



읽쓰 문해력 학습 플랫폼 - 프로젝트 분석 보고서

문서 버전: v1.0

작성일: 2025-10-23

원본 기획서: icks_literacy_plan_20251020.pdf



목차

- 프로젝트 개요
- 시스템 아키텍처
- 주요 기능 분석
- 사용자 플로우
- 데이터 흐름 분석
- 비즈니스 로직
- 기술적 요구사항

1. 프로젝트 개요

1.1 프로젝트 목적

초등학생의 독해력 향상을 위한 체계적인 온라인 학습 플랫폼 구축

- 단계별(A, B, C) 맞춤형 학습 제공
- 상세한 학습 진행 추적 및 분석
- 게임화(Gamification)를 통한 학습 동기 부여

1.2 핵심 가치 제안

구분	내용
학생	재미있는 퀘스트 시스템, 즉각적인 피드백, 성취감
학부모	상세한 학습 결과 리포트, 진행률 추적
교육기관	학생 관리 도구, 라이선스 관리, 선물하기 기능

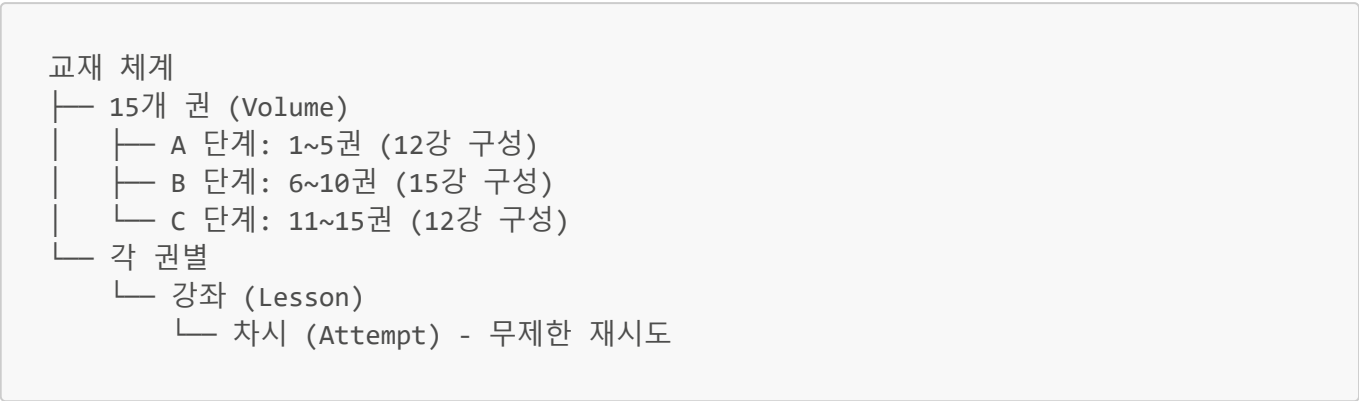
1.3 서비스 구성

읽쓰 문해력 플랫폼

- 강좌 학습 시스템 (15권, 권당 12~15강)
- 종합 어휘장 (오늘의 어휘, 테마 학습)
- 서술 요약 (AI 교정)
- 퀘스트 시스템 (나만의 디저트 만들기)
- 학습 이력 관리 (캘린더, 결과지)

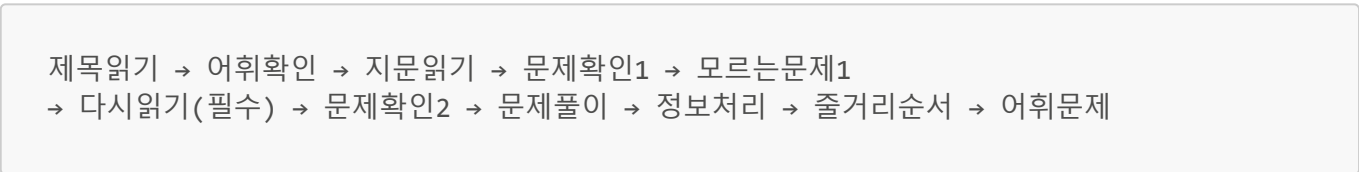
2. 시스템 아키텍처

2.1 교재 구조

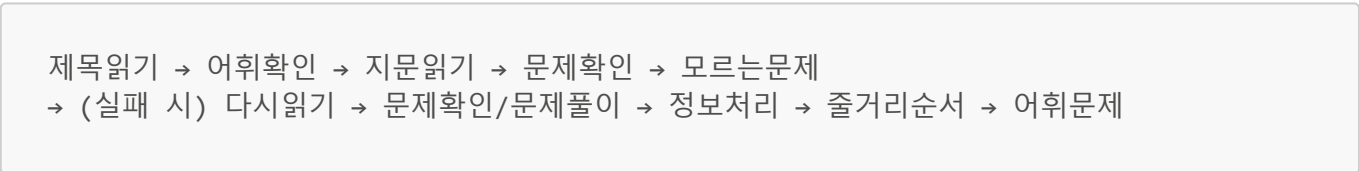


2.2 학습 구간 아키텍처

A 단계 (1~5권) - 기초

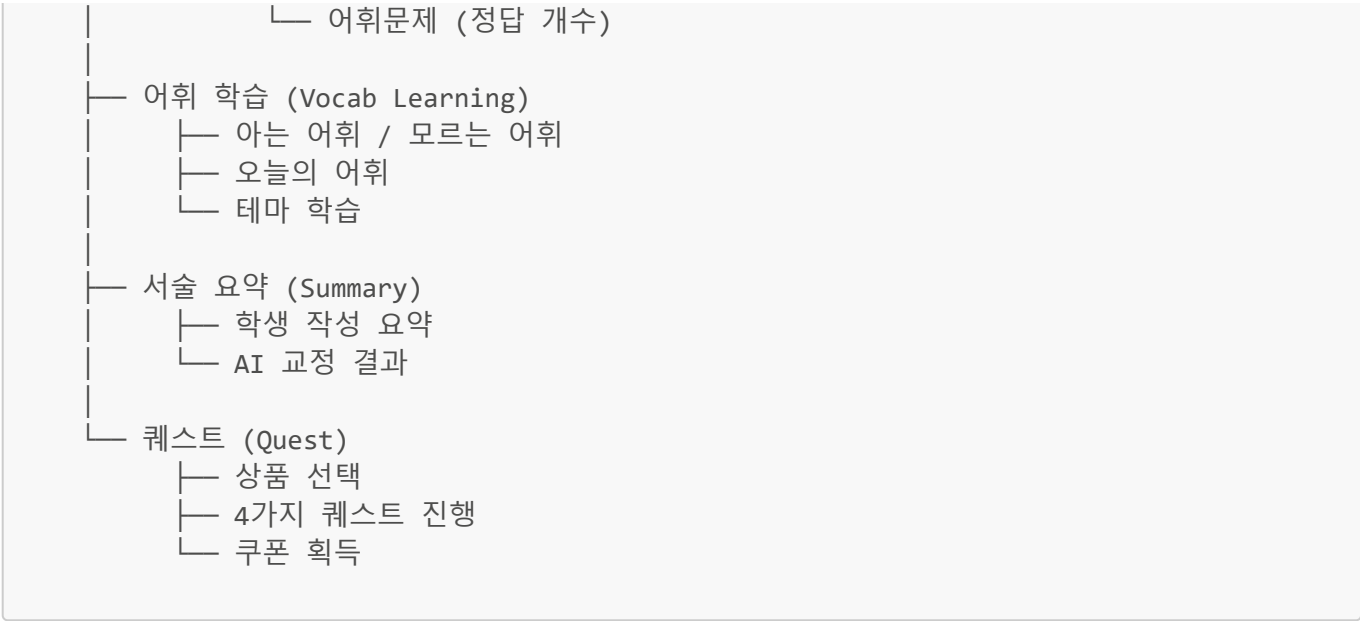


B/C 단계 (6~15권) - 중급/고급



2.3 데이터 계층 구조

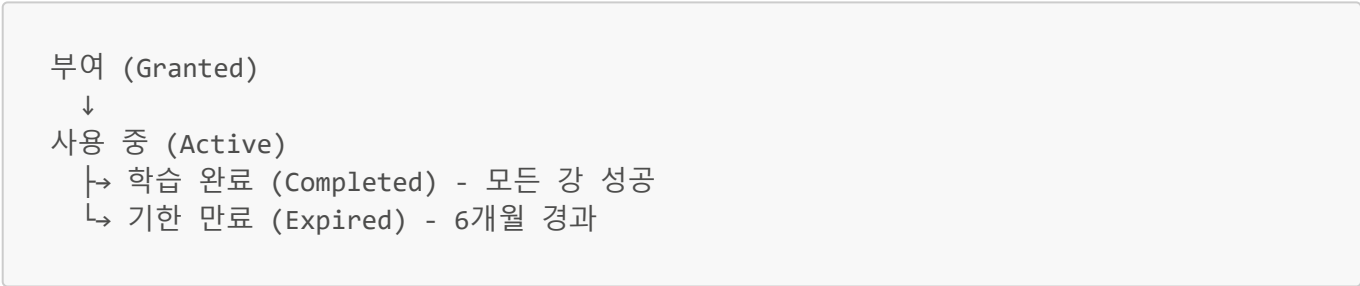




3. 주요 기능 분석

3.1 라이선스 관리 시스템

라이선스 생명주기

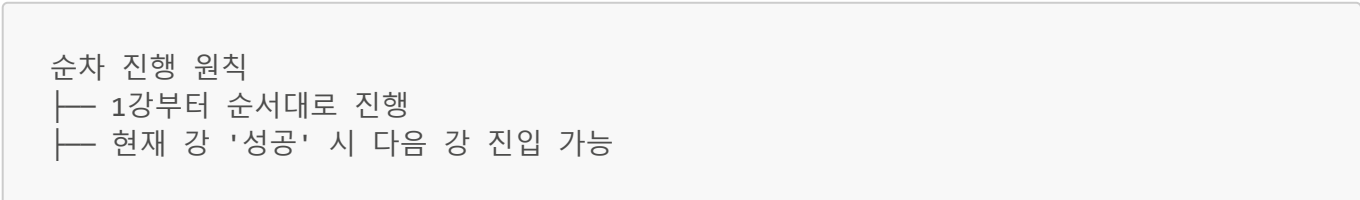


라이선스 규칙

규칙	설명
사용 기한	부여 시점 기준 6개월
중복 부여	동일 권 중복 가능 → 별도 학습관 생성
진행 보장	기한 내 학습 이력 유지
결과 접근	만료 후에도 결과지 조회 가능

3.2 강좌 학습 시스템

강좌 진행 규칙



- └─ 동일 강 내 3차시 연속 실패 → 자동 패스

└─ 패스된 강 클릭 → 재진행 가능

차시 상태 관리

상태	설명	조건
성공	모든 구간 통과	어휘문제까지 완료
실패	탈락 구간 발생	제한 시간 초과, 오답 초과 등
진행 중	학습 수행 중	20분 이내 재접속 가능

이어하기 기능

- 20분 이내: 중단 지점부터 이어하기 가능
- 20분 초과: 차시 이력 초기화, 처음부터 재시작

자동 종료

- 10분 이상 활동 없음: 알림 팝업 표시
- 팝업 미응답: 자동 종료
- 팝업 응답: 학습 계속 진행

3.3 단계별 학습 특성

A 단계 (1~5권) - 기초 단계

- 특징:
- 강좌 수: 12강

- 문항 수: 6문항 (사실형 100%)

- 다시읽기: 필수 진행
- 읽기 시간:
- 지문읽기 목표: 3분 (초과 가능)

- 다시읽기 제한: 2분 40초
- 탈락 조건:
- 다시읽기: 제한 시간 초과

- 문제확인2: 모르는 문제 2개 이상

- 문제풀이: 오답+모름 2개 이상

B 단계 (6~10권) - 중급 단계

- 특징:
- 강좌 수: 15강

- 문항 수: 7~9문항 (사실+추론)

- 다시읽기: 조건부 (실패 시만)

읽기 시간:

- 지문읽기 제한: 3분
- 다시읽기 제한: 2분 50초

탈락 조건:

- 지문읽기: 제한 시간 초과
- 문제확인: 모르는 문제 2개 이상 (1차), 1개 이상 (2차)
- 문제풀이: 오답+모름 2개 이상 (1차), 1개 이상 (2차)
- 비판 문항은 탈락 조건에서 제외

C 단계 (11~15권) - 고급 단계

특징:

- 강좌 수: 12강
- 문항 수: 10문항 (사실+추론+비판)
- 다시읽기: 조건부 (실패 시만)

읽기 시간:

- 지문읽기 제한: 3분
- 다시읽기 제한: 3분

탈락 조건:

- 지문읽기: 제한 시간 초과
- 문제확인: 모르는 문제 1개 이상
- 문제풀이: 오답+모름 1개 이상 (비판 제외)
- 가장 엄격한 통과 기준

3.4 종합 어휘장 시스템

어휘 분류 체계

어휘 초기 상태 (강좌 학습 시)

- ↳ 어휘문제 정답 → [아는 어휘]
- ↳ 어휘문제 오답 → [모르는 어휘]

어휘 학습 후 재분류

- ↳ 2문제 모두 정답 → [아는 어휘]로 이동
- ↳ 1문제 이상 오답 → [모르는 어휘]로 이동

오늘의 어휘

생성 조건:

- 강좌 성공 차시의 어휘문제에서 틀린 문제
- 최근 성공한 강좌 순으로 자동 생성

- 화면에 최대 3개 강만 표시

진행 구간:
살펴보기 → 집중학습 → 문제풀기

완료 조건:

- 문제풀기까지 완료 시 결과지 생성
- 완료된 강은 목록에서 제거

퀘스트 반영:

- 어휘문제에서 모든 어휘 정답 시에도 "오늘의 어휘 완료"로 간주

테마 학습

테마	대상 어휘	출제 개수
내가 선택한 어휘	학생이 직접 선택	최대 10개 (1개 이상 가능)
최근 맞힌 어휘	최근 10개 학습에서 아는 어휘로 이동한 어휘	10개 (랜덤)
추천 어휘	최근/오랫동안 제외한 나머지	10개 (랜덤)
오랫동안 안 본 어휘	1개월 이상 미진행 어휘	10개 (랜덤)

제한사항: 사용 기한 만료 권의 어휘는 테마 학습 불가

3.5 서술 요약 시스템

발문 유형

- 지문 난이도에 따라 **4가지 유형** 존재
- 각 유형별 힌트 제공 (접기/펼치기 가능)

작성 프로세스

줄거리순서 구간 완료

↓

서술 요약 생성 (오늘의 요약)

↓

학생 작성 (나의 요약)

↓

[확인] 버튼 클릭

- ↳ 모범 답안 표시
- ↳ AI 교정 결과 표시

작성 완료 후에도 수정 가능

- ↳ AI 교정도 함께 업데이트

오늘의 요약

- 최근 학습 완료 강좌 순으로 최대 3개 표시
- 완료 시 목록에서 제거
- 사용 기한 만료 권은 조회만 가능 (수정 불가)

3.6 퀘스트 시스템 (나만의 디저트 만들기)

퀘스트 구조

4가지 퀘스트:

1. 강좌 성공: 10개
2. 오늘의 어휘 완료: 7개
3. 오늘의 요약 완료: 7개
4. 테마 학습 1회 완료: 5개

진행률 계산:

(완료 횟수 / 목표 횟수) × 100

상품 획득 조건:

- 4가지 퀘스트 모두 100% 달성

상품 및 보상 시스템

상품 선택

- ↳ 변경 불가 (완료 시까지)
- ↳ 중복 선택 가능
- ↳ 미선택 시에도 학습 이력 누적

완료 시 보상:

- ↳ 일반 쿠폰: 무조건 지급
- ↳ 우수 쿠폰: 순차 완료 시 추가 지급
(추가 활동 미루지 않고 진행)

퀘스트 진행 UI

재료 게이지 표시:

- 진행률 1/3 지점마다 재료 양 증가
- 모두 캐릭터로 진행 독려

초과 횟수 처리:

- 특정 퀘스트 100% 초과 시 횟수 유지
- 다음 상품 선택 시 초과분 일괄 반영

3.7 학습 결과 관리

결과지 종류

결과지	범위	포함 내용
간이 결과	차시별	구간별 소요 시간, 정답 개수, 성공/실패
상세 결과	차시별	구간별 세부 데이터, 문제 분석
권별 종합 결과	권 전체	성공 차시 누적 데이터, 통계
어휘 학습 결과	어휘 학습별	예문 작성, 유형별 이해도

학습 현황 신호등

지문읽기/다시읽기:

- 표준: 평소 읽기 속도 70~130%
- 느림: 70% 미만
- 빠름: 130% 초과
- 제한 시간 초과: 탈락

문제확인:

- 모르는 문제 초과: 탈락

문제풀이:

- 오답 문제 초과: 탈락
- 이해도 70 이상: 좋음
- 이해도 70 미만: 나쁨

줄거리순서:

- 1회 정답: 좋음
- 2회 정답: 좋음
- 2회 오답: 나쁨

3.8 알림 시스템

알림 종류

선물하기:

- 학원장이 포인트 선물 시
- 즉시 팝업 표시
- 알림 이력에 저장

QnA 답변:

- 질문에 답변 생성 시
- 클릭 시 QnA 글로 이동

라이선스 사용 기한:

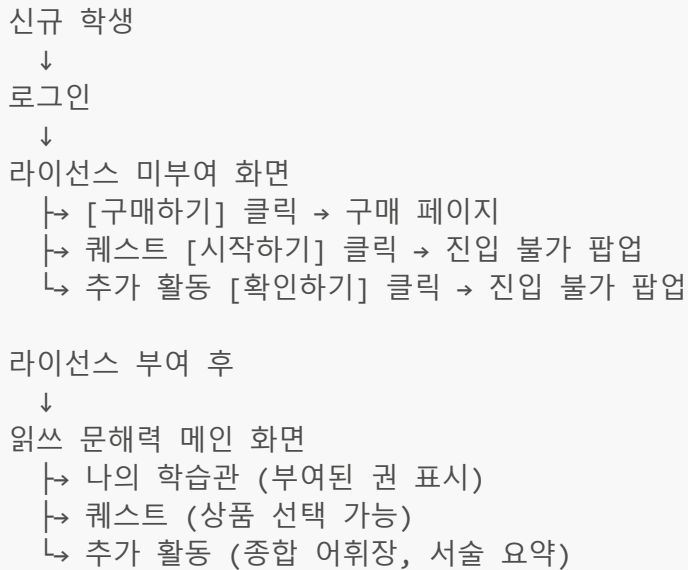
- 사용 기한 7일 미만 시
- 권별로 개별 알림
- 매일 확인 (중복 방지)

알림 관리

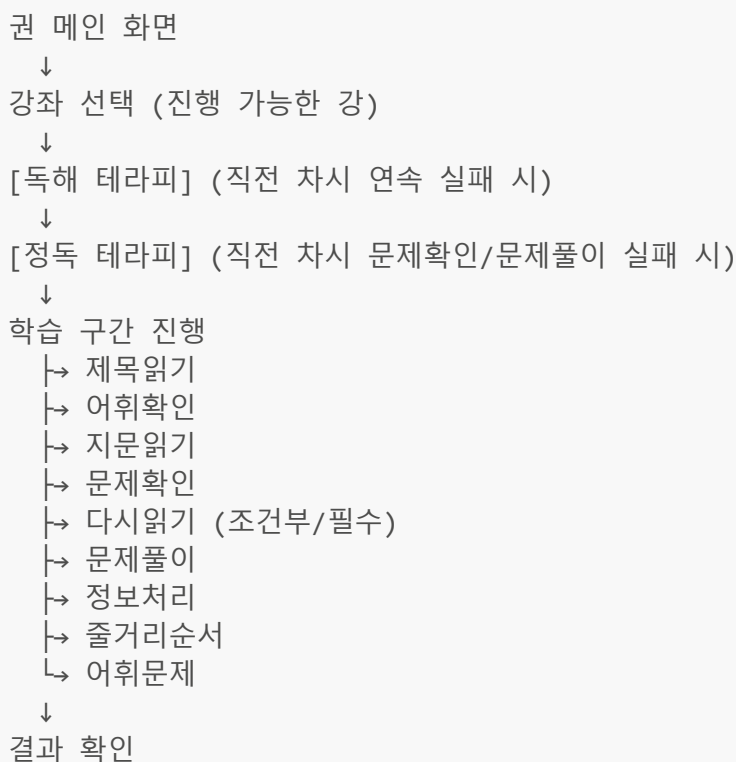
- 최근 10개만 표시
- 읽음/안읽음 상태 관리
- 확인 시 알림 아이콘 숨김

4. 사용자 플로우

4.1 초기 진입 플로우



4.2 강좌 학습 플로우



- ↳ 성공 → 간이 결과 → 권 메인
- ↳ 실패 → 간이 결과 → 권 메인
- ↓
- 추가 활동 가능
 - ↳ 오늘의 어휘 (틀린 어휘 있을 시)
 - ↳ 오늘의 요약 (강 성공 시)

4.3 어휘 학습 플로우

종합 어휘장 메인

↓

오늘의 어휘

- ↳ 강좌 선택 (1개 이상)
- ↳ 살펴보기 (어휘 카드)
- ↳ 집중학습 (예문 작성)
- ↳ 문제풀기 (2순위, 3순위 문제)

↓

결과 확인

- ↳ 2문제 모두 정답 → 아는 어휘로 이동

테마 학습

- ↳ 내가 선택한 어휘 (직접 선택)
- ↳ 최근 맞힌 어휘
- ↳ 추천 어휘
- ↳ 오랫동안 안 본 어휘

↓

문제풀기 (최대 10문제)

↓

결과 확인

4.4 퀘스트 진행 플로우

퀘스트 화면 진입

↓

상품 선택 (처음 또는 완료 후)

↓

학습 활동 수행

- ↳ 강좌 성공
- ↳ 오늘의 어휘 완료
- ↳ 오늘의 요약 완료
- ↳ 테마 학습 완료

↓

퀘스트 진행률 상승

- ↳ 재료 게이지 증가

↓

4가지 퀘스트 모두 100%

↓

상품 획득

- ↳ 일반 쿠폰 발급

↳ (순차 완료 시) 우수 쿠폰 발급
↓
쿠폰함 저장

5. 데이터 흐름 분석

5.1 학습 데이터 생성 흐름

```

강좌 진입
↓
lesson_attempt 생성 (in_progress)
  ↳ attempt_number (차시 번호)
  ↳ started_at (시작 일시)
  ↳ result (진행 중)
↓
각 구간 진행
↓
section_result 생성 (구간별)
  ↳ section_type (구간 유형)
  ↳ elapsed_time (소요 시간)
  ↳ reading_speed (분당 글자 수)
  ↳ correct_count (정답 개수)
  ↳ section_data (JSON 추가 데이터)
↓
어휘문제 완료
↓
lesson_attempt 업데이트
  ↳ result (success/fail)
  ↳ completed_at (완료 일시)
  ↳ total_elapsed_time (총 소요 시간)
↓
어휘 분류
↓
vocab_learning 업데이트
  ↳ status (known/unknown)
  ↳ last_correct_at / last_wrong_at
↓
퀘스트 진행률 업데이트
↓
quest_progress 업데이트
  ↳ completed_count, progress_rate

```

5.2 어휘 학습 데이터 흐름

```

오늘의 어휘 / 테마 학습 진행
↓
vocab_learning_result 생성
  ↳ learning_type (학습 유형)

```

```

└─ total_problems (출제 문항 수)
└─ correct_count (정답 개수)
└─ completed_at (완료 일시)
↓
vocab_learning 업데이트
└─ status 재분류 (known/unknown)
└─ total_correct_count 증가
└─ last_correct_at 업데이트
↓
퀘스트 진행률 업데이트 (해당 시)

```

5.3 퀘스트 데이터 흐름

```

학습 활동 완료
↓
quest_history 생성
└─ quest_type (퀘스트 유형)
└─ activity_date (활동 일자)
└─ lesson_id (강좌 성공 시)
↓
quest_progress 업데이트
└─ completed_count 증가
└─ progress_rate 계산
↓
4가지 퀘스트 모두 100%
↓
quest 업데이트
└─ status (completed)
└─ is_excellent (순차 완료 여부)
└─ completed_at (완료 일시)
↓
coupon 생성
└─ coupon_type (excellent/normal)
└─ issued_at (발급 일시)
└─ status (available)
↓
notification 생성 (쿠폰 발급 알림)

```

6. 비즈니스 로직

6.1 차시 성공/실패 판정 로직

A 단계

```

def check_a_stage_result(section_results):
    # 다시읽기 제한 시간 초과
    if section_results['re_reading']['elapsed_time'] > 160: # 2분 40초
        return 'fail', '다시읽기 제한 시간 초과'

```

```

# 문제확인2 모르는 문제 개수
if section_results['question_check2']['unknown_count'] >= 2:
    return 'fail', '문제확인 모르는 문제 초과'

# 문제풀이 오답+모름
wrong_and_unknown = (section_results['problem_solving']['wrong_count'] +
                      section_results['problem_solving']
['unknown_answer_count'])
if wrong_and_unknown >= 2:
    return 'fail', '문제풀이 오답 문제 초과'

return 'success', None

```

B 단계

```

def check_b_stage_result(section_results, is_retry=False):
    # 지문읽기 제한 시간 초과 (1차)
    if not is_retry and section_results['passage_reading']['elapsed_time'] > 180:
        # 다시읽기 진행
        return 'retry', '지문읽기 제한 시간 초과'

    # 문제확인 모르는 문제
    unknown_limit = 1 if is_retry else 2
    if section_results['question_check']['unknown_count'] >= unknown_limit:
        if is_retry:
            return 'fail', '문제확인 모르는 문제 초과'
        else:
            # 다시읽기 진행
            return 'retry', '문제확인 모르는 문제 초과'

    # 다시읽기 제한 시간 초과 (재시도 시)
    if is_retry and section_results['re_reading']['elapsed_time'] > 170: # 2분 50
    초
        return 'fail', '다시읽기 제한 시간 초과'

    # 문제풀이 오답+모름 (비판 제외)
    wrong_limit = 1 if is_retry else 2
    wrong_and_unknown =
calculate_wrong_count_exclude_critical(section_results['problem_solving'])
    if wrong_and_unknown >= wrong_limit:
        if is_retry:
            return 'fail', '문제풀이 오답 문제 초과'
        else:
            # 다시읽기 진행
            return 'retry', '문제풀이 오답 문제 초과'

    return 'success', None

```

C 단계

```
def check_c_stage_result(section_results, is_retry=False):
    # 지문읽기 제한 시간 초과 (1차)
    if not is_retry and section_results['passage_reading']['elapsed_time'] > 180:
        return 'retry', '지문읽기 제한 시간 초과'

    # 문제확인 모르는 문제 (1개 이상)
    if section_results['question_check']['unknown_count'] >= 1:
        if is_retry:
            return 'fail', '문제확인 모르는 문제 초과'
        else:
            return 'retry', '문제확인 모르는 문제 초과'

    # 다시읽기 제한 시간 초과
    if is_retry and section_results['re_reading']['elapsed_time'] > 180: # 3분
        return 'fail', '다시읽기 제한 시간 초과'

    # 문제풀이 오답+모름 1개 이상 (비판 제외)
    wrong_and_unknown =
calculate_wrong_count_exclude_critical(section_results['problem_solving'])
    if wrong_and_unknown >= 1:
        if is_retry:
            return 'fail', '문제풀이 오답 문제 초과'
        else:
            return 'retry', '문제풀이 오답 문제 초과'

    return 'success', None
```

6.2 자동 패스 로직

```
def check_auto_pass(user_id, lesson_id):
    """
    동일 강 내 3차시 연속 실패 시 자동 패스
    """
    recent_attempts = get_recent_attempts(user_id, lesson_id, limit=3)

    if len(recent_attempts) >= 3:
        all_failed = all(attempt['result'] == 'fail' for attempt in
recent_attempts)
        if all_failed:
            return True # 자동 패스 처리

    return False
```

6.3 진행률 계산 로직

```
def calculate_volume_progress(user_id, license_id):
    """
    권별 진행률 계산
```

```

"""
total_lessons = get_total_lessons(license_id)
completed_lessons = count_completed_lessons(user_id, license_id)

progress_rate = round((completed_lessons / total_lessons) * 100, 0)

return {
    'total_lessons': total_lessons,
    'completed_lessons': completed_lessons,
    'remaining_lessons': total_lessons - completed_lessons,
    'progress_rate': progress_rate
}

```

6.4 신호등 상태 계산 로직

```

def calculate_traffic_light(section_type, elapsed_time, student_avg_time,
target_time):
    """
    구간별 신호등 상태 계산
    """
    if section_type in ['passage_reading', 're_reading']:
        # 학생 평소 읽기 기준
        if student_avg_time:
            ratio = elapsed_time / student_avg_time
            if 0.7 <= ratio <= 1.3:
                return 'normal'
            elif ratio < 0.7:
                return 'fast'
            else:
                return 'slow'

        # 학년별 목표 시간 기준
        if target_time:
            if elapsed_time <= target_time:
                return 'good'
            else:
                return 'exceeded'

    elif section_type == 'problem_solving':
        comprehension = calculate_comprehension_score(section_results)
        if comprehension >= 70:
            return 'good'
        else:
            return 'bad'

    return 'normal'

```

6.5 퀘스트 우수 상품 판정 로직

```
def check_excellent_quest(quest_history):
    """
    순차적으로 완료했는지 판정
    - 추가 활동을 미루지 않고 완료
    """
    lesson_success_dates = get_quest_dates(quest_history, 'lesson_success')
    today_vocab_dates = get_quest_dates(quest_history, 'today_vocab')
    today_summary_dates = get_quest_dates(quest_history, 'today_summary')

    for lesson_date in lesson_success_dates:
        # 강 성공 후 당일 또는 익일 추가 활동 완료 확인
        if lesson_date not in today_vocab_dates and lesson_date not in
today_summary_dates:
            next_day = lesson_date + timedelta(days=1)
            if next_day not in today_vocab_dates and next_day not in
today_summary_dates:
                return False # 미루고 진행함

    return True # 순차적으로 완료
```

7. 기술적 요구사항

7.1 데이터베이스 요구사항

RDBMS: MySQL 8.0+

- InnoDB 엔진
- UTF-8 MB4 인코딩
- DATETIME(6) 마이크로초 지원

인덱싱:

- 복합 인덱스: (user_id, status), (user_id, lesson_id)
- 날짜 인덱스: created_at, started_at
- 외래키 제약조건 적용

JSON 필드 활용:

- section_data (구간별 세부 데이터)
- ingredient_images (퀘스트 재료 이미지)
- weak_vocab_types (취약 어휘 유형)

7.2 성능 요구사항

응답 시간:

- 페이지 로딩: 2초 이내
- 구간 전환: 1초 이내
- 결과 생성: 3초 이내

동시 접속:

- 최소 1,000명 동시 사용자 지원
- 피크 타임 대비 확장 가능

데이터 보존:

- 학습 이력: 영구 보존
- 로그 데이터: 최소 1년

7.3 보안 요구사항

인증/인가:

- 세션 기반 인증
- 학생/관리자 권한 분리

데이터 보호:

- UUID 기본키 (추측 불가)
- SQL Injection 방어
- XSS 방어

개인정보:

- 최소 수집 원칙
- 암호화 저장 (민감 정보)

7.4 확장성 고려사항

수평 확장:

- 읽기 전용 복제본 활용
- 캐싱 레이어 (Redis)
- CDN 활용 (정적 리소스)

파티셔닝:

- 로그 테이블 월별 파티셔닝
- 이력 테이블 연도별 파티셔닝

백업:

- 일일 전체 백업
- 실시간 증분 백업

7.5 모니터링 요구사항

애플리케이션 모니터링:

- 에러 추적 및 알림
- 성능 메트릭 수집
- 사용자 행동 분석

인프라 모니터링:

- DB 성능 모니터링

- 서버 리소스 모니터링
- 네트워크 트래픽 모니터링

참고 문서

- 원본 기획서: [icks_literacy_plan_20251020.pdf](#)
- 기능 정의서: [02_기능_정의서.md](#)
- DB 설계서: [03_DB_설계서.md](#)

문서 변경 이력

- v1.0 (2025-10-23): 초안 작성