|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **프로젝트 명: Festival Metro**  **테스트케이스(v1.0)**   |  |  | | --- | --- | |  | | | **테스트 케이스 ID:** 사용 사례 \_테마별검색 | **테스트 설계자:** 문예진, 이민주, 하유진, 한용진 | | **테스트 중요도(상/중/하):** 상 | **테스트 설계 날짜:** 2019. 11. 19. | | **모듈 이름:** 테마별 즐길거리 검색 | **테스트 실행자:** <이름> | | **테스트 제목:** 테마별 검색 검증 (테마 정보 제공) | **테스트 실행 날짜:** <Date> | | **설명:** 테마별 검색 페이지 테스트 |  | |  |  | |  |  | | **Pre-conditions:** 테마별 검색 페이지에 접속한 상태이다 | | | **Dependencies:** | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Step** | **테스트 단계** | **테스트 데이터** | **예상 결과** | **실행 결과** | **성공/실패** | **Notes** | | 1 | 테마 해시태그를 클릭 | Theme tag =[3] | 선택한 해시태그의 색이 변하고 화면에 해당테마의 명소와 축제들이 출력된다. |  |  |  |   **Post-conditions:**  Theme number에 근거하여 테마 정보가 검색된다. 검색 결과(장소 이름, 위치, 운영 시간, 이용 가격 등)는 화면에 출력되어 목록들을 확인할 수 있다.   |  |  | | --- | --- | |  | | | **테스트 케이스 ID:** 사용 사례 \_호선별검색 | **테스트 설계자:** 문예진, 이민주, 하유진, 한용진 | | **테스트 중요도(상/중/하):** 상 | **테스트 설계 날짜: 2019. 11. 19.** | | **모듈 이름:** 호선별 즐길거리 검색 | **테스트 실행자:** <이름> | | **테스트 제목:** 호선별 검색(호선번호 제공) | **테스트 실행 날짜:** <Date> | | **설명:** 호선별 검색 페이지 테스트 |  | |  |  | |  |  | | **Pre-conditions:** 호선별 검색 페이지에 접속한 상태이다 | | | **Dependencies:** | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Step** | **테스트 단계** | **테스트 데이터** | **예상 결과** | **실행 결과** | **성공/실패** | **Notes** | | 1 | 호선 해시태그를 클릭 | Line tag =[3] | 선택한 해시태그의 색이 변하고 화면에 해당 호선의 명소와 축제들이 출력된다. |  |  |  |   **Post-conditions:**  호선 번호에 따라 해당 노선도와 명소 정보가 담긴 지도와 목록들이 제공된다.   |  |  | | --- | --- | |  | | | **테스트 케이스 ID:** 사용 사례 \_코스추가 | **테스트 설계자:** 문예진, 이민주, 하유진, 한용진 | | **테스트 중요도(상/중/하):** 상 | **테스트 설계 날짜:** 2019.11.19 | | **모듈 이름:** 코스 추가 화면 | **테스트 실행자:** <이름> | | **테스트 제목:** 중복된 코스 검증 | **테스트 실행 날짜:** <Date> | | **설명:** 코스 추가 화면에서 새로운 코스를 추가할 때, 코스 목록이 저장되어 있는 데이터베이스를 검색해서 중복된 코스가 존재하는지 검증한다. 중복된 코스는 하나의 코스에 똑같은 구성물들이 존재하는 것이다. | | |  |  | |  |  | | **Pre-conditions: 관리자 계정으로 로그인한 상태이어야 한다** | | | **Dependencies:** | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Step** | **테스트 단계** | **테스트 데이터** | **예상 결과** | **실행 결과** | **성공/실패** | **Notes** | | 1 | 코스추가 버튼 클릭 |  | 코스 추가 화면으로 전환된다 |  |  |  | | 2 | 코스 테마 선택 | Theme =[3] | 화면에 텍스트로 내가 선택한 테마를 보여준다 |  |  |  | | 3 | 원하는 데이터(축제, 명소 등)을 코스에 담고 추가하기 버튼 클릭(정상적인 코스) | CourseList = [ 명소1, 명소2, 명소3, 명소4, 명소5 … ] | 코스가 생성되었다는 창을 띄워주고, 코스 정보를 데이터베이스에 저장한다 |  |  |  | | 4 | 원하는 데이터(축제, 명소 등)을 코스에 담고 추가하기 버튼 클릭(비정상적인 코스) | CourseList = [ 명소1, 명소2, 명소3, 명소4, 명소5 … ] | 이미 똑같은 코스가 존재한다는 경고창을 띄워 준다 |  |  |  |   **Post-conditions:**  관리자가 선택한 코스 리스트와 테마 정보가 데이터베이스에 기록된다. |

|  |  |
| --- | --- |
| **테스트 케이스 ID:** 사용 사례\_리뷰 작성 | **테스트 설계자:** 문예진, 이민주, 하유진, 한용진 |
| **테스트 중요도(상/중/하):** 상 | **테스트 설계 날짜:** 2019.11.19 |
| **모듈 이름:** 리뷰 작성 화면 | **테스트 실행자:** <이름> |
| **테스트 제목:**  리뷰 작성 | **테스트 실행 날짜:** <Date> |
| **설명:** 리뷰 작성 페이지 테스트이다. 로그인 한 상태에서 글을 작성하지 않고 ‘리뷰 등록’ 버튼을 누른 경우, 또는 로그인을 하지 않고 리뷰 작성을 누른 경우 리뷰가 등록되지 않도록 한다. | |
|  |  |
|  |  |
| **Pre-conditions:** 사용자 계정으로 로그인한 상태여야 한다. | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **테스트 단계** | **테스트 데이터** | **예상 결과** | **실행 결과** | **성공/실패** | **Notes** |
| 1 | 리뷰를 작성할 게시글 선택 | 없음 | 선택한 게시글 화면으로 전환한다. |  |  |  |
| 2 | 리뷰 작성 칸에 직접 글 입력 | review\_con = [리뷰 내용] | 리뷰 작성 칸에 리뷰 내용이 쓰여진다. |  |  |  |
| 3 | 평점 선택 | score = [5] | 사용자가 평점 1, 2, 3, 4, 5중에 하나를 선택하면 그 평점만큼 별의 색이 채워진다. |  |  |  |
| 4 | 정상적인 ‘리뷰 등록’버튼 클릭 | review\_con = [리뷰 내용]  User = [사용자] | ‘리뷰가 등록되었습니다’ 창을 띄워주고 데이터베이스에 저장한다. |  |  |  |
| 5 | 비정상적인 ‘리뷰 등록’버튼 클릭\_1 | review\_con =[]  User = [사용자] | ‘내용을 작성 해주세요’ 창을 띄워준다. |  |  | Review contents가 존재하지 않을 때 |

**Post-conditions:**

사용자의 리뷰가 데이터베이스에 저장된다.

|  |  |
| --- | --- |
|  | |
|  | |
| **테스트 케이스 ID:** 사용 사례\_리뷰 삭제 | **테스트 설계자:** 문예진, 이민주, 하유진, 한용진 |
| **테스트 중요도(상/중/하):** 중 | **테스트 설계 날짜:** 2019.11.19 |
| **모듈 이름:** 리뷰 삭제 화면 | **테스트 실행자:** <이름> |
| **테스트 제목:**  리뷰 삭제 | **테스트 실행 날짜:** <Date> |
| **설명:** 리뷰 삭제 페이지 테스트이다. 관리자 계정으로 로그인한 경우 적절하지 않은 리뷰를 확인하여 삭제한다. | |
|  |  |
|  |  |
| **Pre-conditions:** 관리자 계정으로 로그인한 상태여야 한다. | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **테스트 단계** | **테스트 데이터** | **예상 결과** | **실행 결과** | **성공/실패** | **Notes** |
| 1 | 리뷰 삭제 메뉴 클릭 | reviewList=[리뷰1, 리뷰2, 리뷰3 , ...] | 작성된 모든 리뷰 내용이 보여진다. |  |  |  |
| 2 | 삭제할 리뷰 ‘삭제’버튼 클릭 | reviewList=[리뷰1, 리뷰2, 리뷰3, ...] | ‘리뷰가 삭제되었습니다’ 창이 뜨고 데이터베이스에서 삭제 된다. |  |  |  |

**Post-conditions:**

관리자가 선택한 사용자의 리뷰가 데이터베이스에서 삭제된다.

|  |  |
| --- | --- |
| **테스트 케이스 ID:** 사용 사례 \_코스삭제 | **테스트 설계자:** 문예진, 이민주, 하유진, 한용진 |
| **테스트 중요도(상/중/하):** 중 | **테스트 설계 날짜:** 2019.11.19 |
| **모듈 이름:** 코스 목록 화면 | **테스트 실행자:** <이름> |
| **테스트 제목:** 코스 삭제 검증 | **테스트 실행 날짜:** <Date> |
| **설명:** 코스 목록 페이지에서 코스 삭제 기능 테스트 | |
|  |  |
|  |  |
| **Pre-conditions: 관리자 계정으로 로그인 된 상태이어야 한다.** | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **테스트 단계** | **테스트 데이터** | **예상 결과** | **실행 결과** | **성공/실패** | **Notes** |
| 1 | 코스 삭제 버튼을 누른다. |  | 정말 코스를 삭제할 것인지 확인하는 팝업창이 뜬다 |  |  |  |
| 2 | 팝업창에서 [예] 버튼을 누른다. | Input = [Yes],  # of courseList[] <= 10 | 코스가 삭제할 수 없다는 창을 띄워 준다 |  |  | 저장된 최소 추천코스의 개수는 10개. |
| 3 | 팝업창에서 [예] 버튼을 누른다. | Input = [Yes],  # of courseList[] > 10 | 코스가 삭제되었다는 창을 띄워 주고, 코스 목록 페이지로 전환한다. |  |  |  |
| 4 | 팝업창에서 [아니오] 버튼을 누른다. | Input = [ No ] | 코스 목록 페이지로 전환한다 |  |  |  |

**Post-conditions:**

관리자가 삭제하려고 선택한 코스 정보가 데이터베이스에서 삭제된다.

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **테스트 케이스 ID:** 사용 사례 \_쪽지보내기 | **테스트 설계자:** 문예진, 이민주, 하유진, 한용진 |
| **테스트 중요도(상/중/하):** 하 | **테스트 설계 날짜:** 2019.11.19 |
| **모듈 이름:** 쪽지 보내기 화면 | **테스트 실행자:** <이름> |
| **테스트 제목:** 쪽지가 정상적으로 보내지는지 검증 | **테스트 실행 날짜:** <date> |
| **설명:** 쪽지 보내는 페이지 테스트 |  |
|  |  |
|  |  |
| **Pre-conditions:** 사용자가 로그인한 상태여야 한다. | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **테스트 단계** | **테스트 데이터** | **예상 결과** | **실행 결과** | **성공/실패** | **Notes** |
| 1 | 리뷰작성자 옆에 존재하는 쪽지 보내기 버튼을 클릭 | Id=[DB에 존재하는 수신자의 id] | 쪽지를 정상적으로 보냈다는 메시지가 뜨며 쪽지 보내기 메인 페이지로 되돌아간다 |  |  |  |
| 2 | 리뷰작성자가 탈퇴한 경우 해당 리뷰작성자 옆에 존재하는 쪽지 보내기 버튼을 클릭 | Id=[DB에 존재하지 않는 수신자의 id] | 쪽지를 정상적으로 보낼 수 없다는 오류가 난다. |  |  |  |

**Post-conditions:**

수신자가 데이터베이스에 존재하는지 검증되면 쪽지의 내용이 데이터베이스에 저장되고 쪽지가 성공적으로 보내진다.