중·고등학교 컴퓨팅 사고력 기반 정보 교육 학습 도구 개발

Test Plan / Test Cases Design

건설공학교육과 202203559 김주하
교육학과 202203518 강서현

Programming platforms

Computer Education

Innovation Growth Impact

목차

Introduction

본 웹은 문제해결 기반 컴퓨팅 사고력 함양을 위한 플랫폼으로,

사용자가 문제를 인식하고 분석하며, 패턴을 찾고, 추상화하여 알고리즘을 설계할 수 있도록 4단계로 구성되어 있다.

이 테스트 계획은 특히, 통합 관리 페이지와 문제분석 페이지를 중점적으로 테스트 대상으로 하여

핵심 기능의 신뢰성과 사용성을 확보하는 것을 목표로 한다.

Test Plan

1.1 배경

- 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

Test Plan

1.2 테스트 목적

- 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

Test Plan

2.1 테스트 항목

문제분해 기능

마인드맵 노드 추가, 삭제, 수정
노드 간 연결 및 분기 생성

저장 기능

프로젝트 저장,
불러오기, 삭제 기능

핵심 테스트

4단계 흐름 관리

4단계 문제 해결 흐름의
시각적 표시

데이터 흐름

사용자가 입력한 정보가
잘 저장되는지 확인

Test Plan

2.2 테스트 요소

테스트 될 요소

- 사용자 입력 인터페이스
- 문제분석 기능
- 저장 및 불러오기 기능
- 피드백 요소

VS

테스트 되지 않을 요소

- 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

Test Plan

2.3 접근 방법

- 사용자 시나리오 기반 테스트(페르소나 사용)
- UI 인터랙션 테스트 (드래그/입력/저장 등)

2.4 P/F 기준

다음 각 항이 수행될 때까지 테스트는 완료된 것으로 간주하지 않는다.

- 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.

Test Plan

2.3 접근 방법

- 사용자 시나리오 기반 테스트(페르소나 사용)
- UI 인터랙션 테스트 (드래그/입력/저장 등)

2.4 P/F 기준

다음 각 항이 수행될 때까지 테스트는 완료된 것으로 간주하지 않는다.

- 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.

2.5 산출물

테스트 계획
테스트 케이스 명세
테스트 결과 보고서
발견된 결함 목록

Test Plan

3.1 작업

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 결합을 수정한다.
9 반복 테스트를 수행한다.
10 테스트 결과를 문서화한다.
11 테스트 종료 조건을 기준으로 시스템을 릴리스 할 시점을 결정한다.

Test Plan

3.2 기술 자원

- Chrome 브라우저, React 개발자 도구
- Node.js 및 로컬 서버 환경
- 테스트용 JSON 마인드맵 데이터
- Flow 시각화 라이브러리

3.3 인력 자원

- 사용자 중심 기능의 테스트를 위한 교사나 학생 페르소나
- 테스트 전문가
- 프론트엔드/백엔드 담당자

3.4 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

Test Plan

3.6 비상 대처

- 통합 관리 환경에서 페이지 간 연동 오류가 발생할 경우 기능 테스트에 차질이 생기므로, 연동 로직에 대한 사전 코드 검토 기간을 포함한다.
- 테스트 인력이 시스템 사용법에 익숙하지 않을 경우 테스트 과정에서 비효율이 발생하므로, 테스트 시작 전 시나리오 설명과 예시 영상을 제공한다.

Test Cases

1.1 테스트 범위

- 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

- 개발팀 외부의 테스트 담당자 또는 독립적 역할을 수행하는 개발자가 직접 수행
- 웹의 기능 동작 여부와 UI 상호작용을 검증하는 기능 중심 블랙박스 테스트 기반
- 테스트는 Chrome 브라우저 기반 로컬 서버 환경에서 수행
- React 기반 프론트엔드와 연동된 상태에서 실제 사용자 시나리오에 따라 테스트 진행

Test Cases

2.1 테스트 케이스 명세

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

Test Cases

2.1 테스트 케이스 명세

PA-6	문제분석 페이지 - 불러오기	저장된 마인드맵 존재	기존 프로젝트 데이터	저장된 노드/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 노드 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

Test Cases

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 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

감사합니다

202203559 김주하 | 202203518 강서현