
Test Plan / Test Cases Design Document

Project Name	중고등학교 컴퓨팅 사고력 기반 정보 교육 학습 도구 개발
-----------------	---------------------------------

2 조

202203559 김주하

202203581 강서현

지도교수: 박정희 교수님 (서명)

Table of Contents

1.	INTRODUCTION.....	3
1.1.	OBJECTIVE.....	3
2.	TEST PLAN.....	4
3.	TEST CASES.....	8
4.	AI 도구 활용 정보.....	9

1. Introduction

1.1. Objective

이 문서는 초중등용 교육 웹 시스템의 테스트 계획, 테스트 케이스 명세를 포함하고 있다. 본 웹 애플리케이션은 문제해결 기반 컴퓨팅 사고력 함양을 위한 플랫폼으로, 사용자가 문제를 인식하고 분석하며, 패턴을 찾고, 추상화하여 알고리즘을 설계할 수 있도록 4단계로 구성되어 있다. 이 테스트 계획서는 특히, 통합 관리 페이지와 문제분석 페이지를 중점적으로 테스트 대상으로 하여, 핵심 기능의 신뢰성과 사용성을 확보하는 것을 목표로 한다.

2. Test Plan

1. 배경과 목적
1.1 배경
<p>해당 웹은 정보 교과 과정에서의 실생활 문제 해결 능력 향상을 목적으로 기획되었으며, 문제분해, 패턴인식, 추상화, 알고리즘 설계까지의 전 과정을 사용자 중심으로 지원한다. 사용자는 직관적인 인터페이스를 통해 각 단계별 사고 과정을 시각화할 수 있으며, 통합 관리 환경에서는 진행 중인 프로젝트의 흐름과 데이터를 통합적으로 관리할 수 있다. 중요한 프로세스는 다음과 같다.</p> <ul style="list-style-type: none"> ● 사용자가 실생활 속 문제를 인식하고 자연어로 입력함 ● 입력된 문제를 문제분석 페이지에서 마인드맵으로 분해하고 구조화함 ● 구조화된 정보를 바탕으로 패턴인식 페이지에서 유사한 요소들을 그룹화함 ● 핵심적인 요소만 남기는 추상화 단계를 통해 불필요한 정보를 제거하고 요약함 ● 최종적으로 알고리즘 페이지에서 블록코딩(Blockly)을 통해 문제 해결 알고리즘을 설계함 ● 모든 작업은 통합 관리 환경에서 프로젝트 단위로 저장 및 불러오기가 가능하며, 페이지 간 이동과 상태 추적이 가능함
1.2 테스트 목적
<ul style="list-style-type: none"> ● 문제분석 페이지의 마인드맵 생성 및 연결 기능이 직관적이고 오류 없이 동작하는지 확인 ● 통합 관리 환경에서 전체 사용 흐름(4페이지)을 유기적으로 추적 및 수정할 수 있는지 검증 ● 주요 기능(노드 추가/삭제, 저장 등)의 신뢰성과 반복 사용 가능성을 확보 ● 사용자 경험(UX) 측면에서 페이지 간 연결성과 반응 속도를 점검
2. 테스트 상세
2.1 테스트 항목
<p>본 테스트 계획에서는 아래와 같은 시스템 기능들을 중심으로 테스트를 수행한다. 특히 문제분석 페이지의 시각적 마인드맵 기능과 전체 4 단계 페이지를 통합 관리할 수 있는 기능이 핵심 테스트 항목이다.</p> <p>테스트 항목은 다음과 같다:</p> <ul style="list-style-type: none"> ● 문제분석 기능 테스트 <ul style="list-style-type: none"> ○ 마인드맵 노드 추가, 삭제, 수정 ○ 노드 간 연결 및 분기 생성 ○ 노드 내용 입력 및 드래그 위치 이동

<ul style="list-style-type: none"> ○ 마인드맵 저장 및 불러오기 ● 통합 관리 환경 테스트 <ul style="list-style-type: none"> ○ 4 단계 문제 해결 흐름의 시각적 표시 ○ 페이지 간 이동 및 상태 연동 확인 ○ 프로젝트 저장, 불러오기, 삭제 기능 ○ 이전 단계로 돌아가 수정한 내용 반영 여부 확인 ● 데이터 흐름 테스트 <ul style="list-style-type: none"> ○ 사용자가 입력한 정보가 JSON 구조로 저장되는지 ○ 서버 혹은 로컬 스토리지 기반으로 제대로 관리되는지 확인
2.2 테스트될 요소(features)
<p>테스트 프로젝트는 다음의 요소를 테스트한다.</p> <ul style="list-style-type: none"> ● 사용자 입력 인터페이스: 자연어 문제 입력, 선택 입력 등 ● 문제분석 기능: 노드 시각화, 노드 간 연결, 노드 내용 수정 및 위치 변경 ● 저장 및 불러오기 기능: 로컬/서버 기반 데이터 저장 및 불러오기 ● 피드백 요소: 잘못된 입력/빈 입력 시 오류 메시지 출력 및 입력 유도
2.3 테스트되지 않을 요소
<p>이 시스템의 테스트에서는 다음의 사항은 다루지 않는다.</p> <ul style="list-style-type: none"> ● 디자인 피드백이나 색상 대비 등 UI 디테일 ● 모바일 최적화 테스트 ● 보안/침투 테스트 및 데이터 백엔드 검증 ● 사용자 인증 및 권한 관리 시스템 ● 백업 및 복원 기능 ● 실제 교육현장 적용 및 피드백 수렴 테스트
2.4 접근 방법
<ul style="list-style-type: none"> - 사용자 시나리오 기반 테스트(페르소나 사용) - UI 인터랙션 테스트 (드래그/입력/저장 등)
2.5 테스트 항목의 pass/fail 기준
<p>다음 각 항이 수행될 때까지 테스트는 완료된 것으로 간주하지 않는다.</p> <ul style="list-style-type: none"> ● 요구된 모든 기능을 사용자가 오류 없이 수행 가능해야 함. ● 각 주요 버튼(추가/삭제/저장/전환)의 반응이 정상적이어야 함. ● 마인드맵 기능에서 노드 생성/연결/삭제 후 시각적 이상 없음.
2.6 테스트 산출물(deliverables)
<p>테스트 종료 후 다음의 산출물을 인도한다.</p> <ul style="list-style-type: none"> ● 테스트 계획 ● 테스트 케이스 명세

<ul style="list-style-type: none"> ● 테스트 결과 보고서 ● 발견된 결함 목록
3. 테스트 관리
3.1 작업
<p>중요한 테스트 작업은 다음과 같다.</p> <ol style="list-style-type: none"> 1 테스트 계획을 개발한다. 2 테스트 팀을 구성한다. 3 시스템 요구 사항과 기능 명세를 검토한다. 4 테스트 케이스를 작성하고 테스트 절차를 개발한다. 5 테스트 계획, 테스트 케이스, 절차를 검토하고 우선순위를 지정한다. 6 상세한 테스트 계획에 따라 주요 기능 시나리오별 기능 시뮬레이션을 시행한다. 7 테스트 리포트를 작성하고 오류를 보고한다. 8 결함을 수정한다. 9 반복 테스트를 수행한다. 10 테스트 결과를 문서화한다. 11 테스트 종료 조건을 기준으로 시스템을 릴리스 할 시점을 결정한다.
3.2 기술 자원
<p>테스트 프로젝트를 위하여 테스트 엔지니어는 독립적인 테스트 실험실과 장비가 필요하다. 실험실에 갖추어져야 할 테스트 장비는 다음과 같다.</p> <ul style="list-style-type: none"> ● Chrome 브라우저, React 개발자 도구 ● Node.js 및 로컬 서버 환경 ● 테스트용 JSON 마인드맵 데이터 ● Flow 시각화 라이브러리
3.3 책임과 권한 (인력 자원)
<p>테스트 프로젝트를 위하여 필요한 인력은</p> <ul style="list-style-type: none"> ● 사용자 중심 기능의 테스트를 위한 교사나 학생 페르소나 ● 테스트 전문가 ● 프론트엔드/백엔드 담당자 <p>♦ 앞으로 이 시스템을 지원하고 유지보수 하여야 할 시스템을 잘 이해하는 IT 인력</p>
3.4 훈련
<p>개발자를 대상으로 페이지의 흐름을 브리핑함.</p> <p>테스트 도구 사용법 및 주요 기능 시연 영상을 촬영하고 공유함.</p> <p>페르소나로 지정된 사용자에게 시나리오 예제를 사전에 공유함.</p>

3.5 일정	
단계	일정
테스트 계획 확립	5월 17일
테스트 케이스 작성	5월 19일~5월 21일
기능 테스트 진행	5월 22일~5월 26일
오류 확인 및 검토	5월 27일
테스트 결과 보고	5월 28일
3.6 위험 요소와 비상 대처 상황	
<p>통합 관리 환경에서 페이지 간 연동 오류가 발생할 경우 기능 테스트에 차질이 생기므로, 연동 로직에 대한 사전 코드 검토 기간을 포함한다.</p> <p>테스트 인력이 시스템 사용법에 익숙하지 않을 경우 테스트 과정에서 비효율이 발생하므로, 테스트 시작 전 시나리오 설명과 예시 영상을 제공한다.</p>	

3. Test Cases

1. 서론																																		
1.1 테스트 범위																																		
<p>본 테스트는 정보 교육용 웹의 주요 기능 중 문제분석 페이지와 통합 관리 페이지를 중심으로 기능 테스트를 수행한다. 사용자가 실생활 문제를 분석하고 시각적으로 구조화하며, 전체 문제해결 과정을 관리하는 기능들이 테스트 대상이다.</p> <ul style="list-style-type: none"> ● 문제분해 페이지 내 노드 생성, 연결, 삭제, 수정, 저장, 불러오기 기능 ● 각 단계(문제분해 ~ 알고리즘) 간의 흐름 및 페이지 전환 기능 ● 사용자 입력 유효성 및 기본적인 예외처리 기능 																																		
1.2 테스트 상황																																		
<p>본 테스트는 개발팀 외부의 테스트 담당자 또는 독립적 역할을 수행하는 개발자가 직접 수행하며, 웹의 기능 동작 여부와 UI 상호작용을 검증하는 기능 중심 블랙박스 테스트를 기반으로 한다. 테스트는 Chrome 브라우저 기반 로컬 서버 환경에서 수행되며, React 기반 프론트엔드와 연동된 상태에서 실제 사용자 시나리오에 따라 테스트가 진행된다.</p>																																		
1.3 문서 표기법																																		
특별한 표기법을 사용하지 않음																																		
2. 테스트 케이스																																		
2.1 테스트 케이스 명세																																		
<table border="1"> <thead> <tr> <th>Id</th><th>테스트 대상</th><th>테스트 조건</th><th>테스트 데이터</th><th>예상 결과</th></tr> </thead> <tbody> <tr> <td>PA-1</td><td>문제분석 페이지 - 노드 추가</td><td>빈 마인드맵 상태에서 노드 추가 버튼 클릭</td><td>노드 제목: “문제 정의”</td><td>새 노드가 마인드맵에 추가되고 제목이 반영됨</td></tr> <tr> <td>PA-2</td><td>문제분석 페이지 - 노드 삭제</td><td>노드가 존재하는 상태</td><td>삭제할 노드 선택</td><td>선택한 노드가 마인드맵에서 삭제됨</td></tr> <tr> <td>PA-3</td><td>문제분석 페이지 - 노드 연결</td><td>두 개 이상의 노드가 존재할 때</td><td>노드 A, 노드 B 선택</td><td>노드 A에서 B로 연결선이 생성됨</td></tr> <tr> <td>PA-4</td><td>문제분석 페이지 - 드래그 이동</td><td>노드 위치 변경</td><td>노드 A를 오른쪽으로 드래그</td><td>노드 A의 위치가 드래그한 위치로 이동됨</td></tr> <tr> <td>PA-5</td><td>문제분석 페이지 - 저장 기능</td><td>마인드맵 작성 후 저장 버튼 클릭</td><td>작성된 노드/연결 정보</td><td>JSON 형식으로 로컬/서버에 저장됨</td></tr> </tbody> </table>					Id	테스트 대상	테스트 조건	테스트 데이터	예상 결과	PA-1	문제분석 페이지 - 노드 추가	빈 마인드맵 상태에서 노드 추가 버튼 클릭	노드 제목: “문제 정의”	새 노드가 마인드맵에 추가되고 제목이 반영됨	PA-2	문제분석 페이지 - 노드 삭제	노드가 존재하는 상태	삭제할 노드 선택	선택한 노드가 마인드맵에서 삭제됨	PA-3	문제분석 페이지 - 노드 연결	두 개 이상의 노드가 존재할 때	노드 A, 노드 B 선택	노드 A에서 B로 연결선이 생성됨	PA-4	문제분석 페이지 - 드래그 이동	노드 위치 변경	노드 A를 오른쪽으로 드래그	노드 A의 위치가 드래그한 위치로 이동됨	PA-5	문제분석 페이지 - 저장 기능	마인드맵 작성 후 저장 버튼 클릭	작성된 노드/연결 정보	JSON 형식으로 로컬/서버에 저장됨
Id	테스트 대상	테스트 조건	테스트 데이터	예상 결과																														
PA-1	문제분석 페이지 - 노드 추가	빈 마인드맵 상태에서 노드 추가 버튼 클릭	노드 제목: “문제 정의”	새 노드가 마인드맵에 추가되고 제목이 반영됨																														
PA-2	문제분석 페이지 - 노드 삭제	노드가 존재하는 상태	삭제할 노드 선택	선택한 노드가 마인드맵에서 삭제됨																														
PA-3	문제분석 페이지 - 노드 연결	두 개 이상의 노드가 존재할 때	노드 A, 노드 B 선택	노드 A에서 B로 연결선이 생성됨																														
PA-4	문제분석 페이지 - 드래그 이동	노드 위치 변경	노드 A를 오른쪽으로 드래그	노드 A의 위치가 드래그한 위치로 이동됨																														
PA-5	문제분석 페이지 - 저장 기능	마인드맵 작성 후 저장 버튼 클릭	작성된 노드/연결 정보	JSON 형식으로 로컬/서버에 저장됨																														

PA-6	문제분석 페이지 - 불러오기	저장된 마인드맵 존재	기존 프로젝트 데이터	저장된 노트/연결 상태가 화면에 로드됨
IM-1	통합관리 페이지 - 흐름 시각화	각 단계 페이지 완료 상태	문제분석~알고리즘까지 4단계 모두 완료	전체 흐름이 순차적으로 시각화되어 표시됨
IM-2	통합관리 페이지 - 페이지 이동	문제분석 완료 후 패턴인식 페이지로 이동	문제분석 페이지 데이터 존재	패턴인식 페이지로 전환되며 데이터가 연동됨
IM-3	통합관리 페이지 - 이전단계 수정	알고리즘 단계에서 문제분석 수정	문제분석 페이지로 돌아가 노트 변경	수정된 내용이 이후 단계에도 반영됨
DATA-1	데이터 저장 구조 검증	저장 시 JSON 구조 생성 여부 확인	마인드맵 JSON 예시 데이터	예상된 키/값 구조를 포함한 JSON이 생성됨
DATA-2	저장소 연동 검증	서버 저장/불러오기 기능	저장 요청 후 서버 응답 확인	데이터가 정상적으로 저장되고 다시 로드됨

2.2 테스트 환경

- 백엔드 환경: Node.js 기반 로컬 개발 서버
- 프론트엔드 프레임워크: React 18 (Next.js 기반)
- 데이터베이스: Firebase Firestore
 - 인증 시스템: Firebase Authentication (Google 계정 연동)
 - 데이터 포맷: JSON 기반 마인드맵 구조 및 문제 분석 데이터 저장
 - 테스트 계정: Firebase 콘솔에 사전 등록된 테스트용 Google 계정 사용

2.3 테스트 절차 요구사항

- 선행 조건: 로컬 개발 서버가 실행 중이어야 하며, 로그인된 사용자 상태여야 함
- 후행 조건: 테스트 후 생성된 데이터는 테스트용 데이터베이스 또는 로컬스토리지에서 제거해야 함
- 각 테스트 케이스는 독립적으로 수행되어야 하며, 이전 테스트의 영향을 받지 않아야 함

4.AI 도구 활용 정보

사용 도구 GPT-4, Claude 2.1

사용 목적	인터뷰 질문 초안 작성, 문장 흐름 정리, 사례 리서치 보조
프롬프트	<ul style="list-style-type: none"> ● 사용자 경험 중심으로 문제정의서 예시를 보여줘 ● 디자인 브레인스토밍 아이디어 10개 제안해줘
반영 위치	<ol style="list-style-type: none"> 1. 인터뷰 질문 목록 (p.5) 2. 아이디어 설명 문단 정리 (p.6)
수작업 수정	있음(논리 보강, 사례 교체 등)