게 회원가입 시 회원 테이블에 저장한다. |  |
| 기본  시나리오 | 1. 사용자는 게시판 왼쪽 상단의 접속하기를 누른다. 2. ‘접속하기’의 드롭다운바에서 '회원가입'을 누른다. 3. 조건에 맞은 정보를 입력하고 회원가입 버튼을 누른다. 4. 사용자가 회원가입에 성공할 시 로그인 화면을 출력한다. |  |
| 대안  시나리오 | * 다른 사용자와 중복된 ID 입력 시나리오  1. 회원가입 화면에서 중복된 ID를 입력하고 회원가입을 누르게 되면 경고메시지를 출력하고 회원가입 화면을 유지한다.  * 필수정보 미입력 시나리오  1. 필수 정보를 입력하지 않고 회원가입을 누르게 되면 경고 메시지를 출력하고 회원가입 화면을 유지한다.  * 조건에 맞지 않는 입력 시나리오  1. 입력 조건에 맞지 않은 형식을 작성하고 회원가입을 누르게 되면 경고 메시지를 출력하고 회원가입 화면을 유지한다. |  |

|  |  |
| --- | --- |
| 테스트 방법 | 사용자는 회원가입 버튼을 누른다.  <회원가입 성공>   1. 사용자는 중복되는 ID가 아닌 새로운 아이디로 정해진 형식에 알맞게 모두 입력을 하였다.   ->메인 화면으로 이동한다.  <회원가입 실패>   1. 사용자는 필수 입력해야하는 정보를 모두 입력하지 않았다.   ->’입력이 안 된 사항이 있습니다.’라는 경고메시지를 출력한다.   1. 사용자는 조건(이메일 형식)에 맞지 않게 입력하였다.   ->’바른 전자메일 형식을 입력하세요’라는 경고메시지를 출력한다.   1. 다른 사용자와 중복되는 ID를 사용하여 회원가입을 시도한다.   ->’이미 존재하는 ID입니다.’라는 경고메시지를 출력한다. |
| 번호 | 테스트 결과 |
| 1 | 성공 |
| 2 | 성공 |
| 3 | 성공 |
| 4 | 성공 |

### FR-002-로그인

|  |  |
| --- | --- |
| 개요 | 사용자는 부분적인 시스템 사용을 위해 ID 로그인을 해야 한다. |
| 선행  조건 | * 사용자는 다른 사용자와 중복되는 ID를 사용할 수 없다. * 사용자는 ID입력란에 ID를 입력한다. * 사용자는 PW입력란에 PW를 입력한다. * 사용자는 자신의 ID와 PW에 일치하는 정보를 입력해야 한다. |
| 후행  조건 | * 존재하지 않는 ID 입력 시 경고메시지를 출력한다. * ID와 PW가 DB의 데이터와 일치하지 않을 시 경고메시지를   출력한다.   * DB의 데이터와 일치하는 ID와 PW를 입력할 시 정상적으로 로그인   작업을 수행한다. |
| 기본  시나리오 | 1. 사용자는 게시판 왼쪽 상단의 접속하기 버튼을 누른다. 2. '접속하기'의 드롭다운바에서 '로그인'을 누른다. 3. ID를 입력한다. 4. 사용자는 게시판 왼쪽 상단의 PW입력란을 클릭하거나   Tab키를 이용하여 PW입력란으로 이동한다.   1. PW를 입력한다. 2. 조건에 맞는 데이터를 입력하고 로그인 버튼을 누른다. 3. 사용자가 회원가입에 성공할 시 다시 메인화면을 출력한다. |
| 대안  시나리오 | * 일치하지 않는 ID 입력 시나리오   1. "ID혹은 PW가 일치하지 않습니다"라는 경고메시지를 출력한다.  2. 메인화면으로 돌아간다.   * 일치하는 ID를 입력했으나 PW가 맞지 않는 시나리오   1. "ID혹은 PW가 일치하지 않습니다"라는 경고메시지를 출력한다.  2. 메인화면으로 돌아간다. |

|  |  |
| --- | --- |
| 테스트 방법 | 사용자는 로그인 버튼을 누른다.  <로그인 성공>   1. 사용자 DB에 저장되어있는 회원정보로 로그인 하였다.   ->로그인에 성공한 사용자는 메인화면으로 이동한다.  <로그인 실패>’   1. 사용자ID와 PASSWORD가 일치하지 않는다.   ->’비밀번호가 틀립니다’라는 경고메시지를 출력한다.   1. 사용자ID가 사용자 DB에 저장되어있지 않다.   ->’존재하지 않는 아이디입니다’라는 경고메시지를 출력한다.   1. 데이터베이스와의 연동 오류   ->’데이터베이스 오류가 발생했습니다’라는 경고메시지를  출력한다. |
| 번호 | 테스트 결과 |
| 1 | 성공 |
| 2 | 성공 |
| 3 | 성공 |
| 4 | 성공 |

### FR-003-로그아웃

|  |  |
| --- | --- |
| 개요 | 사용자는 시스템과의 연결을 끊을 때 로그아웃을 한다. |
| 선행  조건 | * 사용자는 반드시 로그인 한 상태여야 한다. * 로그아웃 버튼을 클릭한다. |
| 후행  조건 | * 사용자가 로그아웃 버튼을 클릭하면 시스템과의 접속을 끊는다. * 로그아웃 되었다는 메시지 출력 후 메인화면으로 이동한다. |
| 기본  시나리오 | 1. 사용자가 로그아웃 버튼을 클릭한다. 2. 로그아웃 되었다는 메시지와 함께 메인화면으로 이동한다. |

|  |  |
| --- | --- |
| 테스트 방법 | 사용자는 로그인 되어있는 상태이다.  <로그아웃 성공>   1. 사용자는 로그아웃 버튼을 누르면 로그아웃이 완료된다.   ->로그아웃에 성공한 사용사는 메인화면으로 이동한다. |
| 번호 | 테스트 결과 |
| 1 | 성공 |

### FR-004-지역별 후기 검색

|  |  |
| --- | --- |
| 개요 | 사용자는 화면과 게시판 화면에서 지역별 후기 검색이 가능하다. |
| 선행  조건 | * 사용자는 게시판 화면에 존재한다. * 검색할 지역을 선택지에서 찾아 선택한다. |
| 후행  조건 | * 사용자가 선택한 조건에 알맞은 검색 결과를 최신순으로 화면에 출력한다. * 사용자가 검색 결과 화면에서 게시물을 클릭하면 해당 게시물   화면을 출력한다.   * 사용자가 검색한 결과를 선택하지 않으면 검색 결과 화면에   머무른다. |
| 기본  시나리오 | 1. 사용자는 게시판 화면에서 조건(지역)을 선택지 중 선택한다. 2. 시스템은 선택한 조건에 해당하는 게시물을 최신순으로 검색결과 화면에 출력한다. |
| 대안  시나리오 | * 선택지와 일치하는 조건의 게시물이 없는 경우  1. 선택지와 일치하는 조건의 게시물이 없는 경우   ‘일치하는 결과가 없습니다.’라는 메시지 혹은 화면을 출력한다. |

|  |  |
| --- | --- |
| 테스트 방법 | 사용자는 게시판 화면에 존재한다.  <지역별 검색 성공>   1. 사용자가 지정한 조건에 맞는 검색 결과가 화면에 표시된다.   <지역별 검색 실패>   1. 사용자가 지정한 조건에 맞는 검색 결과가 없다.   ->일치하는 결과가 없습니다’라는 경고메시지를 출력한다.   1. 데이터베이스와의 연동 오류   ->’데이터베이스 오류가 발생했습니다’라는 경고메시지를 출력한다. |
| 번호 | 테스트 결과 |
| 1 | 구현 실패로 테스트 못함 |

### FR-005-게시물 작성

|  |  |
| --- | --- |
| 개요 | 사용자는 개인 프로필 화면에서 위치 카테고리에서 원하는 지역을 선택한 뒤 게시물 작성을 한다. 이후 위치 카테고리에 본인이 작성한 게시물에 관한 위치 정보가 나타내진다. |
| 선행  조건 | * 사용자는 시스템에 로그인해야 한다. * 사용자는 메인 화면에 존재한다. * 사용자는 원하는 지역을 클릭하고 자세한 정보는 '위치'란에 입력한다. * 게시물은 텍스트 형식에 맞게 작성해야 한다. |
| 후행  조건 | * 사용자가 틀린 텍스트 형식으로 입력한 뒤 게시물 작성완료 버튼을 누를 경우 경고메시지를 출력한다. * 사용자가 텍스트 입력란에 아무 내용도 기입하지 않을 경우 경고메시지를 출력한다. * 사용자는 게시물 작성 후 게시물 화면에 머무르거나 메인 화면으로 돌아갈 수 있다. * 작성이 완료되면 여행 정보 테이블에 데이터가 저장된다. |
| 기본  시나리오 | 1. 사용자는 게시물 작성 화면의 위치 선택란에서 위치를 선택한다. 2. 선택한 위치에서 게시물 작성 버튼을 누른다. 3. 시스템은 지도 위에 게시물 작성란을 출력한다. 4. 사용자는 텍스트 형식에 맞게 게시물을 작성한다. |
| 대안  시나리오 | * 형식에 맞지 않는 게시물 작성 시나리오  1. 사용자가 텍스트 형식에 맞지 않게 입력하고 작성 완료 버튼을 클릭할 경우 경고메시지를 출력하고 게시물 작성 화면을 유지한다.  * 게시물 내용 미입력 시나리오  1. 사용자가 텍스트란에 아무 내용도 기입하지 않을 경우 경고메시지를 발생시키고 게시물 작성 화면을 유지한다. |

|  |  |
| --- | --- |
| 테스트 방법 | 사용자는 메인화면에 존재한다.  <게시글 작성 성공>   1. 사용자가 작성해야하는 내용을 빠짐없이 모두 작성한다.   ->게시글 목록에서 사용자가 작성한 글을 확인할 수 있다.  <게시글 작성 실패>   1. 사용자가 작성해야하는 항목을 모두 작성하지 않았다.   ->’입력이 안 된 사항이 있습니다’라는 경고 메시지를 출력한다.   1. 로그인이 되어있지 않다.   ->’로그인을 하세요’라는 경고 메시지를 출력하고 로그인 페이지로 이동한다.   1. 데이터 베이스 오류   ->’글쓰기에 실패했습니다’라는 경고 메시지를 출력한다. |
| 번호 | 테스트 결과 |
| 1 | 성공 |
| 2 | 성공 |
| 3 | 성공 |
| 4 | 성공 |

### FR-006-게시물 수정

|  |  |
| --- | --- |
| 개요 | 사용자는 개인 게시물 화면에서 수정 버튼을 눌러 수정한다. 게시물 수정 후에는 게시판에 사용자가 작성한 게시물의 관한 정보가 나타내진다. |
| 선행  조건 | * 사용자는 시스템에 로그인해야 한다. * 사용자는 자신이 작성한 게시물을 클릭한다. * 사용자는 자신이 작성한 게시물 화면에 존재한다. * 사용자는 수정 버튼을 누른다. * 사용자는 게시물 작성 화면에 존재한다. |
| 후행  조건 | * 사용자가 틀린 형식으로 게시물을 작성 완료 버튼을 클릭할 경우   경고메시지를 출력한다.   * 사용자가 사진과 텍스트란에 아무 내용도 기입하지 않을 경우 경고메시지를 출력한다. * 사용자는 게시물 수정 후 게시물 화면에 머무르거나 메인 화면으로 돌아갈 수 있다. * 작성이 완료되면 여행 정보 테이블에 데이터가 저장된다. |
| 기본  시나리오 | 1. 사용자는 자신이 작성한 게시물 화면에서 수정 버튼을 누른다. 2. 사용자는 텍스트 형식에 맞게 게시물을 수정한다. |
| 대안  시나리오 | * 형식에 맞지 않는 게시물 작성 시나리오  1. 사용자가 텍스트 형식에 맞지 않게 입력하고 작성 완료 버튼을 클릭할 경우 경고메시지를 출력하고 게시물 작성 화면을 유지한다.  * 게시물 내용 미 입력 시나리오  1. 사용자가 텍스트란에 아무 내용도 기입하지 않을 경우 경고메시지를 출력하고 게시물 작성 화면을 유지한다.  * 자신이 작성한 게시물이 아닌 시나리오  1. 사용자가 다른 사람이 작성한 게시물을 보고 있을 경우 수정 버튼을 출력하지 않는다. |

|  |  |
| --- | --- |
| 테스트 방법 | 사용자는 게시물 출력 화면에 존재한다.  <게시글 수정 성공>   1. 게시글을 작성한 본인이 모든 항목에 알맞은 형식으로 게시글을 수정한다.   ->게시글 수정에 성공하면 게시글이 수정되고 게시판 화면으로 돌아간다.  <게시글 수정 실패>   1. 작성하지 않은 항목이 있는경우   ->’입력이 안 된 사항이 있습니다’라는 경고메시지를 출력하고 게시글 수정 화면으로 돌아간다.   1. 데이터베이스 연동 오류   ->’글수정에 실패했습니다’라는 경고메시지를 출력하고 게시글 수정화면으로 돌아간다. |
| 번호 | 테스트 결과 |
| 1. | 성공 |
| 2. | 성공 |
| 3. | 성공 |

|  |  |
| --- | --- |
| 개요 | 사용자는 개인 게시물 화면에서 삭제 버튼을 누른다. 삭제 후에는 게시판에 사용자가 작성한 게시물에 관한 정보가 삭제된다. |
| 선행  조건 | * 사용자는 시스템에 로그인해야 한다. * 사용자는 자신이 작성한 게시물을 클릭한다. * 사용자는 자신이 작성한 게시물 화면에 존재한다. * 사용자는 삭제 버튼을 누른다. |
| 후행  조건 | * 사용자가 잘못 클릭한 경우를 위해 경고메시지를 출력한다. * 삭제가 완료되면 여행 정보 테이블에서 데이터가 삭제된다. |
| 기본  시나리오 | 1. 사용자는 자신이 작성한 게시물 화면에서 삭제 버튼을 누른다. 2. 사용자는 게시판 화면 또는 메인화면으로 돌아갈 수 있다. |
| 대안  시나리오 | * 삭제 버튼을 잘못 클릭한 시나리오  1. 실수로 게시물이 삭제되는 것을 방지하기 위해 시스템은   ‘정말 삭제하시겠습니까?’, ‘예/아니오’ 라는 선택메시지를 출력한다. |

### FR-007-게시물 삭제

|  |  |
| --- | --- |
| 테스트 방법 | 사용자는 게시물 출력 화면에 존재한다.  <게시글 삭제 성공>   1. 사용자가 작성했던 글이 게시글 목록에서 사라진다.   ->게시글 삭제에 성공한 사용자는 게시판 화면으로 이동한다.  <게시글 삭제 실패>   1. .데이터베이스와 연동 오류   ->’글삭제에 실패했습니다’라는 경고메시지를 출력한다. |
| 번호 | 테스트 결과 |
| 1 | 성공 |
| 2 | 성공 |

### FR-008-게시물 출력

|  |  |
| --- | --- |
| 개요 | 메인 화면에서 게시판의 게시물들을 화면상에 출력한다. |
| 선행  조건 | * 사용자는 게시판 화면에 존재한다. * 사용자는 보고자 하는 게시물을 클릭해야 한다. |
| 후행  조건 | * 사용자가 게시판 버튼을 클릭할 경우 게시판 화면을 출력한다. * 사용자가 세부 정보 버튼을 클릭하지 않을 경우 메인 화면을 유지한다. |
| 기본  시나리오 | 1. 사용자는 메인 화면에 표시된 타인 게시물 정보 혹은 게시판을 클릭한다. 2. 타인 게시물 정보를 클릭한 경우 시스템은 메인 화면에서 클릭한 게시물을 타인 게시물 화면에 출력한다. 3. 게시판을 클릭한 경우 시스템은 게시판 화면을 출력한다. 4. 사용자는 게시판 화면에서 보고 싶은 게시물을 클릭한다. 5. 시스템은 게시물 화면을 출력한다. |
| 대안  시나리오 | 없다. |

|  |  |
| --- | --- |
| 테스트 방법 | 사용자는 게시판 화면에 존재한다.  사용자가 보고자하는 게시물을 클릭한다.  <게시물 출력 성공>   1. 사용자가 클릭한 게시물의 상세 내용이 화면에 출력된다.   <게시물 출력 실패>   1. BbsID오류   ->’유효하지 않은 글입니다’라는 경고메시지를 출력한다 |
| 번호 | 테스트 결과 |
| 1 | 성공 |
| 2 | 성공 |

### FR-009-게시물 추천

|  |  |
| --- | --- |
| 개요 | 사용자는 다른 사용자가 작성한 게시물을 추천 할 수 있다. |
| 선행  조건 | * 사용자는 시스템에 로그인 한 상태여야 한다. * 사용자는 게시물이 출력된 화면에 존재해야 한다. |
| 후행  조건 | * 해당 게시물의 추천수가 1만큼 증가한다. |
| 기본  시나리오 | 1. 사용자가 게시물 목록 화면에서 원하는 게시물을 클릭한다. 2. 시스템은 사용자가 선택한 게시물을 화면에 출력한다. 3. 사용자는 게시물이 도움이 되었다고 판단되면 추천 버튼을   클릭한다.   1. 시스템은 해당 게시물의 추천수를 1증가시켜 화면에 출력한다. |
| 대안  시나리오 | * 사용자가 게시물의 추천 버튼을 중복하여 클릭한 경우 시나리오  1. 사용자가 중복해서 게시물 추천했다는 경고메시지를 출력하고 시스템은 추천수를 그대로 유지한다. |

|  |  |
| --- | --- |
| 테스트 방법 | 사용자는 게시판 내용 보기 중 추천하기 버튼을 클릭한다.  <게시물 추천 성공>   1. 게시물의 추천수가 1증가한다.   <게시물 추천 실패>   1. 본인이 작성한 게시물은 추천수가 오르지 않는다.  * 본인글은 추천할 수 없습니다. 라는 경고메시지를 출력한다.  1. 한번 추천한 사용자는 추천할 수 없다. |
| 번호 | 테스트 결과 |
| 1 | 성공 |
| 2 | 성공 |
| 3 | 실패 |

### FR-010-BEST10

|  |  |
| --- | --- |
| 항목 | 내용 |
| 설명 | 추천수가 높은 게시물10가지를 게시판 화면에 출력하기 위한 요구사항 |
| 전제 조건 | 게시물의 추천수가 1이상이다.  게시물의 추천수가 같다면 시간상 최신인 것이 상위로 올라간다. |
| 충족 기준 | 메인 화면에서 BEST10 게시물이 출력된다. |

## Non-Functional requirement

### QR-001-개인정보보호

|  |  |
| --- | --- |
| 항목 | 내용 |
| 설명 | 회원들의 개인정보 보호를 위한 요구사항 |
| 전제 조건 | 회원 개인 정보로 이름, 전화번호, ID, PASSWORD가 저장  되어있다.  전화번호와 PASSWORD는 암호화된 형태로 저장된다. |
| 충족 기준 | 전화번호와 PASSWORD는 타인이 볼 수 없다. |
| 테스트 방법 | 전화번호와 PASSWORD입력시 타인이 볼 수 있는지 확인. |
| 테스트 결과 | 성공 |

### QR-002-게시물 실시간 처리

functional requirement의 FR-005, FR-006, FR-007을 성공적으로 구현한 상태이고 테스트해본 결과 등록, 삭제, 수정 완료한 게시글이 9초 이내에 즉시 반영된 것을 확인할 수 있었다.

이는 updataAction.jsp, write.jsp 실행시 확인 가능하다.

|  |  |
| --- | --- |
| 항목 | 내용 |
| 설명 | 사용자가 새로운 게시물을 등록, 삭제, 수정하면 해당 게시물이 9초 내에 등록, 삭제, 수정 완료 되어야한다. |
| 전제 조건 | 사용자는 로그인 되어있어야 한다.  사용자는 게시글 작성 형식에 맞게 게시물을 작성해야 한다. |
| 충족 기준 | 게시물 작성자가 완료 버튼을 누른 후 게시물 목록에 들어갔을 때 작성한 게시물이 목록에 검색되거나 삭제되어 있어야한다. |
| 테스트 방법 | 게시물을 등록, 수정, 삭제를 해보고 걸린 시간을 확인한다. |
| 테스트 결과 | 대부분의 경우 1초이내에 완료된다. |

### QR-003-게시물 검색 속도

구현하지 못하여 테스트가 불가한다.

|  |  |
| --- | --- |
| 항목 | 내용 |
| 설명 | 지역 카테고리를 통한 후기 검색 속도에 대한 요구사항 |
| 전제 조건 | 지역은 1특별시, 6광역시, 9도인 16개 지역 중 1가지 중에서 선택 가능하다. |
| 충족 기준 | 검색버튼 클릭 이후 검색 결과가 5초 이내에 브라우저에 출력된다. |
| 테스트 결과 | 테스트 불가능(미구현) |

### QR-004-시스템 장애복구

|  |  |
| --- | --- |
| 항목 | 내용 |
| 설명 | 장애 발생시 5시간 이내에 정상 상태로 복구되기 위한 요구사항 |
| 전제 조건 | 시스템 장애가 발생하면 시스템은 장애가 발생한 상황을 사용사들에게 알릴 수 있다. |
| 충족 기준 | 장애가 발생한시 5시간 내에 정상 상태로 복구되어야 한다. |
| 테스트 결과 | 테스트 불가능(코드로 인한 오류가아닌 시스템장애가 발생한적이 없어 확인이 불가능하였다.) |

### QR-005-24시간 서비스

QR-006에 대한 테스팅은 시스템의 전체적인 기능이 구현 완료된 후에 진행할 예정이다.

|  |  |
| --- | --- |
| 항목 | 내용 |
| 설명 | 서버가 24시간 동안 서비스되기 위한 요구사항 |
| 전제 조건 | 서버 점검시간(매월 마지막 주 수요일 새벽 1시~2시)을 제외하고 24시간 서비스 이용이 가능하다. |
| 충족 기준 | 점검시간을 제외하고 24시간 이용이 가능하다. |
| 테스트 방법 | 서버를 24시간 사용이 가능하도록 30시간동안 켜놓아 보았다. |
| 테스트 결과 | 성공(30시간동안 아무런 이상이없었다.) |

### QR-006-접근성

접근성을 높이기 위해 배경은 흰색/회색, 글씨는 검정색으로 대조되게 구현하였다.

글꼴은 고딕류의 글꼴을 사용하여 가독성을 높였다.

|  |  |
| --- | --- |
| 항목 | 내용 |
| 설명 | 인터넷 사용에 미숙한 장애인, 고령자도 이용이 용이하게 하기위한 요구사항 |
| 전제 조건 | 글꼴은 필기체를 사용하지 않는다.  배경색과 대조되는 글자색을 사용한다. |
| 충족 기준 | 한글을 읽을 수 있는 사용자는 소프트웨어의 사용이 가능하다. |
| 테스트 결과 | 성공 |

# 결함 및 개선방향

|  |  |
| --- | --- |
| 결함 기능 | 개선방향 |
| 카테고리별 검색 기능 | 프로젝트 기간은 끝났지만 구현 예정입니다. |
| 지도 클릭으로 인한 각 지역별 게시판으로 이동 | 지도 클릭으로 인한 이동은 구현하였으므로 지역별 검색만 구현할 예정입니다. |