**2306008 김명훈 테스트계획서**

**주제: 개선된 학사 관리 시스템 테스트**

**제출자: 2306008 김명훈**

**학년/반: 3-A**

**팀명: EMAS(Enhanced Academic Management System)**

**프로젝트 참여자: 2106117 김태연, 2306007 김수혁, 2306008 김명훈**

**제출일자: 2025년 6월 16일**

|  |
| --- |
| 1. **개요**   **1.1 테스트 목적**  **1.2 테스트 범위**  **1.3 용어 정의 및 약어**  **1.4 테스트 기법**  **1.5 참고**   1. **기능 요구사항 테스트 계획**   **2.1 외부 인터페이스 테스트**  **2.2 하드웨어 인터페이스 테스트**  **2.3 통신 인터페이스 테스트**  **2.4 기능 요구사항 테스트**   1. **비기능 요구사항 및 기타 요구사항 테스트**   **3.1 성능 테스트 (반응 시간, 처리 소요 시간, 처리율)**  **3.2 하드웨어 요구사항 테스트 (메모리 용량, 통신 수용도)**   1. **테스트 환경 및 일정**   **5.1 테스트 환경 구성**  **5.2 테스트 일정 및 인력 배치**   1. **참고 자료** |

**1.개요**

**본 문서는 동서울 대학교 졸업과제 프로젝트인 개선된 학사 관리 시스템에 대한 테스트 계획서이다.  
해당 문서는 시스템의 기능적 및 비기능적 요구사항을 검증하기 위한 테스트 범위, 방법, 환경, 일정 등을 정의하여 전체 프로젝트에 완성도를 측정하기 위한 목적으로 작성하였다.**

**1.1 테스트 개요**

**본 테스트 계획서는 학사 관리 시스템에 대한 요구사항을 검증하기 위한 테스트 절차를 정의한다.  
각 테스트는 동적 테스트 방식(dynamic testing)으로 수행되며, 시스템의 기능적 요구사항과 비기능적 요구사항 모두를 테스트하는 것을 목적으로 한다.**

**1.2 테스트 범위**

**본 테스트 계획서의 테스트 범위는 기존 요구사항 명세서에 명시된, 인터페이스 요구사항, 통신 요구사항, 사용자 하드웨어 요구사항, 기능 요구사항, 성능 요구사항 등 총 35개의 요구사항에 대한 테스트 케이스를 작성한다.**

**단 요구사항 명세서에서 프로젝트 내 비용 관련, 인터페이스 직관성, 최대기능 점수 요구사항 등 테스트가 불가능한 요구사항에 경우 테스트 케이스에서 제외된다.  
또한 향후 프로젝트 진행 중 추가되는 요구사항 또한 테스트 범위에 포함된다.**

**1.3 용어 정의 및 약어**

**정의**

**수정: 데이터베이스 안의 데이터를 변경하는 행위(UPDATE)를 의미한다**

**삭제: 데이터베이스 안의 데이터를 삭제하는 행위(DELETE)를 의미한다.**

**학생: 시스템을 이용하는 학생 사용자(액터)를 의미한다.**

**교수: 시스템을 이용하는 교수 사용자(액터)를 의미한다.**

**조교: 시스템을 이용하는 조교 사용자(액터)를 의미한다.**

**직원: 시스템을 이용하는 직원 사용자(액터)를 의미한다.**

**반응시간: 버튼 등 웹사이트 요소를 클릭했을 때 응답이 시작되기까지의 시간.**

**처리소요시간: 웹사이트에 접근하여 데이터를 불러오는 데 소요되는 전체 시간.**

**안정상태: 하드웨어에 과도한 부하가 없는 정상 동작 상태를 의미한다.**

**약어**

**인터페이스: 사용자 인터페이스(UI)를 지칭할 때 ‘인터페이스’로 표현한다.**

**하드웨어 인터페이스, 통신 인터페이스: 각각 그대로 ‘하드웨어 인터페이스’, ‘통신 인터페이스’로 사용한다.**

**사용자: 학생, 교수, 조교, 직원을 모두 포함하는 전체 액터를 총칭한다.**

**교직원: 학생을 제외한 교수, 조교, 직원 등 총칭한다.**

**1.4 테스트 케이스 작성 기법**

**계획서에서 작성된 모든 테스트 케이스는 전부 원인-결과 그래프 기법을 통해 도출한다.**

**1.5 참고  
본 테스트 계획서에서 각 테스트는 글로 테스트에 사용된 요구사항 내역, 테스트 목적과 테스트 절차, 등을 명확하게 기술하며, 여러 분기 및 경우의 수가 많은 테스트의 경우에는 테스트 케이스 표를 함께 작성하여 오해의 소지가 없도록 한다.**

**2. 기능 요구사항 테스트 계획**

**2.1 인터페이스 요구사항 테스트**

**TC-001 내부 웹사이트 연동 확인**

* **요구사항 내역:  
  학사관리 시스템 포털사이트에서 동서울 대학교 내부에 다른 웹사이트와 연결되어야 한다.**
* **목적: 학사 관리 시스템에서 동서울대학교 내의 타 웹사이트(포털, 도서관, 메일 등)로의 연동이 정상적으로 되는지 확인한다.**
* **테스트 대상:   
  동서울 대학교 내부 웹사이트 연결**
* **테스트 절차:**
  1. **학생 계정으로 로그인한다.**
  2. **웹사이트 상단 또는 메뉴에서 “도서관” 또는 학사홈페이지 클릭**
  3. **버튼을 클릭 시 페이지로 이동 시 되는지 확인한다.**

**4-1. 도서관 클릭했을 경우 URL** https://dlibrary.du.ac.kr/ **(도서관 URL)인지 확인한다**

**4-2 학과 홈페이지 클릭했을 URL이 https://www.du.ac.kr/main.do 인지 확인한다.**

* **기대 결과:**
  1. **모든 연동 메뉴 클릭 시 대상 웹사이트로 정상적으로 이동**
  2. **연동 오류, 인증 오류, 접근 불가 등의 문제없음**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **입력,출력/테스트 케이스** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **입력: 도서관 버튼 클릭** | **T** | **T** | **T** | **F** | **F** | **F** | **T** | **F** |
| **입력: 학과홈페이지 클릭** | **F** | **F** | **F** | **T** | **T** | **T** | **F** | **T** |
| **출력: 도서관 이동** | **T** | **F** | **F** | **F** | **F** | **F** | **F** | **T** |
| **출력: 학과 홈페이지 이동** | **F** | **F** | **F** | **T** | **F** | **F** | **T** | **F** |
| **출력: 이동 및 통신오류** | **F** | **F** | **T** | **F** | **F** | **T** | **F** | **F** |
| **출력: 학과, 도서관이 아닌 사이트로 이동** | **F** | **T** | **F** | **F** | **T** | **F** | **F** | **F** |

**테스트 케이스**

**각 테스트 케이스 설명:  
케이스 1: 도서관 클릭 시 도서관 사이트로 이동**

**케이스 2: 도서관 클릭 시 학과, 도서관이 아닌 사이트로 이동**

**케이스 3: 도서관 클릭 시 이동 통신 및 이동 오류(404 NOT FOUND)**

**케이스 4: 도서관 클릭 시 학과 홈페이지로 이동**

**케이스 5: 학과 홈페이지 클릭 시 학과 홈페이지로 이동**

**케이스 6 학과 홈페이지 클릭 시 로 학과, 도서관이 아닌 사이트로 이동**

**케이스 7: 학과 홈페이지 클릭 시 이동 통신 및 이동 오류(404 NOT FOUND)**

**케이스 8: 학과 홈페이지 클릭 시 도서관으로 이동**

**테스트 케이스 전제 조건:  
사용자는 한 번에 하나의 버튼만 클릭할 수 있으며,  
각 테스트 케이스는 독립적으로 실행되어야 한다.**

**TC-002 비로그인 사용자 접근 제한 확인**

* **요구사항 내역:  
  메인 포털 사이트 및 모든 사이트들은 로그인한 학생 및 교직원들만 이용할 수 있어야 한다.**
* **목적:  
  로그인을 하지 않은 사용자가 포털사이트 및 내부 웹사이트(도서관, 메일 등)에 접근할 수 없는지 확인한다.**
* **테스트 대상:  
  비로그인 상태의 웹 브라우저**
* **테스트 절차:**
  1. **학사관리 시스템 포털사이트에 로그인하지 않은 상태로 웹사이트에 접속한다.**
  2. **포털 사이트의 내부 내용(메뉴, 데이터등)이 보이지 않는지 확인한다.**
  3. **수강신청사이트, 통합처리메뉴 등 다른 내부 시스템에도 로그인 없이 접근을 시도한다.**
  4. **접근 시도 시 제대로 접근을 차단하는지 확인한다.**
* **기대 결과:**
  1. **로그인하지 않은 사용자가 접근을 시도할 경우, 로그인 페이지로 자동 리다이렉트 되거나 접근 불가 안내 메시지가 표시되어야 한다.**
  2. **로그인하지 않고는 어떠한 내부 시스템(포털, 도서관, 메일 등)도 이용할 수 없어야 한다.**

**TC-003 학생용 포털 사이트 및 통합정보 메뉴 이동 처리 확인**

* **요구사항 내역:  
  사용자 신분에 따라 메인 포털사이트 및 통합정보 사이트에서 사용할 수 있는 기능이 달라야 한다.**
* **목적:  
  학생 계정으로 포털사이트 로그인 시, 학생용 포털 사이트로 정상 이동하고, 통합정보 메뉴에서도 학생 전용 기능(예: 휴학신청 등)이 정상 노출되는지 확인한다.**
* **테스트 대상:  
  포털사이트 로그인 폼, 포털사이트 메인, 통합정보 메뉴**
* **테스트 데이터:**
  + **학생 계정 ID: 11101001 PW: 1234 역할: 학생**
* **테스트 절차:**
  1. **학사관리 시스템 포털사이트에 테스트 학생 계정(11101001/1234)으로 로그인한다.**
  2. **로그인 후 학생용 포털사이트로 이동되는지 확인한다.**
  3. **학생용 포털사이트 중앙에 출석조회, 성적조회 등 학생 관련 메뉴가 노출되는지 확인한다.**
  4. **통합정보 메뉴로 이동하여, 학생 전용 메뉴(예: 휴학신청)가 존재하는지 확인한다.**
* **기대 결과:**
  1. **학생 계정으로 로그인 시, 학생용 포털사이트로 정상적으로 이동해야 한다.**
  2. **포털사이트에서 학생 관련 메뉴가 중앙에 배치되어 있어야 한다.**
  3. **통합정보 메뉴로 이동 시, 학생 전용 메뉴(휴학신청)가 노출되어야 한다.**

**TC-004 교수용 포털 사이트 및 통합정보 메뉴 이동 처리 확인**

* **요구사항 내역:**  
  **사용자 신분에 따라 메인 포털사이트 및 통합정보 사이트에서 사용할 수 있는 기능이 달라야 한다.**
* **목적:  
  교수 계정으로 포털사이트 로그인 시 교수용 포털 사이트로 정상 이동하는 지 확인한다.**
* **테스트 대상:  
  포털사이트 로그인 폼**
* **테스트 데이터**
* **교수 계정  
  ID: 10115 PW: 1234 역할: 교수**
* **테스트 절차:**
  1. **학사관리 시스템 로그인 사이트에서 테스트 교수계정(10115 / 1234)으로 로그인을 진행한다.**
  2. **로그인 후 교수용 포털사이트로 정상적으로 이동되는지 확인한다.**
  3. **교수용 포털사이트 중앙에 강좌개설, 성적입력 등 교수 관련 메뉴가 올바르게 배치되어 있는지 확인한다.**
  4. **통합정보 메뉴로 이동하여, 교수** **전용 메뉴(예: 수업관리)가 존재하는지 확인한다.**
* **기대 결과:**
  1. **교수 계정으로 로그인 시, 교수용 포털사이트로 정상적으로 이동해야 한다.**
  2. **포털사이트에서 교수 관련 메뉴(강좌개설, 성적입력 등)가 중앙에 배치되어 있어야 한다.**
  3. **통합정보 메뉴로 이동 시, 교수 전용 메뉴(수업관리 등)만 노출되어야 한다.**

**TC-005 직원용 포털 사이트 및 통합정보 메뉴 이동 처리 확인**

* **요구사항 내역:  
  사용자 신분에 따라 메인 포털사이트 및 통합정보 사이트에서 사용할 수 있는 기능이 달라야 한다.**
* **목적:  
  직원 계정으로 포털사이트 로그인 시, 직원용 포털 사이트로 정상 이동하고,  
  통합정보 메뉴에서도 직원 전용 기능(예: 휴학등록, 수강신청 마감기간처리 등)이 정상 노출되는지 확인한다.**
* **테스트 대상:  
  포털사이트 로그인 폼**
* **테스트 데이터**
* **직원 계정  
  ID: 01115 PW: 1234 역할: 직원**
* **테스트 절차:**
  1. **학사관리 시스템 포털사이트에 테스트 직원 계정(01115/1234)으로 로그인을 진행한다.**
  2. **로그인 후 직원용 포털사이트로 정상적으로 이동되는지 확인한다.**
  3. **직원용 포털사이트에 들어갈 내용인 수강신청 마감기간처리, 휴학처리 등이 제대로 중앙에 배치되어 있는지 확인한다.**
  4. **통합정보 메뉴로 이동하여 직원 메뉴에 휴학등록 메뉴가 존재하는지 확인한다.**
* **기대 결과:**
  + **직원 계정으로 로그인 시, 직원용 포털사이트로 정상적으로 이동해야 한다.**
  + **포털사이트에서 직원 관련 메뉴가 중앙에 배치되어 있어야 한다.**
  + **통합정보 메뉴로 이동 시, 직원 전용 메뉴(강좌개설, 휴학등록 등)가 노출되어야 한다.**

**TC-006 조교용 포털 사이트 및 통합정보 메뉴 이동 처리 확인**

* **요구사항 내역:  
  사용자 신분에 따라 메인 포털사이트 및 통합정보 사이트에서 사용할 수 있는 기능이 달라야 한다**
* **목적:  
  조교계정으로 포털사이트 로그인 시 조교용 포털 사이트로 이동하는지 확인한다.**
* **테스트 대상:   
  포털사이트 로그인 폼**
* **테스트 데이터**
* **조교 계정  
  ID: 11115 PW: 1234 역할: 조교**
* **테스트 절차:**
  1. **학사관리 시스템 포털사이트에 테스트 조교 계정으로 로그인을 진행한다.**
  2. **조교용 포털사이트에 들어갈 내용인 학생 수강신청 번경이 제대로 중앙에 배치되어 있는지 확인한다.**
* **기대 결과:**
  + **조교 계정으로 로그인 시, 조교용 포털사이트로 정상적으로 이동해야 한다.**
  + **포털사이트에서 조교 관련 메뉴(학생 수강신청 변경 등)가 중앙에 배치되어 있어야 한다.**

**TC-006 뷰어 포함**

* **요구사항 내역:  
  jpf,pdf 등 같은 외부 파일을 열 수 있는 뷰어를 통해 파일들을 볼 수 있어야 한다.**
* **목적:  
  게시판 내부에 pdf,word파일을 볼 수 있어야 한다.**
* **테스트 대상:   
  게시판**
* **테스트 데이터  
  test1.pdf,test2.jpg**
* **학생 계정  
  ID: 1101001 PW: 1234**
* **테스트 절차:**
  1. **포털사이트에서 게시판을 탭으로 들어간다.**
  2. **게시판 파일을 클릭한다.**

**3-1. jpg파일 클릭하여 업로드 한다.**

**3-2. pdf파일을 클릭하여 업로드를 진행한다.**

**4. 등록 버튼을 클릭하여 게시판에 업로드를 한다.**

**5. 등록된 게시글에 업로드한 파일들이 출력상태를 확인한다.**

* **기대 결과:**
  + **게시판 내부 뷰어들이 정상적으로 작동된다는 것을 보여준다.**

**테스트 케이스**

| **입력,출력/테스트 케이스** | **1** | **2** | **3** | **4** | **5** | **6** |
| --- | --- | --- | --- | --- | --- | --- |
| **입력: JPG 파일 업로드** | **T** | **F** | **T** | **T** | **F** | **F** |
| **입력: PDF 파일 업로드** | **F** | **T** | **F** | **F** | **T** | **T** |
| **출력: JPG 파일 출력** | **T** | **F** | **F** | **F** | **F** | **F** |
| **출력: PDF 파일 출력** | **F** | **T** | **F** | **F** | **F** | **F** |
| **출력: 파일 출력 오류** | **F** | **F** | **T** | **F** | **T** | **F** |
| **출력: 파일 업로드 오류** | **F** | **F** | **F** | **T** | **F** | **T** |

**테스트 케이스 설명**

**케이스 1: JPG 업로드 시 JPG출력**

**케이스 2: PDF 업로드 시 PDF출력**

**케이스 3: JPG 업로드 시 파일 출력 오류**

**케이스 4: JPG 업로드 시 파일 업로드 오류**

**케이스 5: PDF 업로드 시 파일 출력 오류**

**케이스 6 PDF 업로드 시 파일 업로드 오류**

**2.2 기능 요구사항 테스트**

**TC-006 강의 관리**

* **요구사항 내역:  
  교수는 자신이 등록한 강의를 등록, 수정, 조회할 수 있어야 한다**
* **목적:  
  교수가 강의 업로드가 정상적으로 가능한지 확인한다.**
* **테스트 대상: 교수 액터 시간표 강좌개설 유스케이스**
* **테스트 데이터**

**테스트 데이터1:  
강의명: c 교수명: 홍길동 학점:3 수업시간 5,6,7교시 수업요일: 수요일**

**테스트 데이터2:  
강의명:c 교수명: 홍길동 학점:4 수업시간 5,6,7,8교시 수업요일: 수요일**

* **교수 계정  
  ID: 10115 PW: 1234 역할: 교수**
* **학생 계정:**

**ID: 1101001 PW: 1234**

* **테스트 절차:**
  1. **교수 계정으로 통합 정보에서 수업 메뉴 클릭**
  2. **강의 개설 탭 클릭**

**3. 강의 등록 클릭**

**4. 테스트 데이터1대로 강의정보를 입력 및 등록**

**5. 수업 탭에 자신 강좌 조회 클릭**

**6. 등록된 강의가 강의 목록에서 정확히 조회되는 지 확인한다.**

**7. 강의 정보(강의명, 교수명, 학점, 수업시간, 요일)가 입력값과 일치하는지**

**확인한다**

**8. 테스트 데이터2 대로 강의 정보 입력**

**9. “학점과 수업시간이 일치하지 않습니다” 출력**

**9. 학생계정으로 url을 이용하여 강의 개설 탭 접근**

**10. “해당 기능을 사용” 메시지 출력”**

* **기대 결과:**

1. **강의가 정상적으로 등록되고, “내 강좌” 목록에서 정확히 조회된다.**
2. **등록된 정보(강의명, 교수명, 학점, 시간, 요일)가 모두 입력한 값과 동일하다.**

**테스트 케이스**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **입력,출력/테스트케이스** | **1** | **2** | **3** | **4** |
| **입력: 교수계정** | **T** | **T** | **F** | **T** |
| **입력: 학생계정** | **F** | **F** | **T** | **F** |
| **입력: 테스트 데이터1** | **T** | **F** | **-** | **F** |
| **입력: 테스트 데이터2** | **F** | **T** | **-** | **F** |
| **출력: 강의 개설 탭 접근 여부** | **T** | **T** | **F** | **T** |
| **출력: 강의 등록 성공** | **T** | **F** | **-** | **F** |
| **출력: 강의 등록 실패** | **F** | **T** | **-** | **T** |

**테스트 케이스 설명:**

**테스트 케이스 1: 교수 계정으로 테스트 데이터1로 강의 등록 시 등록 성공**

**테스트 케이스 2: 교수 계정으로 테스트 데이터2로 강의 등록 시 등록 실패**

**테스트 케이스 3 학생 계정으로 강의 개설 탭 접근 시 실패**

**테스트 케이스 4 교수 계정으로 데이터 없이 강의 동록 시 등록 실패**

**TC-006 수강자 명단 조회**

* **요구사항 내역:  
  교수는 자신이 등록한 강의를 등록, 수정, 조회할 수 있어야 한다**
* **목적:  
  교수가 수강자 명단을 정상적으로 조회 가능하게 한다.**
* **테스트 대상:   
  교수 액터 시간표 수강자 명단 조회 유스케이스**
* **테스트 데이터**

**교수 계정  
ID: 10115 PW: 1234 역할: 교수**

**학생1 정보:  
이름: 홍길동 학번:2010116**

**학생2 정보:  
이름: 유관순 학번:2311117**

**학생4 정보:  
이름: 이순신 학번:2110122**

* **테스트 절차:**
  1. **통합 정보에서 수업 메뉴 클릭**
  2. **수강자 명단 탭 클릭**
  3. **수업 선택**

**3. 수업의 학생 명단이 테스트 데이터와 일치하는지 확인**

* **기대 결과:**
  1. **교수자가 수강자 명단을 조회할 때, 실제로 수강 신청한 학생들의 정보(이름, 학번 등)가 시스템에 정확히 반영되고 일치함을 확인할 수 있다.**
  2. **수강자 명단에서 보여지는 학생1, 학생2, 학생3의 학번 및 이름이**

**테스트데이터의 것과 동일할 것을 기대됨**

**TC-006 성적 관리**

* **요구사항 내역:  
  교수가 담당하는 강의에 대해 수강한 학생들의 성적을 입력하고, 기존 성적을 수정하거나 조회할 수 있어야 한다**
* **목적:  
  교수가 성적을 정상적으로 입력을 가능하게 한다.**
* **테스트 대상:   
  교수 액터 시간표 성적관리 유스케이스**
* **테스트 데이터**

**교수 계정: ID:   
10115 PW: 1234 역할: 교수**

**학생1 정보: 이름:   
홍길동 학번:2010116 입력 성적 ‘F’**

**학생2 정보: 이름:   
유관순 학번:2311117 입력 성적 ‘G’**

**수업 데이터:   
강의명: c 교수명: 홍길동 학점:3 수업시간 5,6,7교시 수업요일: 수요일**

**테스트 절차:**

* 1. **통합 정보에서 수업 메뉴 클릭**
  2. **성적관리 탭 클릭**
  3. **수업 c 선택**

**4. 수업의 테스트 데이터의 학생1의 성적 입력**

**5. “성적입력 완료되었습니다.” 출력**

**5. 수업의 테스트 데이터의 학생2의 성적 입력**

**6. “잘못된 성적입니다.” 메시지 출력 후 입력 취소**

**7. 성적 조회 탭 클릭**

**8. 저장된 학생의 성적이 입력 값과 일치하는지 확인한다.**

* **기대 결과:**
  1. **정상 입력 시(학생1): 입력한 성적이 정확히 저장/조회됨**
  2. **비정상 입력 시(학생2): 오류 메시지가 출력되고, 입력이 저장되지 않음**
  3. **입력 실패한 학생2 외에, 이미 저장된 학생1의 성적은 정상적으로 유지됨**

|  |  |  |
| --- | --- | --- |
| **테스트케이스** | **1** | **2** |
| **입력값** | **F** | **G** |
| **예상 결과** | **승인** | **반려** |
| **실제 결과** | **미정** | **미정** |

**TC-006 출석 관리**

* **요구사항 내역:  
  교수는 자신이 담당하는 강의에 대해 수강 중인 학생들의 출석 정보를 입력하고, 수정하거나 조회할 수 있어야한다.**
* **목적:  
  교수가 출석을 정상적으로 입력을 가능하게 한다.**
* **테스트 대상:  
   교수 액터 시간표 출석관리 유스케이스**
* **테스트 데이터**

**교수 계정  
ID: 10115 PW: 1234 역할: 교수**

**학생1 정보:  
이름: 홍길동 학번:2010116 결석시간 입력 3**

**학생2 정보:  
이름: 유관순 학번:2311117 결석시간 입력 4**

**수업 데이터:   
강의명: c 교수명: 홍길동 학점:3 수업시간 5,6,7교시 수업요일: 수요일**

* **테스트 절차:**
  1. **통합 정보에서 수업 메뉴 클릭**
  2. **출석관리 탭 클릭**
  3. **수업 c 선택**

**4. 학생1의 결석시간 ‘3’ 입력 후 저장 시도**

**5. “출석입력이 완료되었습니다.” 출력**

**5. 학생2의 결석시간 ‘4’ 입력 후 저장 시도**

**6. “잘못된 성적입니다.” 메시지 출력 후 입력 취소**

**7. 출석 조회 탭 클릭**

**8. 학생1의 결석시간이 ‘3’으로 정확히 저장되어 있는지 확인**

**9. 학생2의 결석시간이 저장되지 않았는지 확인**

**기대 결과:**

* 1. **정상 입력 시(학생1): 입력한 출석이 정확히 저장된다.**
  2. **비정상 입력 시(학생2): 오류 메시지가 출력되고, 입력이 저장되지 않음**
  3. **입력 실패한 후에도 학생2 외에, 이미 저장된 학생1의 출석은 정상적으로 유지됨**

|  |  |  |
| --- | --- | --- |
| **출력/테스트 케이스** | **1** | **2** |
| **입력값** | **3** | **4** |
| **예상 결과** | **승인** | **반려** |
| **실제 결과** | **미정** | **미정** |

**TC-006 과제 입력**

* **요구사항 내역:  
  교수는 자신이 담당하는 강의에 대해 과제를 등록하고, 등록된 과제의 내용을 수정하거나 마감 기한 내 제출된 학생 과제를 평가할 수있어야 한다**
* **목적:  
  교수가 과제를 정상적으로 생성할 수 있는 지 확인한다.**
* **테스트 대상:  
   교수 액터 시간표 과제관리 유스케이스**
* **테스트 데이터**

**교수 계정  
ID: 10115 PW: 1234 역할: 교수**

**과제1:  
이름: C에서의 입출력, 내용: 1234, 파일 결과.PNG**

**수업 데이터:   
강의명: c 교수명: 홍길동 학점:3 수업시간 5,6,7교시 수업요일: 수요일**

* **테스트 절차:**
  1. **통합 정보에서 수업 메뉴 클릭**
  2. **과제 생성 탭 클릭**
  3. **수업 c 선택**

**4. 과제 등록 클릭**

**5. 과제 데이터 기반으로 입력 진행**

**6. 등록 버튼 클릭**

**7. 과제가 데이터와 동일하게 생성되었는 지 확인한다.**

**8. 파일이 정상적으로 출력되는 지 확인한다.**

* **기대 결과:**
  1. **과제 데이터와 동일한 과제가 생성된다.**

**2. 첨부 파일로 작성된 사진이 깨짐 없이 정상적으로 출력된다**

**TC-006 과제 제출 확인**

* **요구사항 내역:  
  교수는 자신이 담당하는 강의에 대해 과제를 등록하고, 등록된 과제의 내용을 수정하거나 마감 기한 내 제출된 학생 과제를 평가할 수있어야 한다**
* **목적:  
  교수가 제출된 과제를 정상적으로 확인가능 한 지 테스트한다.**
* **테스트 대상:  
   교수 액터 시간표 과제관리 유스케이스**
* **테스트 데이터**

**교수 계정  
ID: 10115 PW: 1234 역할: 교수**

**과제 제출 데이터:  
이름: 2010116 홍길동 과제 제출합니다. , 내용124 파일 과제결과.PNG**

**수업 데이터:   
강의명: c 교수명: 홍길동 학점:3 수업시간 5,6,7교시 수업요일: 수요일**

* **테스트 절차:**
  1. **통합 정보에서 수업 메뉴 클릭**
  2. **과제 결과 탭 클릭**
  3. **수업 c 선택**

**4. 2010116 홍길동 과제 제출합니다 클릭**

**5. 저장된 데이터와 출력된 데이터가 동일한지 확인한다.**

* **기대 결과:**
  1. **과제제출데이터와 동일한 내용이 출력된다.**
  2. **첨부 파일로 작성된 사진이 깨짐 없이 정상적으로 출력된다.**

**TC-006 수강 변경**

* **요구사항 내역:  
  조교는 특정 학생의 수강 정보를 조회할 수 있다.**

**또한 조교는 학생의 기존 신청 과목을 다른 과목으로 변경할 수 있어야 한다.**

* **목적:  
  조교가 학생의 수강정보를 조회 및 변경 시 데이터가 무결한지 확인한다.**
* **테스트 대상:  
   조교 액터 수강변경 유스 케이스**
* **테스트 데이터**
* **조교 계정  
  ID: 11115 PW: 1234 역할: 조교**
* **학생1 정보:  
  이름: 홍길동 학번:2010116**

**수강 데이터:  
삭제 데이터: 수업명: c 학점:3 수업시간: 3교시~6교시 수업요일: 월요일**

**입력 데이터: 수업명: python 학점:3 수업시간:2교시~5교시 수업요일: 금요일**

* **테스트 절차:**
  1. **통합 정보에서 수강변경 메뉴 클릭**
  2. **테스트 데이터 학생 학번 입력**
  3. **출력된 목록 수강 데이터에서 수업명 c인 수업 삭제 진행**

**4. 새로고침 클릭 후 수강 목록에서 결과 확인**

**5. 수업검색에서 python 조회**

**6. python 강의 신청**

**7.새로고침 후 학생의 수강 목록에 python이 추가되었는지 확인한다**

**8. dbms에서 테스트 학생에 수강 데이터 조회하여 일치하는지 확인**

* **기대 결과:**
  1. **삭제 시 수강목록에서 데이터가 보이지 않고 삭제된다.**

**2. 신청한 수업이 제대로 수강목록에 추가되는지 확인한다.**

**3. 삭제 및 입력이 실제 데이터에 반영된다.**

**TC-006 수강 신청**

* **요구사항 내역:  
  학생들은 수강신청사이트에서 신청가능 한 수업들을 보고 신청 및 신청취소를 할 수 있어야 한다.**

**이 때 수업 요일 및 시간이 겹칠 시 수강이 안되야 한다.**

* **목적:  
  학생이 수강 신청사이트에서 신청 및 신청취소를 진행할 시 결합요소들이 존재하는지 체크한다.**
* **테스트 대상:  
   학생액터 수강신청 유스케이스**
* **테스트 데이터:**

**학생 계정 ID: 11101001 PW: 1234 역할: 학생**

**수강 데이터:  
취소데이터: 수업명: c 학점:3 수업시간: 3교시~6교시 수업요일: 월요일**

**신청데이터: 수업명: java 학점:3 수업시간:2교시~5교시 수업요일: 금요일**

**신청데이터: 수업명: 예술과 미학 수업시간:2교시~4교시 수업요일: 금요일**

* **테스트 절차:**
  1. **통합 정보에서 수강신청 메뉴 클릭**
  2. **수업목록 java 신청 클릭**

**3. 새로고침 클릭 후 신청 목록에서 java수업 존재 확인**

**4. 신청 목록에서 c수업 신청 취소 클릭**

**5. 새로고침 후 신청 목록에서 c가 삭제되었는지 확인**

**6. 수업목록 예술과 미학 수업 클릭 신청**

**7. “수업시간이 겹쳐 신청할 수 없습니다.” 출력확인**

* **기대 결과:**
  1. **신청 시 신청목록에서 이를 반영하는지 확인한다.**

**2. 신청취소 시 신청목록에서 이를 반영하는지 확인한다.**

**3. 수업요일 및 수업시간이 겹치는 수업이 신청이 거절되는지 확인한다.**

**TC-006 시간표**

* **요구사항 내역:  
  학생들은 자신이 수강한 수업들의 시간을 알 수 있어야 한다.**

**이때 수업들의 시간을 시간표 형태로 볼 수 있어야 한다.**

* **목적:  
  학생이 자신이 수강한 수업들의 시간표가 정상적으로 출력되는지 확인한다.**
* **테스트 대상:  
   학생 액터 시간표 유스케이스**
* **테스트 데이터:**

**학생1 계정 ID: 11101001 PW: 1234 역할: 학생**

**학생1 수강 데이터:**

**수업명: java 학점:3 수업시간:2교시~5교시 수업요일: 금요일**

**수업명: c++ 수업시간5교시~7교시 수업요일: 금요일**

**학생2 계정 ID:112010002 PW:1234 역할: 학생**

**학생2 수강 데이터:**

**없음**

**테스트 절차:**

* 1. **학생1으로 통합 정보에서 시간표 메뉴 클릭**
  2. **시간표 조회 클릭**

**3. 현재 수강데이터와 맞게 시간표과 출력되었는지 확인 진행한다.**

**4. 학생2으로 통합 정보에서 시간표 메뉴 클릭**

**5. 시간표 조회 클릭**

**6. “현재 수강한 수업이 없습니다. 출력되는 지 확인”**

* **기대 결과:**
  1. **수강 데이터가 존재할 경우 정상적으로 수강 데이터에 맞춰 시간표가 출력된다.**

**2. 수강한 수업이 없을 경우 시간표 출력이 거절된다.**

**TC-006 휴학 신청**

* **요구사항 내역:  
  학생들은 수강 포털에서 휴학신청을 진행할 수 있어야 한다.**

**이때 휴학 기록을 조회하여 일반 휴학기록이 있을 시 자동으로 거절되어야 한다.**

* **목적:  
  학생이 많이 신청하는 일반 휴학유형으로 휴학신청 진행할 시 정상적으로 신청된 데이터가 조회되는지 확인한다.**
* **테스트 대상:  
   학생 액터 휴학신청 유스케이스**
* **테스트 데이터:**

**학생1 계정 ID: 11101001 PW: 1234 역할: 학생**

**학생1 휴학기록 데이터:**

**데이터 없음**

**학생1 휴학신청 데이터:**

**휴학유형: 일반 휴학 휴학사유: 진로 관련 휴학일자:2002-11-11**

**학생2 계정 ID:112010002 PW:1234 역할: 학생**

**학생2 휴학 데이터:**

**학번:112011002 휴학유형: 일반 휴학 휴학사유:기타 휴학일자:2001-11-11**

**테스트 절차:**

* 1. **학생1으로 통합 정보에서 휴학신청 메뉴 클릭**
  2. **휴학 신청서 작성 클릭**

**3. 휴학 신청 데이터에 맞게 신청서 입력 후 등록 버튼 클릭**

**4. 휴학 신청 기록 클릭하여 신청서가 있는지 확인**

**5. 학생2로 휴학신청 메뉴 클릭**

**6. 신청 유형 일반 휴학으로 신청서 입력 후 등록 버튼 클릭**

**7. “일반 휴학을 이미 했었기에 일반 휴학을 진행할 수 없습니다.” 출력되는지 확인**

* **기대 결과:**
  1. **일반 휴학 기록이 없을 경우 정상적으로 일반 휴학이 진행된다.**

**2. 일반 휴학 기록이 있을 경우 일반 휴학이 거절된다.**

**TC-006 휴학처리**

* **요구사항 내역:  
  직원들은 학생들이 신청한 휴학 신청을 반려하거나 승인할 수 있어야 한다.**
* **목적:  
  휴학처리가 정상적으로 되는 지 확인한다.**
* **테스트 대상:  
  직원 액터 유스케이스 휴학처리**

**테스트 데이터:  
학생1 휴학신청 데이터:**

**휴학유형: 일반 휴학 휴학사유: 진로 관련 휴학일자:2002-11-11**

**학생2 휴학신청 데이터:**

**휴학유형: 일반 휴학 휴학사유: 사생활 휴학일자:2002-12-31**

**직원 계정**

**ID: 01115 PW: 1234 역할: 직원**

**학생1 계정:**

**ID: 1101001 PW: 1234**

**학생2 계정:**

**ID: 1102002 PW: 456777**

* **테스트 절차:**
  1. **직원 계정으로 통합 정보에서 휴학처리 클릭**

**2. 테스트 데이터에 학생1 휴학신청을 승인 버튼 클릭**

**3. 학생1의 휴학 신청의 승인 여부가 승인으로 바뀌었는지 확인**

**4. 테스트 데이터에 학생2 휴학신청을 반려 버튼 클릭**

**5. 학생1의 휴학 신청의 승인 여부가 반려로 바뀌었는지 확인**

* **기대 효과**

**1. 휴학처리에서 승인이 정상적으로 작동된다.**

**2. 휴학처리에서 반려가 정상적으로 작동된다**

**TC-006 자신정보 조회**

* **요구사항 내역:  
  자신의 개인정보(이름, 나이, 주소, 성별, 학년, 학과, 전화번호, 등)을 조회할 수 있으며 자주 변경되는 정보(주소, 계좌번호)들은 수정할 수 있도록 해야 한다.**
* **목적:  
  사용자가 자신 정보를 조회가 원하는 하게 이루어지는 확인하고 수정사항이 제대로 반영되는지 확인한다.**
* **테스트 대상:  
   사용자 액터 정보 조회 수정 유스케이스**
* **테스트 데이터:**

**학생1 계정  
ID: 11101001 PW: 1234 역할: 학생**

**이름: 홍길동 학번: 11101001 학과: 컴퓨터 소프트웨어학과 학년:3**

**수정 시 입력 값: 주소: 가나시 다라구 마바동**

**계좌번호: 0000-0000-0000-0000로 입력**

**교수1 계정  
ID: 10115 PW: 1234 역할: 교수**

**이름: 이상 학번: 10115 학과: 컴퓨터 소프트웨어학과 신분: 교수**

**수정 시 입력 값: 주소: 가시 다구 바동**

**계좌번호: 1111-1111-0000-0000로 입력**

**조교1 계정  
ID: 11115 PW: 1234 역할: 조교**

**이름: 유관순 학번: 10115 학과: 컴퓨터 소프트웨어학과 신분: 조교**

**수정 시 입력 값: 주소: 사시 아구 차동**

**계좌번호: 1111-1111-0000-0000로 입력**

* **직원1 계정  
  ID: 01115 PW: 1234 역할: 직원**

**이름: 이순신 학번: 01115 학과: 컴퓨터 소프트웨어학과 신분: 직원**

**수정 시 입력 값: 주소: 나시 구구 마마동**

**계좌번호: 1111-1111-1111-0000로 입력**

**테스트 절차:**

* 1. **학생1으로 통합 정보에서 자신정보 메뉴 클릭**
  2. **학생1의 테스트 데이터 값과 조회창에서 출력된 값이 일치하는지 확인진행**

**3. 수정 버튼을 클릭해 주소 및 계좌번호를 입력 값으로 입력 후 등록**

**4. 조회하여 입력값이 제대로 출력되었는지 확인**

**5. 교수1로 통합 정보에서 자신정보 메뉴 클릭**

**6. 교수1의 테스트 데이터에 값과 조회창에서 출력된 값이 일치하는 확**

**인진행**

**7. 수정 버튼을 클릭해 주소 및 계좌번호를 입력 값으로 입력 후 등록**

**8. 조회하여 입력값이 제대로 출력되었는지 확인한다.**

**9. 조교 계정으로 통합정보 메뉴에서 자신정보 조회를 클릭**

**10.조교1의 테스트 데이터 값과 조회창에서 출력된 값이 일치하는지 확인진행**

**11.수정 버튼을 클릭해 주소 및 계좌번호를 입력 값으로 입력 후 등록**

**12.조회하여 입력값이 제대로 출력되었는지 확인**

**13.직원1 계정으로 통합정보 메뉴에서 자신정보 조회를 클릭**

**10.직원1의 테스트 데이터 값과 조회창에서 출력된 값이 일치하는지 확인진행**

**11.수정 버튼을 클릭해 주소 및 계좌번호를 입력 값으로 입력 후 등록**

**12.조회하여 입력값이 제대로 출력되었는지 확인**

* **기대 결과:**
  1. **학생의 정보를 조회할 시 테스트 데이터와 일치하는 값이 조회된다.**

**2. 학생 데이터를 수정할 시 입력 값이 정상적으로 수정된다.**

**3. 교수의 정보를 조회할 시 테스트 데이터와 일치하는 값이 조회된다.**

**4. 교수 데이터를 수정할 시 입력 값이 정상적으로 출력된다.**

**5. 조교의 정보를 조회할 시 테스트 데이터와 일치하는 값이 조회된다.**

**6. 조교 데이터를 수정할 시 입력 값이 정상적으로 수정된다.**

**7. 직원의 정보를 조회할 시 테스트 데이터와 일치하는 값이 조회된다.**

**8. 직원 데이터를 수정할 시 입력 값이 정상적으로 출력된다.**

**.**

**TC-006 게시글 작성, 조회**

**요구사항 내역:  
메인 포털 사이트와 연결된 글을 작성할 수 있는 게시판사이트가 만들어져야 한다.**

**사용자들은 게시판에서 글 작성,조회를 할 수 있다.**

* **목적:  
  사용자가 개시글 작성에 결합요소가 존재 확인하고 이를 수정한다.**
* **테스트 대상:  
   사용자 액터 게시글 조회 유스케이스**
* **테스트 데이터:**

**학생1 계정 ID: 11101001 PW: 1234 역할: 학생**

**게시글 작성 값:  
 제목: 테스트 게시글입니다. 내용: 내용이 출력되는지 확인합니다.**

**사진: 고양이1.jpg**

**테스트 절차:**

* 1. **학생1으로 포털사이트에서 게시판 메뉴 클릭**
  2. **게시판에서 “테스트 게시판” 클릭**

**3. 게시판에서 게시글 작성 클릭**

**4. 게시글 작성에서 게시글 작성 값을 입력한 후 게시글 등록 클릭**

**5. 게시글 목록에서 입력한 게시글이 있는지 확인한다.**

**6. 입력한 게시글을 클릭하여 게시글에서 입력한 값과 조회된 값이 일치하는 지 확인한다.**

* **기대 결과:**
  1. **게시글 작성이 정상적으로 이루어진다.**

**2. 작성된 게시글이 정상적으로 조회된다.**

**TC-006 비밀번호 변경**

* **요구사항 내역:  
  사용자가 비밀번호를 잃어버렸을 시 비밀번호 찾기를 통해 비밀번호를 재설정할 수 있어야 한다.  
  목적:  
  사용자가 비밀번호를 분실했을 때, 비밀번호 찾기 기능을 통해 안전하고 정상적으로 비밀번호를 재설정할 수 있는지를 확인한다.**
* **테스트 대상:  
   사용자 액터 비밀번호 변경 유스케이스**
* **테스트 데이터:**

**학생1 계정**

**ID: 11101001 전화번호: 010-1111-1111 이름: 홍길동**

**교수1 계정**

**ID: 10115 전화번호: 010-2222-2222 이름: 이상**

**조교1 계정**

**ID: 11115 전화번호: 010-3333-3333 이름: 유관순**

**직원1 계정**

**ID: 01115 전화번호: 010-4444-4444 이름: 이순신**

* **테스트 절차:  
  1. 학사관리시스템을 실행하고 로그인 사이트에 접속한다.  
  2. 로그인 화면에서 "비밀번호 찾기" 또는 "비밀번호 재설정" 버튼을 클릭한다.  
  3. 사용자 식별 정보(예: 학번, 이름, 이메일 등)를 입력한다.  
  4. "인증 요청" 버튼을 클릭하여 본인 확인 절차를 수행한다.  
  5. 수신한 인증 코드를 입력하고 "확인"을 클릭한다.  
  6. 새로운 비밀번호와 비밀번호 확인란에 동일한 값을 입력한다.(예:4321)  
  7. "비밀번호 변경" 또는 "재설정" 버튼을 클릭  
  8. 변경한 새 비밀번호(예:4321)로 다시 로그인 한다.**
* **기대 결과:  
  1. 로그인 페이지가 정상적 으로 표시된다.  
  2. 비밀번호 재설정 화면으로 이동된다.  
  3. 입력란에 값이 정상적으로 입력되고, 다음 단계로 이동할 수 있는 버튼이 활성화된다.  
  4. 인증 코드가 전송되고, 인증 입력란이 표시된다.  
  5. 인증이 성공하며 비밀번호 재설정 화면이 표시된다.  
  6. 두 입력란의 값이 일치하면 "변경" 버튼이 활성화된다.  
  7. 비밀번호가 정상적으로 변경되었다는 메시지가 표시되고, 로그인 화면으로 이동된다.  
  8. 로그인이 성공하며 포털 메인 화면으로 이동된다.**

**TC-006 로그아웃**

* **요구사항 내역:  
  사용자는 로그인 사이트에서 학번 및 비밀번호를 입력하여 포털 사이트로 이동할 수 있으며 포털사이트에서 "로그아웃" 버튼 클릭 시 포털  
  사이트에서 나가지며 로그인 사이트로 이동한다.**
* **목적:  
  사용자가 로그아웃이 정상적으로 실행되는지 확인한다**
* **테스트 대상:  
   사용자액터 로그아웃 유스케이스**
* **테스트 데이터:**

**학생1 계정**

**ID: 11101001 PW: 1234 역할: 학생**

**교수1 계정**

**ID: 10115 PW: 1234 역할: 교수**

**조교1 계정**

**ID: 11115 PW: 1234 역할: 조교**

**•직원1 계정**

**ID: 01115 PW: 1234 역할: 직원**

* **테스트 절차:  
  학생1 계정  
  1. 학사관리시스템을 실행하고 로그인 사이트에 접속한다.  
  2. 학번 입력란에 11101001을 입력한다.  
  3. 비밀번호 입력란에 1234를 입력한다.  
  4. "로그인" 버튼을 클릭한다.  
  5. 포털 화면 상단 또는 메뉴에서 "로그아웃" 버튼을 클릭한다.  
  6. 다시 브라우저에서 "뒤로 가기" 버튼을 클릭한다.  
  7. 로그인 페이지에서 로그인 버튼 외 다른 포털 기능을 클릭하려 시도한다.**

**8.직원,교수,조교계정을 대상으로 이를 반복하여 수행한다.**

* **기대 결과:  
  모든 사용자에 대해서  
  1. 로그인 화면이 정상적으로 출력된다.  
  2. 학번 입력란에 입력값이 정상적으로 표시된다.  
  3. 비밀번호 입력란에 입력값이 ●●●● 형태로 표시된다.  
  4. 포털 사이트의 메인 화면으로 정상적으로 이동된다.  
  5. 사용자 세션이 종료되고 로그인 화면으로 이동된다.  
  6. 포털 화면이 보이지 않고, 로그인 페이지가 유지된다. 다음을 만족한다**

**TC-006 로그인**

* **요구사항 내역:  
  사용자는 로그인 사이트에서 학번 및 비밀번호를 입력하여 포털 사이트로 이동할 수 있으며 포털사이트에서 "로그아웃" 버튼 클릭 시 포털  
  사이트에서 나가지며 로그인 사이트로 이동한다.**
* **목적:  
  사용자가 로그인이 정상적으로 실행되는지 확인한다**
* **테스트 대상:  
   사용자액터 로그인 유스케이스**
* **테스트 데이터:**

**학생1 계정**

**ID: 11101001 PW: 1234 역할: 학생**

**교수1 계정**

**ID: 10115 PW: 1234 역할: 교수**

**조교1 계정**

**ID: 11115 PW: 1234 역할: 조교**

**•직원1 계정**

**ID: 01115 PW: 1234 역할: 직원**

* **테스트 절차:  
  학생1 계정  
  1. 학사관리시스템을 실행하고 로그인 사이트에 접속한다.  
  2. 학번 입력란에 11101001을 입력한다.  
  3. 비밀번호 입력란에 1234를 입력한다.  
  4. "로그인" 버튼을 클릭한다.  
  5. 포털 사이트에서 사용자 이름 또는 학번이 화면에 표시되는지 확인한다.  
  교수1 계정  
  1. 학사관리시스템을 실행하고 로그인 사이트에 접속한다.  
  2. 학번 입력란에 10115을 입력한다.  
  3. 비밀번호 입력란에 1234를 입력한다.  
  4. "로그인" 버튼을 클릭한다.  
  5. 포털 사이트에서 사용자 이름 또는 학번이 화면에 표시되는지 확인한다.  
  조교1 계정  
  1. 학사관리시스템을 실행하고 로그인 사이트에 접속한다.  
  2. 학번 입력란에 11115을 입력한다.  
  3. 비밀번호 입력란에 1234를 입력한다.  
  4. "로그인" 버튼을 클릭한다.  
  5. 포털 사이트에서 사용자 이름 또는 학번이 화면에 표시되는지 확인한다.  
  직원1 계정  
  1. 학사관리시스템을 실행하고 로그인 사이트에 접속한다.  
  2. 학번 입력란에 01115을 입력한다.  
  3. 비밀번호 입력란에 1234를 입력한다.  
  4. "로그인" 버튼을 클릭한다.  
  5. 포털 사이트에서 사용자 이름 또는 학번이 화면에 표시되는지 확인한다.**
* **기대 결과:**

**모든 사용자에 대해서  
1. 로그인 화면이 정상적으로 출력된다.  
2. 학번 입력란에 입력값이 정상적으로 표시된다.  
3. 비밀번호 입력란에 입력값이 ●●●● 형태로 표시된다.  
4. 포털 사이트의 메인 화면으로 정상적으로 이동된다.  
5. 로그인된 사용자 정보가 화면에 표시된다.**

**다음을 만족한다.**

**TC-006 출석조회**

**요구사항 내역:  
학생은 현재 자신이 신청한 강의에 출석 정보를 볼 수 있으며 직원, 조교의 경우 모든 수업들의 학생들의 출석 정보들을 조회할 수 있다.**

* **목적:  
  사용자가 개시글 작성에 결합요소가 존재 확인하고 이를 수정한다.**
* **테스트 대상:  
   사용자 액터 출석조회 유스케이스**
* **테스트 데이터:**

**학생1 계정  
ID: 11101001 PW: 1234 역할: 학생 이름: 홍길동 학번:11101001**

**조교1 계정  
ID: 11115 PW: 1234 역할: 조교**

**직원1 계정  
ID: 01115 PW: 1234 역할: 직원**

**출석 데이터**

**수업: java 이름: 홍길동 학번:2010116 출석날짜: 2025-06-26일짜**

**결석시간 입력 3**

**테스트 절차:**

* 1. **학생1 계정으로 통합정보에서 출석조회 클릭**
  2. **수업에서 java클릭 출석데이터의 결석 날짜가 출석조회창에 출력값과 일치하는 지 확인한다.**

**3. 조교1 계정으로 통합정보에서 출석조회 클릭**

**4. 수업에서 java클릭 후 출석데이터의 출석날짜일치 하는 날짜 클릭 후 학생1의 결석시간이 출석데이터값 과 일치하는 지 확인한다.**

**5. 직원1 계정으로 통합정보에서 출석조회 클릭**

**6. 수업에서 java클릭 후 출석데이터의 출석날짜일치 하는 날짜 클릭 후 학생1의 결석시간이 출석데이터값 과 일치하는 지 확인한다.**

**기대 결과:**

**1. 학생이 출석조회가 출석데이터가 정상적으로 출력되는 것을 확인한다.**

**2. 조교,직원이 출석조회가 출석데이터가 정상적으로 출력되는 것을 확인한다.**

**TC-006 게시판 작성**

**요구사항 내역:  
메인 포털 사이트와 연결된 글을 작성할 수 있는 게시판사이트가 만들어져야 한다.**

**교직원들은 게시판에서 게시판 창설을 할 수 있어야한다.**

* **목적:  
  교직원이 게시글 작성을 진행할 때 결합요소가 존재 확인하고 이를 수정한다.**
* **테스트 대상:  
   교직원 액터 게시판 작성 유스케이스**
* **테스트 데이터:**

**조교1 계정  
ID: 11115 PW: 1234 역할: 조교**

**직원1 계정  
ID: 01115 PW: 1234 역할: 직원**

**교수1 계정  
ID: 10115 PW: 1234 역할: 교수 이름: 이순신**

**게시판 데이터:  
이름 테스트 게시판 게시판정보: 테스트용 게시판입니다. 관리자: 이순신**

**게시글 데이터:  
제목: 테스트 게시글입니다. 내용: 테스트 내용입니다.**

**테스트 절차:**

* 1. **교수1으로 포털사이트에서 게시판 메뉴 클릭**
  2. **게시판에서 게시판 창설 클릭**

**3. 게시판 창설에서 게시판 데이터대로 입력 진행 후 등록 버튼 클릭**

**4. 게시판 탭에서 입력한 게시판이 존재하는 지 확인**

**5. 테스트 게시판 클릭 후 게시판정보가 입력값과 동일한지 확인**

**6. 테스트 게시판에서 게시글 작성 클릭 게시글 데이터 입력한다.**

**7. 게시글 조회하여 제대로 일치하는 확인한다.**

* **기대 결과:**
  1. **게시판 창설이 정상적으로 이루어진다**

**2. 게시판에서 게시글을 작성이 제대로 이뤄지는 확인한다.**

**3. 비기능 요구사항 테스트 계획 및 시나리오**

**TC-006 반응시간**

* **요구사항 내역:  
  버튼을 클릭하였을 때 웹사이트가 2초 이내로 반응하여야 하며 길어도 4초 이내로 반응이 와야 한다.**
* **목적:  
  포털사이트에 주요 버튼(통합정보 버튼,수강신청 버튼, 성적조회 버튼) 클릭하였을때 반응 시간을 확인한다.**
* **테스트 대상:  
   성능 요구사항 반응**
* **테스트 데이터**

**학생1 정보:  
ID: 11101001 PW: 1234 역할: 학생**

* **테스트 절차:**
  1. **포털사이트에서 통합정보버튼 클릭**
  2. **lighthouse에서 First Contentful Paint시간 측정**
  3. **포털사이트에서 수강신청버튼 클릭**
  4. **lighthouse에서 First Contentful Paint시간 측정**
  5. **포털사이트에서 성적조회 클릭**
  6. **lighthouse에서 First Contentful Paint시간 측정**
* **기대 결과:**
  1. **수강신청, 통합정보, 성적조회 버튼을 클릭 시간에 평균이 2초이내이다.**

**TC-006 처리율**

* **요구사항 내역:  
  시스템은 피크 타임(Peak Time)에서 1초당 10건 요청을 안정적으로 처리할 수 있어야 한다.**
* **목적:  
  JMeter를 통해 1초당 10건을 처리할 수 있는지 확인하여 향우 방향성을 결정한다.**
* **테스트 대상:  
   성능 요구사항 반응**
* **테스트 데이터**

**학생1 정보:  
ID: 11101001 PW: 1234 역할: 학생**

* **테스트 절차:**
  1. **JMeter를 실행하여 서버 및 네트워크 환경을 점검한다.**
  2. **Thread Group(스레드 그룹) 또는 Throughput Controller를 이용해 초당 10건의 요청이 발생하도록 설정한다.**
  3. **du.potal.com으로 url을 설정한다.**
  4. **결과 기록을 위한 리스너(그래프, 요약 리포트 등)를 추가한다.**
  5. **적은 부하(예: 초당 10~20건)로 사전 테스트를 실행하여 시스템 및 스크립트 동작을 확인한다.**
  6. **정상 응답 및 오류 발생 여부를 확인한다.**
* **기대 결과:**
  1. **1초당에 10건에 요청에 대해 안정적으로 수용한다.**

**TC-006 하드디스크**

* **요구사항 내역:  
  학사관리 시스템 운영에 필요한 최소 30GB (테라바이트) 이상의 사용 가능한 저장 공간을 확보해야 한다. 이는 시스템 데이터, 로그, 백업 등을 포함하는 용량이다.**
* **목적:  
  클라우드를 통한 30GB이 원할하게 이루어지는 확인한다.**
* **테스트 대상:  
   성능 요구사항 반응**
* **테스트 데이터**

**학생1 정보:  
ID: 11101001 PW: 1234 역할: 학생**

* **테스트 절차:**
  1. **클라우드 관리 콘솔접속**
  2. **서버/인스턴스/스토리지 서비스의 ‘할당 용량’ 정보가 30GB로 세팅되어 있는지 확인한다.**
  3. **임의 파일을 생성하여 30GB에 근접할 때까지 저장 시도하여 업로드중 성능하락이 있는지 확인한다.**
  4. **30GB를 초과하여 용량 업로드 진행**
  5. **임의 파일을 생성하여 30GB에 근접할 때까지 저장 시도 시**
  6. **“용량 초과” 안내 또는 오류 메시지가 적절히 출력되는지 확인한다.**
  7. **콘솔에서 실제로 용량이 30GB가 되는지 확인한다.**
* **기대 결과:**
  1. **클라우드에서 30GB에 해당하는 용량이 제공된다.**

**TC-006 안정성**

* **요구사항 내역:  
  서버가 점검시간 오전 4시부터 5시까지를 제외한 나머지 시간에는 서버가 안정인 상태를 유지해야 한다**
* **목적:  
  실제 24시간 모니터링이 어려우므로,  
  점검 시간(오전 4~5시) 전후, 일반 운영 시간(예: 오전 3:55, 4:05, 5:01, 6:00 등)에 테스트를 여러 번 반복하여 안정성을 확인한다.**
* **테스트 대상:  
   비기능 요구사항 안정성 부분**
* **테스트 데이터**

**학생1 정보:  
ID: 11101001 PW: 1234 역할: 학생**

* **테스트 절차:**
  1. **접속시간 전인 3:55분에 웹사이트에 직접 접속한다.**
  2. **이후 오전 4:00, 4:30, 5:00 등 점검 시간대에 접속을 시도한다.**
  3. **“점검중” 메시지를 확인한다.**
  4. **오전 5:01에 웹사이트 또는 서비스에 다시 접속한다.**
  5. **정상적으로 서버에 접속되는지 확인한다.**
* **기대 결과:**
  1. **점검시간일때 접속이 차단된다.**
  2. **점검시간이후에 접속이 되는 지 확인한다.**

**TC-006 안정성**

* **요구사항 내역:  
  동서울 대학교 정원인 8051명이기에 이에 1/4인 2000명 정도는 수용할 수 있을 정도의 서버가 구축되어야 한다.**
* **목적:  
  서버가 약, 2000명 이상의 동시 접속 사용자를 안정적으로 수용할 수 있는지를 확인한다.**
* **테스트 대상 :  
  비 기능 요구사항 트래픽 수용도 부분**
* **테스트 데이터 :  
  1.비용 문제로 인해 실제 수용도는 20명 정도의 동시 접속 사용자   
  2.사용자 행위 시나리오 : 로그인, 강의조회, 수강신청 페이지 이동, 로그아웃 등 시나리오별 요청 흐름  
  3. 테스트 시간 : 각 부하 단계마다 5~10분 지속  
  4. 응답 시간 기준값 : 2초 이내 정상, 3초 이상 경고, 5초 이상 오류로 판단**
* **테스트 절차 :   
  1. 테스트를 수행할 시스템과 서버 환경을 점검한다.  
  2. 테스트에 사용할 시나리오를 설정한다.  
  (예: 로그인 -> 주요 페이지 접속 -> 로그아웃)  
  3. 동시 접속 사용자 수를 1명부터 점진적으로 증가시킨다  
  (예: 1 -> 5 -> 10 -> 15 -> 20)  
  4. 각 접속 단계별로 서버의 응답 상태 및 시스템 부하를 관찰한다.  
  (예: 화면 응답속도, 페이지 전환 속도, 오류 발생 여부 등)  
  5. 최대 동시 접속자 수를 기준으로 최종 테스트를 진행한다.  
  6. 테스트 결과를 기록하고, 응답 시간 및 오류 발생률 등을 분석한다.  
  7. 테스트 종료 후 시스템을 초기화하고 정리한다.**
* **기대 결과:**

**1. 시스템이 정상적으로 작동 가능하며, 테스트 준비가 완료된다.  
2. 시나리오가 명확하게 정의되어 테스트 흐름을 따를 수 있어야한다.  
3. 각 단계에서 시스템이 안정적으로 반응해야 한다.  
4. 모든 사용자가 정상적으로 서비스를 이용할 수 있어야한다.  
5. 서버가 다운되자 않고, 서비스 지연이나 오류 없이 작동해야 한다.  
6. 시스템이 요구되는 트래픽 수준을 수용 가능한지 판단할 수 있어야한다.  
7. 서버가 정상 상태로 복구되고, 테스트 환경이 정리된다.**