**[유스케이스 시나리오]**

[공지사항 작성]

목표 : 관리자로 로그인 한 유저만 공지사항의 작성이 가능하다. 공지사항은 제목 및 내용이 반드시 포함되어 있어야 한다.

성공 시나리오

1. 관리자로 로그인한 유저에게만 웹페이지의 공지사항 작성 탭으로 접근 가능하다.

2. 공지사항 작성 탭으로 접근하면 웹페이지에서 토큰이 생성됨(새로고침 등의 중복요청으로 인한 게시물의 등록을 방지하기 위한 일련번호 생성)

3. 공지사항 제목 및 내용을 작성한 후 작성 버튼을 클릭하면 공지사항 작성시 생성된 토큰과 요청 토큰을 비교한 후 서로 일치하는 경우에만 데이터베이스에 공지사항 번호, 제목, 내용, 작성자 아이디, 작성 날짜가 등록됨.

4. 데이터베이스에 등록이 성공하면 성공 알림을 띄운 뒤에 공지사항 탭으로 돌아간다.

확장 시나리오

3.1. 제목이나 내용 중 누락된 값이 하나라도 존재하는 경우 작성하지 않은 부분이 있다는 알림을 띄운 후, 해당 텍스트 박스에 커서를 이동시킨다.

3.2. 생성된 토큰과 공지사항 작성 요청시의 토큰이 일치하지 않는 경우, 등록 실패 알림을 띄운다.

4.1. 데이터베이스에 등록이 실패한 경우, 등록 실패 알림을 띄운다.

[게시판 글 작성]

목표 : 회원 및 관리자로 로그인 한 유저만 게시판의 글 작성이 가능하다. 게시판 글은 제목 및 내용이 반드시 포함되어 있어야 한다.

성공 시나리오

1. 회원 및 관리자로 로그인 한 유저에게만 게시판에서 글 작성 탭으로 접근 가능하다.

2. 게시판 글 작성 탭으로 접근하면 웹페이지에서 토큰이 생성됨

3. 제목 및 내용(글 혹은 그림을 반드시 포함)을 작성한 후 작성 버튼을 클릭하면 생성된 토큰과 게시물 작성 시의 요청 토큰을 비교한 후 서로 일치하는 경우에만 데이터베이스에 게시물 번호, 제목, 내용, 작성자 아이디, 작성 날짜가 등록됨.

4. 데이터베이스에 등록이 성공하면 성공 알림을 띄운 뒤에 게시판 탭으로 돌아간다.

확장 시나리오

3.1. 제목이나 내용 중 누락된 값이 하나라도 존재하는 경우 작성하지 않은 부분이 있다는 알림을 띄운 후, 해당 텍스트박스에 커서를 이동시킨다.

3.2. 생성된 토큰과 게시물 작성 요청 시의 토큰이 일치하지 않는 경우, 등록 실패 알림을 띄운다.

4.1. 데이터베이스에 등록이 실패한 경우, 등록 실패 알림을 띄운다.

[댓글 작성]

목표 : 회원 및 관리자로 로그인 한 유저만 게시판, 갤러리의 게시물에 댓글 작성이 가능하다. 댓글에는 내용이 반드시 포함되어야 한다.

성공 시나리오

1. 회원 및 관리자로 로그인 한 유저만 게시판, 갤러리의 게시물에서 댓글 작성에 접근할 수 있다.

2. 댓글 내용을 작성한 후 작성 버튼을 클릭하면 작성 시의 요청 토큰과 현재 토큰 값을 비교하여 서로 일치하는 경우에만 데이터베이스에 댓글 번호, 게시물 번호, 작성자 아이디, 댓글 내용, 현재 날짜가 등록된다.

3. 데이터베이스에 등록이 성공하면 성공 알림을 띄운 뒤, 댓글을 작성한 게시물로 다시 이동한다.

확장 시나리오

2.1. 요청 토큰과 현재 토큰 값이 일치하지 않는 경우, 등록 실패 알림을 띄운다.

2.2. 내용이 누락된 경우, 내용이 누락되었다는 알림을 띄우고, 해당 텍스트박스에 커서를 이동시킨다.

3.1. 데이터베이스에 등록이 실패한 경우, 등록 실패 알림을 띄운다.

[게시판(커뮤니티) 글 수정(SNS스타일)]

목표 : 사용자는 작성한 글에 접근하여 내용을 수정할 수 있다.

사전조건 : 로그인이 되어있어야 한다.

성공시나리오

1. 사용자가 작성한 글의 수정 버튼을 클릭한다

2. 시스템은 기존 게시글 내용을 읽어와 수정 페이지 입력 화면에 표출한다.

3. 사용자는 원하는 내용으로 글을 수정하고 수정완료 버튼을 클릭한다.

4. 시스템은 수정사항을 저장할지 확인/취소 선택버튼을 출력한다.

5. 사용자는 확인 버튼을 클릭한다.

6. 시스템은 수정된 내용으로 글을 대체한다.

7. 시스템은 게시판 글 수정 완료 메세지를 보여주고 게시판(커뮤니티)을 재 호출한다.

확장시나리오

3.1. 수정 화면에서 취소를 클릭하는 경우 :

3.1.1. 시스템은 게시글 접근 화면을 출력한다.

3.2. 내용 및 업로드 할 이미지 까지 두개 항목이 다 비어있는 경우 :

3.2.1. 작성하지 않은 부분이 있다는 알림을 띄운 후,

내용 작성 텍스트 박스에 커서를 이동시킨다.

4.1. 수정 완료 버튼 클릭 후 저장 여부 확인에서 취소를 클릭하는 경우 :

4.1.1. 시스템은 수정 화면 상태에 머무른다.

7.1. 수정 완료 후 게시판 재 호출시 :

7.1.1. 시스템은 재 호출시 수정 시간을 기준으로 게시글을 재정렬한다.

[공지사항 수정]

목표 : 관리자는 작성한 공지사항에 접근하여 내용을 수정할 수 있다.

사전조건 : 로그인이 되어있어야 한다./ 관리자여야한다.

성공시나리오

1. 공지사항 글에 접근하여 수정 버튼을 클릭한다

2. 시스템은 기존 공지사항 내용을 읽어와 입력 화면에 표출한다.

3. 관리자는 원하는 형태로 내용을 수정하고 수정완료 버튼을 클릭한다.

4. 시스템은 수정사항을 저장할지 확인/취소 선택버튼을 출력한다.

5. 관리자는 확인 버튼을 클릭한다.

6. 시스템은 수정된 내용으로 공지사항 글을 대체한다.

7. 시스템은 공지사항 수정 완료 메세지를 보여주고 수정된 공지사항 접근 화면을 보여준다.

확장시나리오

3.1. 수정 화면에서 취소를 클릭하는 경우

3.1.1. 시스템은 해당 공지사항 접근 화면을 출력한다.

3.2. 제목이나 내용 중 누락된 값이 하나라도 존재하는 경우

3.2.1 작성하지 않은 부분이 있다는 알림을 띄운 후, 해당 텍스트 박스에 커서를 이동시킨다.

4.1. 수정 완료 버튼 클릭 후 저장 여부 확인에서 취소를 클릭하는 경우

4.1.1. 시스템은 수정 화면 상태에 머무른다.

[게시판 열람]

목표 : 사용자는 게시판을 열람하다.

(게시판은 SNS 스타일)

(게시판 열람은 게시판에 접속, 게시판에서 게시글 조회를 포함)

(이미지나 글내용이 게시판에 접속하면 바로 보인다고 가정 / 아래로 쭉 내리면 새 글들이 나옴)

(게시글은 최신순 / 20개씩 표기)

성공시나리오

1. 사용자는 메뉴바에서 게시판을 클릭한다.

2. 시스템은 게시판을 응답한다.

3. 사용자의 화면에 게시판이 표시된다.

4. 사용자는 원하는 게시글을 클릭한다

5. 시스템은 사용자가 클릭한 게시글을 응답한다.

(이미지가 여러개면 슬라이드 형식으로 보임)

6. 사용자 화면에 게시글이 표시된다.

(댓글들은 게시글 아래에 있음, 댓글 작성은 댓글들 아래 있음)

확장시나리오

4.1. 페이지를 아래로 끝까지 내림

4.1.1. 사용자는 페이지를 아래로 끝까지 내린다.

4.1.2. 시스템이 사용자 화면에 띄어줄 게시글들을 반환한다.

4.1.3. 사용자의 화면에 새 게시글들이 추가된다.

4.3. 조회기능 사용

4.3.1. 사용자는 검색창에 검색어를 입력, "검색" 버튼 클릭

4.3.2. 시스템은 검색어에 따라서 해당하는 게시글들을 응답

4.3.2. 사용자의 화면에 검색된 게시글들이 표시된다

6.1. 게시글에서 좋아요 or 즐겨찾기 클릭

6.1.1. 로그인 상태인지 확인

6.1.1.1. 로그인 상태가 아니면 로그인 페이지 띄움

6.1.1.2. user 화면에 로그인 페이지 나옴

6.1.2.1. 로그인 상태면 좋아요 수 증가, 사용자 계정의 좋아요한 글에 해당 글을 추가

[유스케이스 시나리오]

[공지사항 작성]

목표 : 관리자로 로그인 한 유저만 공지사항의 작성이 가능하다. 공지사항은 제목 및 내용이 반드시 포함되어 있어야 한다.

성공 시나리오

1. 관리자로 로그인한 유저에게만 웹페이지의 공지사항 작성 탭으로 접근 가능하다.

2. 공지사항 작성 탭으로 접근하면 웹페이지에서 토큰이 생성됨(새로고침 등의 중복요청으로 인한 게시물의 등록을 방지하기 위한 일련번호 생성)

3. 공지사항 제목 및 내용을 작성한 후 작성 버튼을 클릭하면 공지사항 작성시 생성된 토큰과 요청 토큰을 비교한 후 서로 일치하는 경우에만 데이터베이스에 공지사항 번호, 제목, 내용, 작성자 아이디, 작성 날짜가 등록됨.

4. 데이터베이스에 등록이 성공하면 성공 알림을 띄운 뒤에 공지사항 탭으로 돌아간다.

확장 시나리오

3.1. 제목이나 내용 중 누락된 값이 하나라도 존재하는 경우 작성하지 않은 부분이 있다는 알림을 띄운 후, 해당 텍스트 박스에 커서를 이동시킨다.

3.2. 생성된 토큰과 공지사항 작성 요청시의 토큰이 일치하지 않는 경우, 등록 실패 알림을 띄운다.

4.1. 데이터베이스에 등록이 실패한 경우, 등록 실패 알림을 띄운다.

[게시판 글 작성]

목표 : 회원 및 관리자로 로그인 한 유저만 게시판의 글 작성이 가능하다. 게시판 글은 제목 및 내용이 반드시 포함되어 있어야 한다.

성공 시나리오

1. 회원 및 관리자로 로그인 한 유저에게만 게시판에서 글 작성 탭으로 접근 가능하다.

2. 게시판 글 작성 탭으로 접근하면 웹페이지에서 토큰이 생성됨

3. 제목 및 내용(글 혹은 그림을 반드시 포함)을 작성한 후 작성 버튼을 클릭하면 생성된 토큰과 게시물 작성 시의 요청 토큰을 비교한 후 서로 일치하는 경우에만 데이터베이스에 게시물 번호, 제목, 내용, 작성자 아이디, 작성 날짜가 등록됨.

4. 데이터베이스에 등록이 성공하면 성공 알림을 띄운 뒤에 게시판 탭으로 돌아간다.

확장 시나리오

3.1. 제목이나 내용 중 누락된 값이 하나라도 존재하는 경우 작성하지 않은 부분이 있다는 알림을 띄운 후, 해당 텍스트박스에 커서를 이동시킨다.

3.2. 생성된 토큰과 게시물 작성 요청 시의 토큰이 일치하지 않는 경우, 등록 실패 알림을 띄운다.

4.1. 데이터베이스에 등록이 실패한 경우, 등록 실패 알림을 띄운다.

[댓글 작성]

목표 : 회원 및 관리자로 로그인 한 유저만 게시판, 갤러리의 게시물에 댓글 작성이 가능하다. 댓글에는 내용이 반드시 포함되어야 한다.

성공 시나리오

1. 회원 및 관리자로 로그인 한 유저만 게시판, 갤러리의 게시물에서 댓글 작성에 접근할 수 있다.

2. 댓글 내용을 작성한 후 작성 버튼을 클릭하면 작성 시의 요청 토큰과 현재 토큰 값을 비교하여 서로 일치하는 경우에만 데이터베이스에 댓글 번호, 게시물 번호, 작성자 아이디, 댓글 내용, 현재 날짜가 등록된다.

3. 데이터베이스에 등록이 성공하면 성공 알림을 띄운 뒤, 댓글을 작성한 게시물로 다시 이동한다.

확장 시나리오

2.1. 요청 토큰과 현재 토큰 값이 일치하지 않는 경우, 등록 실패 알림을 띄운다.

2.2. 내용이 누락된 경우, 내용이 누락되었다는 알림을 띄우고, 해당 텍스트박스에 커서를 이동시킨다.

3.1. 데이터베이스에 등록이 실패한 경우, 등록 실패 알림을 띄운다.

[게시판(커뮤니티) 글 수정(SNS스타일)]

목표 : 사용자는 작성한 글에 접근하여 내용을 수정할 수 있다.

사전조건 : 로그인이 되어있어야 한다.

성공시나리오

1. 사용자가 작성한 글의 수정 버튼을 클릭한다

2. 시스템은 기존 게시글 내용을 읽어와 수정 페이지 입력 화면에 표출한다.

3. 사용자는 원하는 내용으로 글을 수정하고 수정완료 버튼을 클릭한다.

4. 시스템은 수정사항을 저장할지 확인/취소 선택버튼을 출력한다.

5. 사용자는 확인 버튼을 클릭한다.

6. 시스템은 수정된 내용으로 글을 대체한다.

7. 시스템은 게시판 글 수정 완료 메세지를 보여주고 게시판(커뮤니티)을 재 호출한다.

확장시나리오

3.1. 수정 화면에서 취소를 클릭하는 경우 :

3.1.1. 시스템은 게시글 접근 화면을 출력한다.

3.2. 내용 및 업로드 할 이미지 까지 두개 항목이 다 비어있는 경우 :

3.2.1. 작성하지 않은 부분이 있다는 알림을 띄운 후,

내용 작성 텍스트 박스에 커서를 이동시킨다.

4.1. 수정 완료 버튼 클릭 후 저장 여부 확인에서 취소를 클릭하는 경우 :

4.1.1. 시스템은 수정 화면 상태에 머무른다.

7.1. 수정 완료 후 게시판 재 호출시 :

7.1.1. 시스템은 재 호출시 수정 시간을 기준으로 게시글을 재정렬한다.

[공지사항 수정]

목표 : 관리자는 작성한 공지사항에 접근하여 내용을 수정할 수 있다.

사전조건 : 로그인이 되어있어야 한다./ 관리자여야한다.

성공시나리오

1. 공지사항 글에 접근하여 수정 버튼을 클릭한다

2. 시스템은 기존 공지사항 내용을 읽어와 입력 화면에 표출한다.

3. 관리자는 원하는 형태로 내용을 수정하고 수정완료 버튼을 클릭한다.

4. 시스템은 수정사항을 저장할지 확인/취소 선택버튼을 출력한다.

5. 관리자는 확인 버튼을 클릭한다.

6. 시스템은 수정된 내용으로 공지사항 글을 대체한다.

7. 시스템은 공지사항 수정 완료 메세지를 보여주고 수정된 공지사항 접근 화면을 보여준다.

확장시나리오

3.1. 수정 화면에서 취소를 클릭하는 경우

3.1.1. 시스템은 해당 공지사항 접근 화면을 출력한다.

3.2. 제목이나 내용 중 누락된 값이 하나라도 존재하는 경우

3.2.1 작성하지 않은 부분이 있다는 알림을 띄운 후, 해당 텍스트 박스에 커서를 이동시킨다.

4.1. 수정 완료 버튼 클릭 후 저장 여부 확인에서 취소를 클릭하는 경우

4.1.1. 시스템은 수정 화면 상태에 머무른다.

[게시판 열람]

목표 : 사용자는 게시판을 열람하다.

(게시판은 SNS 스타일)

(게시판 열람은 게시판에 접속, 게시판에서 게시글 조회를 포함)

(이미지나 글내용이 게시판에 접속하면 바로 보인다고 가정 / 아래로 쭉 내리면 새 글들이 나옴)

(게시글은 최신순 / 20개씩 표기)

성공시나리오

1. 사용자는 메뉴바에서 게시판을 클릭한다.

2. 시스템은 게시판을 응답한다.

3. 사용자의 화면에 게시판이 표시된다.

4. 사용자는 원하는 게시글을 클릭한다

5. 시스템은 사용자가 클릭한 게시글을 응답한다.

(이미지가 여러개면 슬라이드 형식으로 보임)

6. 사용자 화면에 게시글이 표시된다.

(댓글들은 게시글 아래에 있음, 댓글 작성은 댓글들 아래 있음)

확장시나리오

4.1. 페이지를 아래로 끝까지 내림

4.1.1. 사용자는 페이지를 아래로 끝까지 내린다.

4.1.2. 시스템이 사용자 화면에 띄어줄 게시글들을 반환한다.

4.1.3. 사용자의 화면에 새 게시글들이 추가된다.

4.3. 조회기능 사용

4.3.1. 사용자는 검색창에 검색어를 입력, "검색" 버튼 클릭

4.3.2. 시스템은 검색어에 따라서 해당하는 게시글들을 응답

4.3.2. 사용자의 화면에 검색된 게시글들이 표시된다

6.1. 게시글에서 좋아요 or 즐겨찾기 클릭

6.1.1. 로그인 상태인지 확인

6.1.1.1. 로그인 상태가 아니면 로그인 페이지 띄움

6.1.1.2. user 화면에 로그인 페이지 나옴

6.1.2.1. 로그인 상태면 좋아요 수 증가, 사용자 계정의 좋아요한 글에 해당 글을 추가

[회원관리]

목적: 관리자로 로그인 한 유저만 회원관리가 가능하다. 회원관리에는 회원조회 및 회원삭제가 가능하다.

성공 시나리오

1. 관리자로 로그인 한 유저(이하 관리자)만 웹페이지의 회원관리 탭으로 접근 가능하다.

2. 회원관리 탭에서는 현재 데이터베이스에 저장되어 있는 회원들의 목록 열람 및 검색이 가능하다.

3. 회원 검색에서는 관리자가 입력한 값과 아이디나 이름이 일치하는 회원들의 목록을 열람할 수 있다.

4. 회원 목록은 아이디와 이름, 이메일을 확인할 수 있는 한 페이지에 30명의 회원 목록으로 구성된 서브페이지이며, 페이지 하단의 화살표를 통해서 이전 서브페이지, 다음 서브페이지로 이동하거나, 서브페이지 번호를 눌러 그 서브페이지로 바로 이동할 수 있다.

5. 관리자가 회원 목록에서 회원의 아이디를 클릭하면 그 회원의 정보 조회 탭으로 접근할 수 있다.

6. 회원 정보 조회 탭에서 관리자는 회원의 비밀번호를 제외한 데이터베이스에 등록된 모든 정보를 열람할 수 있다.

7. 회원 정보 조회에서 회원 삭제 버튼을 클릭하면 회원을 삭제할지 재차 확인하는 알림이 뜨고, 이에 확인 버튼을 클릭하는 경우에만 데이터베이스에 회원 정보 삭제 요청을 보낸다.

8. 데이터베이스에서 삭제가 성공하면 회원 삭제 성공 알림을 띄운 뒤에 회원 목록으로 돌아간다.

확장 시나리오

2.1. 데이터베이스에서 회원 조회를 실패했을 경우, 회원 조회 실패 알림을 띄운다.

8.1. 데이터베이스에서 삭제가 실패했을 경우, 회원 삭제 실패 알림을 띄운다.