

Peng'sQL

오픈소스 전문프로젝트 최종 보고서

Team 2

2025. 6. 20

보고서 작성자

윤하민

보고서 검수자

김태영 전설민 박서현

목차

1. 아이디어 기술서
2. pangsql 확장 기술
3. 요구사항
4. 포기된 기능들
5. 추가된 요청들
6. 실제로 구현된 기능
7. 개발 방법론
8. 일정과 출시
9. 최종 평가

1. 아이디어 기술서

아이디어 명	sqlite+db (가제)			
제안자	◀ 기업 ▶ 교수 <input checked="" type="checkbox"/> 학생			
	성명	윤하민	회사명/소속학과	소프트웨어 학과
	전화번호	010-3425-2314	e-Mail	ynhamin@gmail.com

팀원	성명	소속학과(학년)	전화번호	e-Mail
	윤하민	소프트웨어학부(3)	010-3425-2314	ynhamin@gmail.com
	김태영	소프트웨어학부(3)	010-5142-7737	rlaxodud7737@naver.com
	전설민	소프트웨어학부(3)	010-5120-9502	jsm041046@gmail.com
	박서현	소프트웨어학부(3)	010-4552-1557	psh_0226@naver.com

제안 배경	<p>1. 유저 친화적 UI 대부분의 DB 관련 제품들은 복잡하고 불편한 UI를 가지고 있습니다. 물론 DB 특성상 데이터를 테이블 형태로 표현해야 하는 한계점을 지니고 있지만(엑셀 형태), 개선할 수 있는 방안은 다양하게 존재한다고 생각합니다. 직관적이면서도 사용자 친화적인 디자인이 있으면 좋을 거 같습니다.</p> <p>2. DB 설계 자원의 부족 대부분의 제품은 DB 조회 및 편집 기능에 집중되는 반면, 설계와 관련된 앱은 소수에 불과합니다. 즉 시 떠오르는 아이디어를 빠르게 구조화할 수 있는 도구도 필요하지만, 이를 효과적으로 지원하는 자원이 부족하다는 단점이 있었습니다.</p> <p>3. 한국어 지원 부족 기존 앱들의 리뷰를 조사한 결과 한글화 요청에 대한 리뷰가 상당수를 차지하였습니다. 기존의 앱들은 대개 한국어 지원이 부족한 경우가 많았기에, 영어와 한국어를 기본 지원한다면 사용자의 요구를 충족시킬 수 있다고 생각합니다.</p> <p>4. 커뮤니티 지원 DB는 다른 소프트웨어에 비해 오류의 발생 빈도가 낮습니다만, 문제가 발생하게 되면 피해의 크기는 매우 큼니다. DB 관련 개발자 커뮤니티를 만들어 문제와 해결 방안을 공유할 수 있다면, 안정적인 개발이 가능할 것으로 생각합니다.</p>
아이디어 개요	<p>1. SQLite 에디터 숙련자를 위한 CLI 환경을 제공함과 동시에, 터치 방식으로 자료형과 자료값을 조작할 수 있도록 확장성을 고려하여 설계합니다. 대부분의 앱에서 경시되던 '커밋' 및 '롤백' 기능을 활용하여 사용자 오류를 최소화합니다. 또한, 임의 데이터 생성 및 테스트 자료 생성 모듈을 제공하여 보다 적극적이고 실험적인 DB 설계를 가능하게 할 것입니다.</p> <p>2. DB 다이어그램 설계를 위한 기초적인 DB 스키마 다이어그램뿐만 아니라, 현재 구상 중인 DB 설계를 지원하기 위해 기존 DB의 테이블을 다이어그램화하는 모듈을 추가할 예정입니다. DB 설계를 위해 직접 다이어그램을 만들지 않고, 테스트로 만든 DB를 클릭 한 번으로 다이어그램으로 변환할 수 있도록 합니다.</p> <p>3. 개발자 커뮤니티</p>

	<p>자료집과 참고사례를 추가하여 사용자 간 논의를 활성화합니다. 개발자들과 소통할 수 있는 경로를 마련하여 긍정적인 피드백 루프를 형성하는 것이 주요 목표입니다. 기본적인 사용자 서비스 시스템이 적용되며, 필요할 경우 이를 위한 서버 자원도 활용될 것입니다.</p> <p>4. 가로 모드를 중심</p> <p>모바일 화면에서 많은 정보를 효율적으로 볼 수 있도록 기존 앱과는 달리 가로 모드를 중심으로 설계할 것입니다.</p>																																
관련 기존 제품(서비스)들과의 차별성	<p>기존 제품</p> <table><tr><th>이름</th><th>다운로드</th><th>기능</th><th>특징</th></tr><tr><td>DBViewer - MS Access Viewer</td><td>5만</td><td>뷰어</td><td>시장 내 일반적 뷰어 수준</td></tr><tr><td>SQLite Viewer</td><td>1만+</td><td>뷰어</td><td>기본적인 DB 조회 기능 제공</td></tr><tr><td>PortoDB Database</td><td>10만+</td><td>에디터</td><td>이미지 처리 기능 포함</td></tr><tr><td>RemoDB SQL Client (MySQL, MsSQL)</td><td>10만+</td><td>에디터</td><td>최신 업데이트(2025. 3. 2.)</td></tr><tr><td>SQLPhone: SQL Interpreter</td><td>10만+</td><td>에디터</td><td>CU 형태로 학생 공부용 개발됨</td></tr><tr><td>Database Designer</td><td>10만+</td><td>스키마</td><td>뛰어난 스키마 다이어그램 기능 제공</td></tr><tr><td>SQLite Database Editor</td><td>50만+</td><td>에디터</td><td>업계 최대 다운로드, UI가 불편함</td></tr></table>	이름	다운로드	기능	특징	DBViewer - MS Access Viewer	5만	뷰어	시장 내 일반적 뷰어 수준	SQLite Viewer	1만+	뷰어	기본적인 DB 조회 기능 제공	PortoDB Database	10만+	에디터	이미지 처리 기능 포함	RemoDB SQL Client (MySQL, MsSQL)	10만+	에디터	최신 업데이트(2025. 3. 2.)	SQLPhone: SQL Interpreter	10만+	에디터	CU 형태로 학생 공부용 개발됨	Database Designer	10만+	스키마	뛰어난 스키마 다이어그램 기능 제공	SQLite Database Editor	50만+	에디터	업계 최대 다운로드, UI가 불편함
이름	다운로드	기능	특징																														
DBViewer - MS Access Viewer	5만	뷰어	시장 내 일반적 뷰어 수준																														
SQLite Viewer	1만+	뷰어	기본적인 DB 조회 기능 제공																														
PortoDB Database	10만+	에디터	이미지 처리 기능 포함																														
RemoDB SQL Client (MySQL, MsSQL)	10만+	에디터	최신 업데이트(2025. 3. 2.)																														
SQLPhone: SQL Interpreter	10만+	에디터	CU 형태로 학생 공부용 개발됨																														
Database Designer	10만+	스키마	뛰어난 스키마 다이어그램 기능 제공																														
SQLite Database Editor	50만+	에디터	업계 최대 다운로드, UI가 불편함																														
개발 방안	<p>1. 강력한 모듈화 구조</p> <p>프로젝트에서 독자적으로 존재할 수 있는 기능은 일단은 분리한다는 원칙을 둔다. 주요 기능은 싱글톤으로 가장 베이스 레이어로 만든다. 기능을 모듈화를 통해 구현하고, 모듈의 확장을 종속 모듈로 분리한다. 종속 모듈은 많은 부모 모듈을 가질 수 있지만, 문제는 언제나 종속 모듈에서 터지게 만든다.</p> <p>2. 언어 체인</p> <p>JAVA (기능) - Kotlin (중간층) - XML 각 인원이 강점이 있는 언어 존재하기에 제시된 강력한 프로젝트 모듈화의 강점을 살려 기능 모듈 개발은 JAVA로 XML 과 연동과 언어 스트링 처리는 Kotlin을 통해 언어 체인을 형성하고 개발한다.</p> <p>3. 프로토타입 개발</p> <p>모듈 개발자들은 디자인의 정체나 선행 구현 요소에 발이 묶이지 않도록 프로젝트 진행 관리자에 의해 '일단 구현'이라는 목적하에 프로토타입 개발 상태로 전환한다. 이 상태에서는 정해진 요소가 해결될 때까지 구조를 설계하고 도전적인 설계 및 실험을 진행한다.</p> <p>4. 업무 분장</p> <p>김태영 팀장 / 총괄 개발, 프론트 엔드 책임자 윤하민 조원 / 기능 개발자, 프로토타입 책임자 전설민 조원 / 기능 개발자, 서버 개발자 박서현 조원 / 프론트 엔드, 디자인 개발자</p>																																
기대 효과	<p>1. 모바일 DB 에디터의 접근성 향상</p> <p>사용자는 모바일에서도 손쉽게 DB를 설계하고 수정할 수 있으며, PC 환경에서만 가능했던 작업을 언제 어디서든 할 수 있게 됩니다.</p>																																

2. 개발자 커뮤니티와의 연계

오픈소스로 제공함으로써 광고 없이 사용자와 커뮤니티와의 활발한 소통을 통해 제품을 개선하고, 시장에 적합한 기능을 지속적으로 추가할 수 있습니다.

3. 국내외 사용자에게 유용한 도구 제공

한국어 지원을 포함하여 국내외 사용자들에게 편리한 DB 브라우징 및 편집 도구를 제공하여, DB 작업의 접근성을 크게 향상 시킬 수 있습니다.

2. pangsql 확장 기술

1. pangsql은 무엇인가

2. 왜 pangsql을 만들었는가

2.1. 왜 pangsql이 필요한가

2.2. 시장 상황과 가능성

3. 약속된 주 기능들

3.1. 디자인 정책

3.2. 뷰어 에디터

3.3. 다이어그램

3.4. 커뮤니티 지원 자원

3.5. 비주얼 모듈 플러그인

pangsql 확장 기술

1. pangsql은 무엇인가

Pangsql은 안드로이드 앱에서 시행되는 SQLite3 db를 위한 뷰어, 에디터, 플래너, 분석 툴로 안드로이드 환경에 특화된 유틸리티 툴입니다. 또는 안드로이드 상의 db 브라우저입니다.

2. 왜 pangsql을 만들었는가

2.1. 왜 pangsql이 필요한가

기존의 대부분 DB 관련 앱들은 복잡하고 불편한 UI를 가지고 있으며, 데이터 구조상 테이블 형태의 표현에 제한이 따르기는 하지만, 엑셀과 유사한 직관적이고 사용자 친화적인 인터페이스로 충분히 개선할 수 있는 여지가 있다고 생각합니다. 또한 DB 다이어그램 기능을 통해 기존 테이블을 시각적으로 변환하거나, 설계 중인 구조를 쉽게 테스트하고 다이어그램화할 수 있도록 하며, 개발자 커뮤니티를 통해 피드백과 자료를 순환시켜 사용자 참여를 유도할 것입니다. 마지막으로, 모바일 환경에서도 많은 정보를 효율적으로 표현할 수 있도록 가로 모드를 중심으로 화면을 설계할 예정입니다.

2.2. 시장 상황과 가능성

현재 시장에서는 DB의 조회 및 편집 기능에 집중된 제품이 많고, 반면 떠오른 아이디어를 구조화할 수 있는 설계 도구는 매우 부족한 실정입니다. 특히, 한글화에 대한 사용자 리뷰가 많았던 것을 감안하면, 영어와 한국어를 기본적으로 지원하는 기능은 사용자의 만족도를 크게 높일 수 있을 것으로 보입니다. 여기에 더해, DB 분야는 다른 소프트웨어에 비해 오류 발생률은 낮지만 한 번 문제가 생기면 큰 피해로 이어지므로, 관련 개발자 커뮤니티를 구축하여 문제 해결 및 노하우 공유가 가능한 안정적인 개발 생태계를 마련하는 것이 중요합니다. 이를 바탕으로 본 프로젝트는 SQLite 에디터를 통해 숙련자뿐만 아니라 초보자도 터치 기반으로 자료형과 값을 조작할 수 있도록 하고, 커밋 및 롤백 기능을 활용해 사용자 오류를 최소화하며, 임의 데이터 생성 기능을 제공해 실험적 설계를 지원합니다.

3. 약속된 주 기능들

PangSQL은 단순한 데이터 조회 도구가 아닌, 설계, 편집, 시각화, 커뮤니티 지원까지 아우르는 종합적인 DB 도구를 목표로 개발되고 있습니다. 각 기능은 실 사용자들의 실제 문제와 요구에 대응하기 위해 구체적으로 설계되었으며, 특히 모바일 환경에서의 직관성과 접근성을 고려하여 확장성과 활용성을 동시에 확보하는데 중점을 두었습니다.

3.1.

PangSQL은 엑셀과 유사한 구조로 데이터를 다루면서도, 일반적인 테이블 UI의 한계를 극복하고자 직관적이고 사용자 친화적인 인터페이스를 지향합니다. 복잡하고 난해한 구조로 인해 초보자들이 접근하기 어려웠던 기존 앱들과는 달리, 명확한 시각적 구분과 미니멀한 설계 원칙을 통해 데이터 구조를 더 쉽게 이해하고 조작할 수 있도록 합니다. 터치 기반 UI와 데스크탑 환경을 모두 아우를 수 있는 유연한 설계는 다양한 사용자의 니즈를 수용하고, 더 빠르고 편리한 설계 과정을 가능하게 할 것입니다.

3.2. 뷰어 에디터

숙련자를 위한 CLI 환경을 제공하는 동시에, 터치 기반의 조작을 지원하여 초보자도 손쉽게 자료형과 값의 수정이 가능하도록 합니다. PangSQL은 대부분의 앱에서 경시되던 커밋 및 롤백 기능을 명시적으로 제공함으로써, 작업 중 실수를 최소화하고 데이터 안정성을 높입니다. 여기에 더해 임의 데이터 생성 및 테스트용 자료 생성 기능을 탑재하여, 실험적이고 유연한 DB 설계를 적극적으로 지원합니다. 단순히 데이터를 보여주고 편집하는 수준을 넘어, 개발과 설계 과정 전체를 견인하는 에디터 역할을 수행합니다.

3.3. 다이어그램

PangSQL은 설계를 위한 기초적인 DB 스키마 다이어그램 기능은 물론, 현재 구상 중인 데이터 구조를 시각적으로 정리하고 추적할 수 있도록 지원합니다. 직접 다이어그램을 그리지 않더라도, 임시로 생성한 DB 테이블을 클릭 한 번으로 다이어그램화할 수 있어, 꼭 필요하지만 많은 시간이 소요되던 과정을 획기적으로 줄여줍니다. 이를 통해 설계자는 빠르게 구조를 시각화하고, 그 내용을 동료나 커뮤니티와 공유하는 과정을 더욱 효율적으로 수행할 수 있습니다.

3.4. 커뮤니티 지원 자원

DB는 오류 발생 빈도는 낮지만, 일단 문제가 생기면 손실이 크고 해결까지 오랜 시간이 걸릴 수 있습니다. PangSQL은 이러한 문제를 고려하여, 사용자들이 서로의 문제 해결 경험을 공유하고, 기능적 요구나 개선안을 논의할 수 있는 커뮤니티 플랫폼을 함께 제공합니다. 자료집과 참고사례, 자주 묻는 질문 등을 통해 사용자들이 단순한 도구 사용을 넘어, 실질적인 학습과 상호 지원을 경험할 수 있도록 설계할 예정입니다. 이 커뮤니티는 안정적인 DB 개발 환경을 조성하는 핵심 축이 될 것입니다.

3.5. 비주얼 모듈 플러그인

PangSQL은 단순한 뷰어 수준을 넘어, 데이터를 분석하고 시각화할 수 있는 비주얼 모듈을 지원합니다. 단순한 텍스트 형태의 데이터를 유의미한 정보로 전환할 수 있도록 다양한 분석 도구를 통합하고, 이를 기반으로 시각화 결과를 바로 확인할 수 있는 인터페이스를 제공합니다. 사용자는 모바일 기기에서 DB 파일만 열어도 다양한 분석 결과를 즉시 확인할 수 있으며, 이는 전통적인 데스크탑 기반 분석 환경을 넘어서는 유연성과 접근성을 부여합니다. PangSQL은 데이터 분석가, 디자이너, 개발자 모두를 위한 범용 도구로 확장될 것입니다.

3. 요구사항

1. 외부 요구사항 자원

1.1. 분석 다이어그램

2. 구현 요구사항 구조

1. 외부 요구사항 자원

개발 프로젝트의 순항을 위해서 설계적 안정성 및 검증을 위한 분석이 필요했고 이를 검증할 수 있는 가용 도구와 자원을 동원하였다.

외부 요구사항 분석

- 요구사항 명세 -

1. 소개

1-1. 목적

안드로이드 환경에서 원활하게 sqlite DB를 기획 및 수정, 관리 나아가 분석이 가능한 어플리케이션 개발을 위한 요구사항을 서술한 문서입니다.

1-2. 범위

사용자가 .db 파일과 .dbia 파일을 손 쉽게 편집 관리 할 수 있습니다.

1-3. 정의 및 약어

- 사용자: 일반 사용자
 - Sysops : 어플리케이션 메인 시스템
-

2. 전반적인 설명

2-1. 제품 관점

본 시스템은 독립적으로 실행되는 앱이며, sqlite를 지원하는 모든 안드로이드 환경에서 사용가능하다.

2-2. 사용자 특성

- 사용자: 기본적인 웹 사용 가능

2-3. 제약 조건

- 데이터는 SQLite DB에 저장
 - 시스템은 영어와 한국어를 지원해야 함
 - 구글 앱 스토어 정책 준수
-

3. 기능적 요구사항 (Functional Requirements)

3-1. sysops 기본 기능

- 사용자가 파일을 추가할 수 있다(.db .dbia 등록 가능)
- 즐겨찾기에서 선택된 .db를 에디터로 연다
- 즐겨찾기에서 선택된 .dbia를 다이어그램 컴파일러로 연다
- 새로운 db 생성 시 내부에 임시 파일 생성후 해당 .db를 에디터로 연다
- 새로운 dbia 생성 시 내부에 임시 파일 생성후 해당 .dbia를 다이어그램 컴파일러로 연다

3-2. 뷰어에디터

- 데이터베이스 객체 (테이블, 뷰, 인덱스, 트리거) 는 시각화 된다
- 데이터베이스 객체를 생성, 수정, 삭제 할 수 있다
- 사용자는 현재 에디터의 db를 커밋/롤백 할 수 있다

3-3. 다이어그램

- cli를 통해 .dbia를 수정할 수 있다
- dbia에 수정이 있다면 컴파일한다
- 그리드 좌표계에 테이블 자료를 업데이트 한다
- 사용자는 현재 진행사항을 저장할 수 있다

3-4. 모듈 관리자

- 등록된 모듈을 적용할 수 있다

3-5. 분석 모듈 A (정규분포)

- 테이블 자료에 정규분포 공식을 적용한다
- 좌표계에 정규분포 도표를 제공한다
- 두가지 이상의 서로 다른 정규분포 도표를 시각화 할 수 있다

3-6.. 분석 모듈 B (지도)

- 위치 정보를 가진 테이블 자료가 지도에 핀을 찍는다
- 두 개 의 테이블 자료를 사용해 위치 정보에 서로 다른 색의 핀을 찍는다

3-7. 분석 모듈 C (도표화)

- 사용자는 자료 구분 기준을 작성할 수 있다
 - 테이블 자료는 구분 기준에 따라 정렬되며 시각화된다
-

4. 비기능적 요구사항 (Non-Functional Requirements)

1. Performance Requirement

- 시스템은 24시간동안 작동할 수 있어야 한다.
- CPU 사용률은 70%를 넘지 않아야 한다
- 시스템은 초당 5,000건 이상의 요청을 처리할 수 있어야 한다.

다이어그램 CLI

- UI 업데이트가 2초 이내 되어야 한다
- 즐거찾기
- 선택된 파일이 앱에서 지원하지 않는 경우 예외한다

뷰어에디터

- 테이블 자료에 페이징을 제공한다

2. Security & Compatibility Requirement

뷰어에디터

- 자료형 제약조건 체크를 제공한
- 구글 플레이스토어 정책을 준수한다 saf 접근
- 앱 삭제시 앱내 모든 db 자료를 삭제한다
- 다른 하드웨어의 외부 접근은 하지않는다
- 접근권한이 없는 db를 접근하면 안된다
- 안드로이드 환경에서 작동해야 한다

3. Cultural Requirement

- 한글 및 영어를 지원한다

4. Usability Requirement

즐거찾기

- 추가된 파일은 하나의 리스트에서 최근 접근 순으로 나열된다

5. Operational Requirement

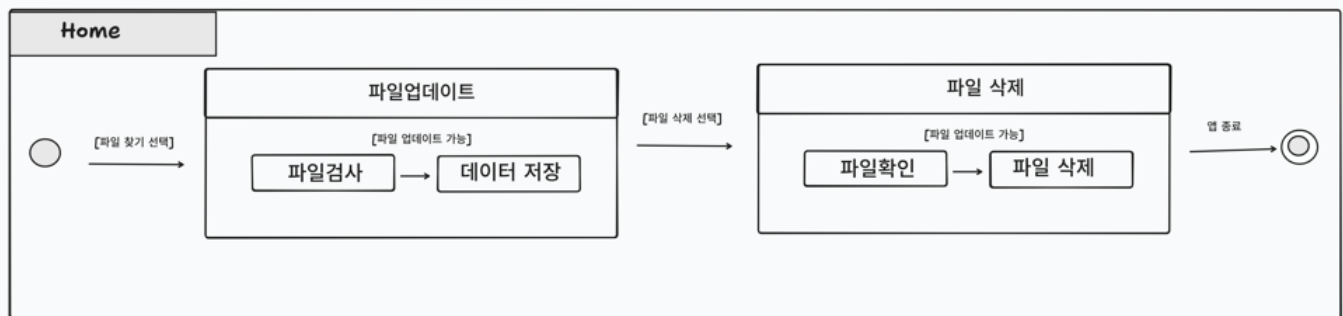
- 안드로이드에서 작동하는 앱이다
-

5. 인터페이스 요구사항

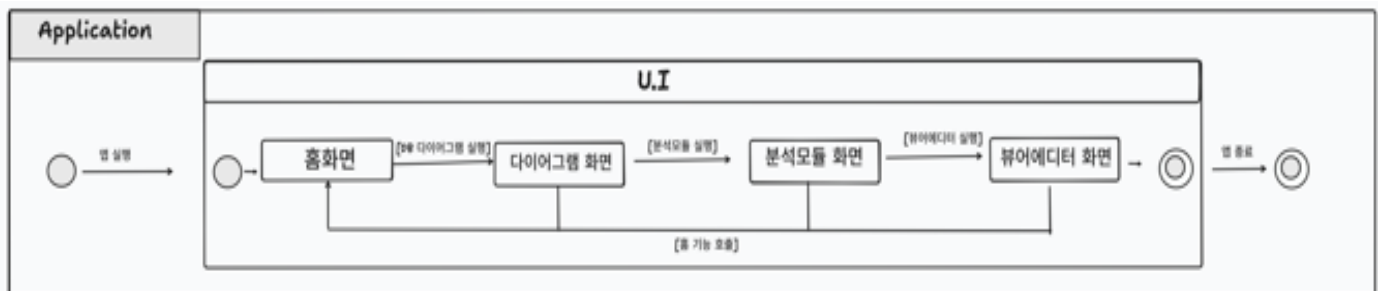
- 사용자 인터페이스: 안드로이드 모바일 환경
 - 데이터베이스 인터페이스: SQLite
 - API: 지도 api
-

1.1. 분석 다이어그램

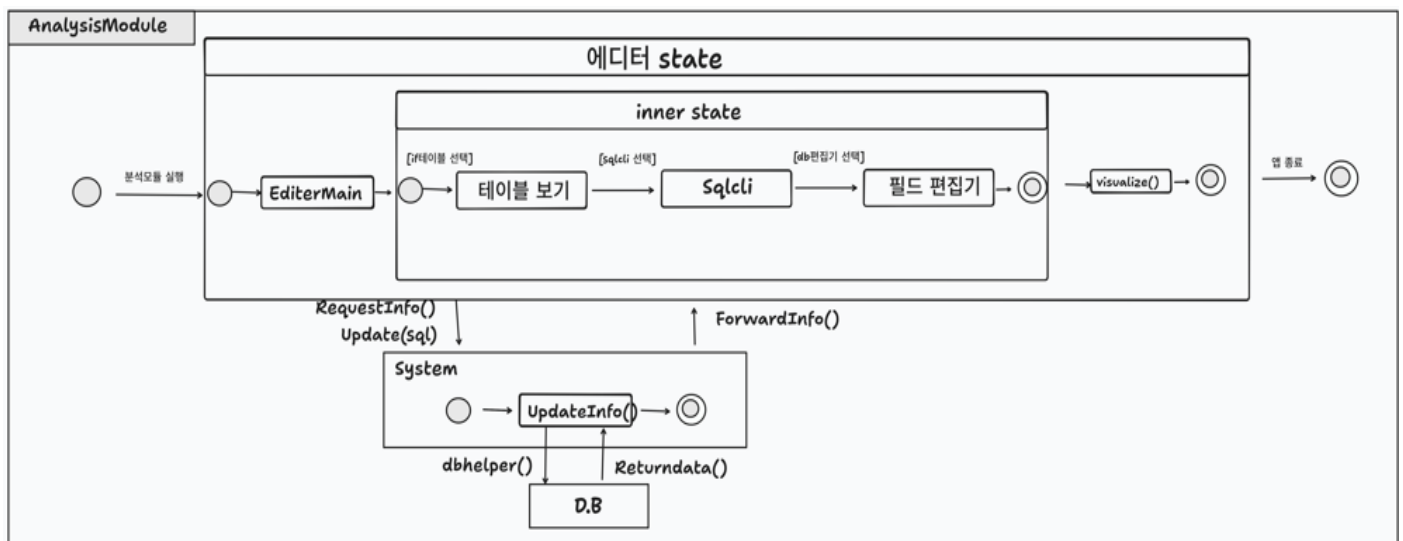
파일 시스템



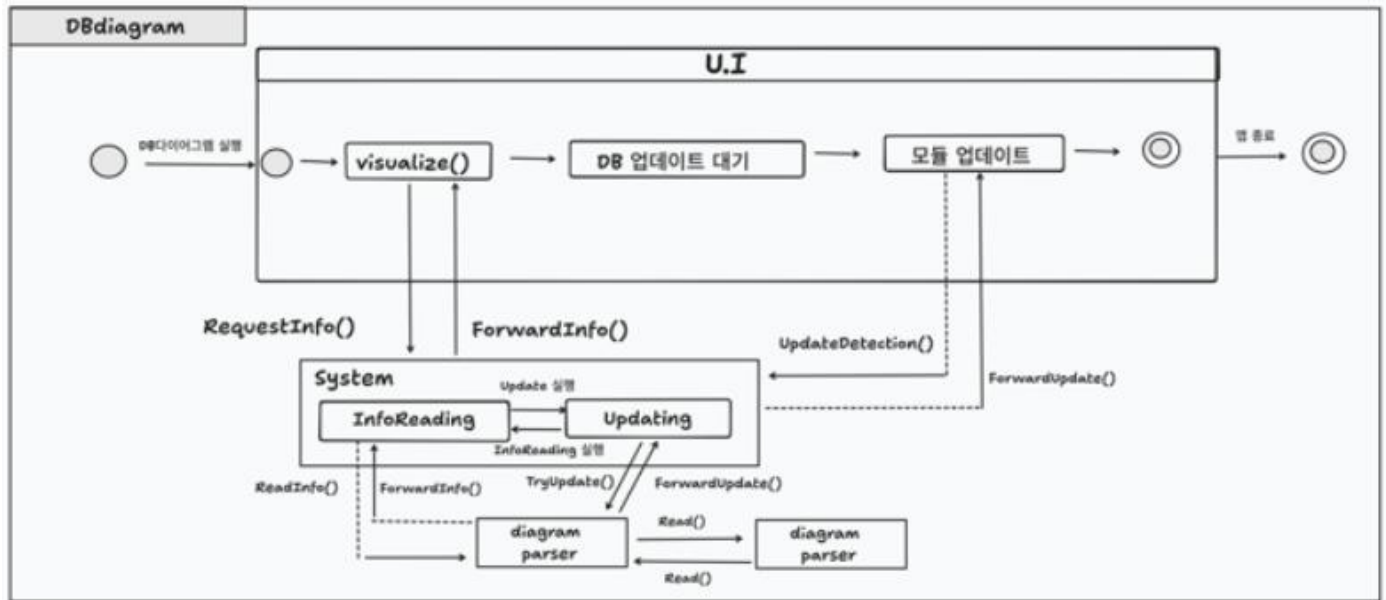
뷰어 에디터 접근



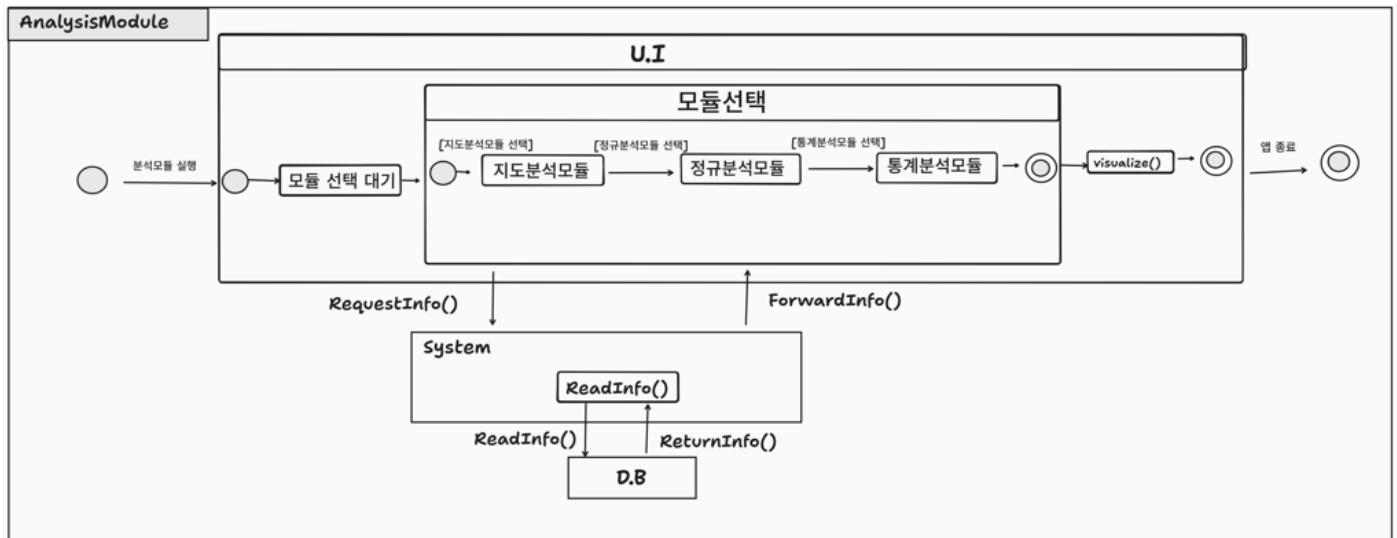
뷰어 에디터와 분석 모듈 접근



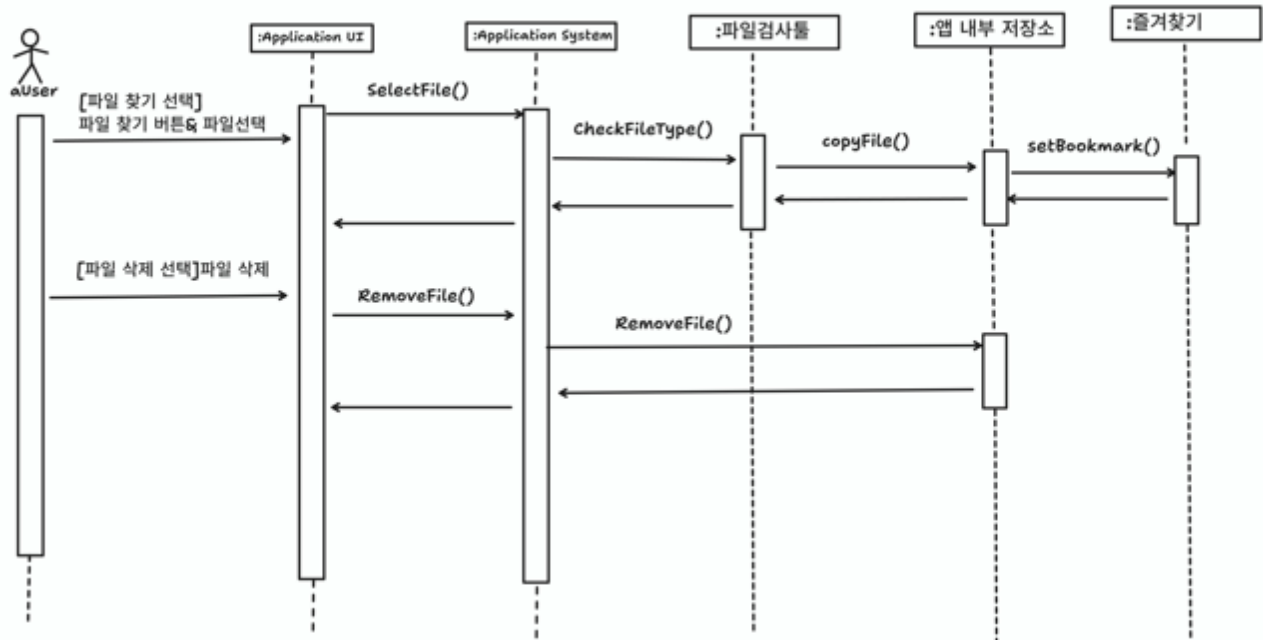
다이어그램



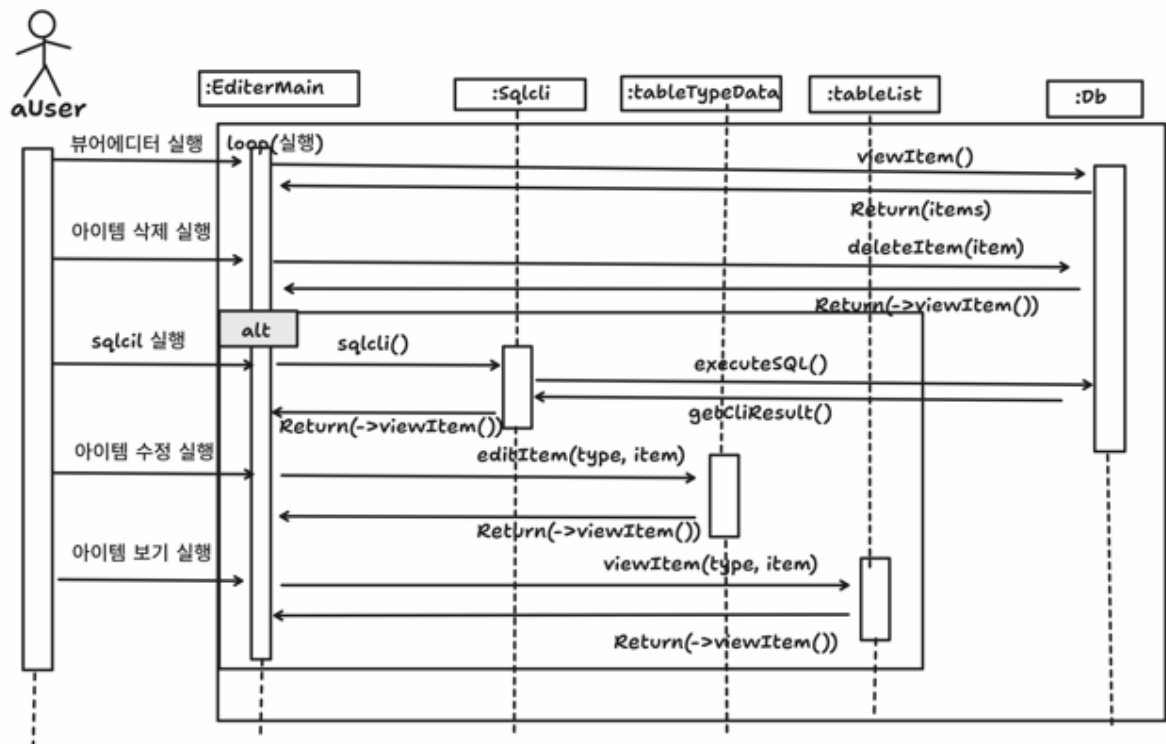
분석 모듈



sd Home



sd viewerEditor



2. 개발 요구사항 구조

+sqlite 앱 에디터

[외부 종속]

1. 파일 수정 권한 요청
2. db 파일 탐색 선택

[필수] (싱글톤 의견)

1. 실질적 최종 실행 sql
2. 유저 타이핑 sql 창
타이핑 창
결과창
3. commit의 중요성
 - 프로젝트 저장하기 이전 버전 유지해야함
 - 저장 터치 시 확장 commit
 - 롤백
4. 잠금 접근
 - 비밀번호가 걸린 SQLite DB 파일을 열 수 있는 기능.

[모듈]

1. db 생성 삭제
2. 구조 자체 생성/삭제/관리
 - table UX/UI
 - view UX/UI
 - index UX/UI
 - trigger UX/UI
 - 예외처리 동작 제한
3. 구조 내 데이터 생성/삭제/관리
 - table UX/UI
 - view UX/UI
 - index UX/UI
 - trigger UX/UI
 - 데이터 수정
 - 페이징
 - 테이블 전환
 - 정렬 (안드로이드 자체 data table
 - 정렬 기능 존재시 정렬은 이곳에 맡김)
 - ui 쪽에서 특정 셀을 선택시 기능추가
 - 수정 삭제 null 변경
 - 해당 셀 복사
4. 대용량 sql.txt -> sql 처리 시도
5. import/export

+개발자 커뮤니티

서버시 계층(컴에서 돌려도 됨)

-서버 계층

-db 계층

유저계층

로그인 계층

개인 계정

통합 게시판

포스트

내용

댓글

좋아요

북마크

공유하기

+db 다이어그램

db 불러오고 시각화 해야함

특수키 관계도 정리

블록화

그리드

(제외)스키마 수정시 db 갱신

=>불러오는 것 까지만 하고 편집 하는 수준으로 다시 고려

기능 개요

SQLite 기반의 뷰어·에디터는 단순한 데이터 조회에 그치지 않고, 사용자가 직접 데이터를 설계하고 조작할 수 있는 종합적 환경을 제공하는 것을 목표로 한다. 이를 위해 기본적인 데이터 입출력, 구조 조작, 커밋/롤백 기반의 버전 제어 기능과 더불어, 대용량 SQL 처리, 시각화, 자동완성, 문서화 등 다양한 모듈이 통합되었다. 뷰어·에디터는 파일 접근 권한 요청과 같은 외부 종속성을 고려하며, 사용자의 실수를 방지하고 설계를 돕는 다양한 UI/UX 기능을 포함하고 있다. 모두 구현되지는 않았지만 기반하였던 내용은 다음과 같다.

외부 종속 기능

에디터의 작동을 위해 반드시 확보되어야 하는 외부 기능으로는 다음과 같은 사항이 있다. 첫째, 파일 접근 권한 요청을 통해 SQLite 파일을 탐색하고 열 수 있어야 하며, 사용자가 직접 DB 파일을 탐색해 선택하는 UI도 제공되어야 한다. 둘째, 비밀번호가 걸린 SQLite 파일을 열 수 있는 기능도 함께 구현되어야 한다. 이는 보안이 요구되는 환경에서의 유연한 대응을 위한 것이다.

기본 SQL 인터페이스

에디터의 핵심 기능은 실질적인 SQL 실행 환경 제공에 있다. 이는 두 부분으로 구성된다. 첫째, 사용자가 SQL을 직접 입력할 수 있는 타이핑 창을 제공하고, 둘째, 입력된 SQL 결과를 확인할 수 있는 결과 창을 함께 배치하여 직관적인 작동 흐름을 만든다. 실질적으로 실행된 SQL과 유저 입력 SQL의 구분은 명확히 하되, 싱글톤 기반으로 SQL 처리를 관리하는 설계도 고려된다.

커밋 및 롤백 지원

SQLite의 강점 중 하나인 커밋/롤백 기능을 활용하여, 사용자는 작업 중 실수를 최소화하고 이전 상태로의 복구가 가능한 환경을 갖게 된다. 사용자가 명시적으로 저장을 선택할 경우에는 내부적으로 커밋이 발생하며, 저장되지 않은 변경 사항은 롤백을 통해 쉽게 초기화할 수 있다. 이 기능은 프로젝트 수준에서의 데이터 안정성과 버전 관리를 지원한다.

DB 구조 관리 기능

사용자는 에디터 내에서 DB 자체를 생성하거나 삭제할 수 있으며, 그 내부 구조인 테이블, 뷰(View), 인덱스(Index), 트리거(Trigger) 등을 자유롭게 생성, 삭제, 관리할 수 있다. 각각의 구조 요소는 전용 UI/UX 인터페이스를 통해 시각적이고 직관적으로 접근할 수 있도록 설계되며, 특히 트리거나 인덱스 등에서 발생할 수 있는 예외에 대한 동작 제한 및 안내도 함께 제공된다.

데이터 편집 기능

DB 구조 내의 각 테이블과 뷰에서 데이터 편집 기능이 제공되며, 이를 위해 페이징 처리, 테이블 전환, 정렬 기능 등을 포함한다. 안드로이드의 기본 데이터 테이블 구조를 활용하되, 정렬 등은 OS 레벨의 최적화를 우선시한다. 사용자는 특정 셀을 선택하여 해당 셀의 값을 수정하거나 삭제할 수 있으며, null 값으로 초기화하거나 셀 값을 복사하는 기능도 함께 제공된다.

대용량 SQL 처리 및 입출력

사용자는 외부의 .sql 또는 .txt 파일을 읽어들이어 대용량 SQL을 한 번에 처리할 수 있다. 또한 import/export 기능을 통해 외부 데이터를 삽입하거나, 기존 DB를 백업 형태로 추출할 수 있도록 한다. 이는 테스트 및 생산 환경 간의 원활한 전환을 지원한다.

종속 모듈 및 보조 기능

SQL 에디터의 데이터 보기 기능에 종속된 서브 모듈들도 함께 구현된다.

첫째, 서식 관련 기능을 통해 글꼴, 문자 정렬, 테이블 정렬 방식 등을 설정할 수 있으며, 둘째, 문서화 및 출력 기능을 통해 DB 데이터를 문서 형태로 변환하거나 출력할 수 있도록 지원한다. 셋째, 검색 및 필터 기능은 데이터 보기 화면에서 텍스트 기반의 빠른 필터링을 지원하며, 마지막으로 SQL 자동완성과 힌트 기능은 타이핑 창에서 사용자의 작성 편의를 높인다.

개발자 커뮤니티 통합

에디터는 단일 기능을 넘어, 개발자 간 커뮤니티를 직접 통합하여 운영하는 구조를 가진다. 이를 위해 서버 계층, DB 계층을 포함한 서비스 인프라가 구성되며, 사용자 계층에서는 로그인 기능과 개인 계정, 통합 게시판, 포스트 작성 및 피드백 시스템이 작동한다. 각 포스트에는 댓글, 좋아요, 북마크, 공유하기 등 기본적인 커뮤니케이션 기능이 포함된다. 이 커뮤니티는 단순한 Q&A를 넘어, 사용자 간 피드백 루프를 통한 기능 발전의 원천으로 작용할 것이다.

DB 다이어그램 시각화

PangSQL은 현재 사용 중인 DB의 구조를 시각화하는 기능도 제공한다. 사용자는 불러온 DB를 기반으로 테이블 간의 특수키(외래키 등) 관계도를 시각적으로 정리할 수 있으며, 이를 그리드 기반의 블록 형태로 표현함으로써 이해와 공유가 용이하게 된다. 다이어그램은 설계의 명확화를 위한 도구이며, 스키마 수정 시 자동으로 갱신되는 기능은 현재 제외된다.

분석 모듈

PangSQL은 단순한 DB 편집 도구를 넘어, 데이터를 가공하고 시각적으로 해석할 수 있는 분석 모듈을 함께 제공한다. 이 모듈은 기본 데이터를 단순히 보여주는데 그치지 않고, 분석 도구로서의 기능까지 수행할 수 있도록 설계되어야 하며, 사용자는 DB 데이터를 기반으로 다양한 시각화 방식(그래프, 차트 등)을 선택하여, 데이터를 보다 직관적으로 파악할 수 있도록 그래픽 기능(혹은 이에 준하는 도표)로 표현되어야 한다.

4. 포기된 기능들

1. 커뮤니티 지원 자원
2. 비주얼 모듈 플러그인

1. 커뮤니티 지원 자원

커뮤니티 자원을 기반으로 한 사용자 간 상호 피드백 기능은 분명히 가치 있는 기능이었지만, 서버 구축 및 관련 자원 탐색에 필요한 비용과 인력 투입이 사실상 별도 프로젝트 수준으로 요구되었다. 초기 개발 단계에서 제한된 리소스를 해당 영역에 배분하기엔 무리가 있었고, 핵심 기능 구현에 집중하기 위해 해당 항목은 우선 제외되었다. 향후 별도의 확장 모듈 또는 독립적인 서비스로 고려될 수는 있으나, 본 프로젝트의 범위 내에서는 개발되지 않았다.

2. 비주얼 모듈 플러그인

데이터를 시각적으로 분석하고 가공할 수 있는 비주얼 모듈 역시 초기에 계획되었고, 뷰어 에디터와 다이어그램 기능이 완성된 이후, 우선 개발 대상으로 우선순위가 매겨졌던 기능이다. 그러나 개발 상호 충돌 처리 구조(플러그인 기능)에 대한 연구가 필요했고, 시각화 대상 자료 선택 방식, 지원 타입 정의, 시각화 결과의 UI 통합 등에서 모듈 간 연계에 필요한 설계와 테스트 비용이 상당하다는 판단이 내려졌다. 또한 더 현대적인 요소(AI)를 고려하지 못한다는 앱의 비판은 초기부터 이어져오고 있었기 때문에 구성되었던 모듈 중 하나가 본 기능으로 이전되었고, 본 모듈은 공식적으로 중단되었다.

5. 추가된 요청들

1. 출시 요청

2. MCP 모듈

1. 출시 요청

본 프로젝트는 단순한 프로젝트용이 아니라, 실제 사용자 배포와 공개 출시 가능성을 전제로 개발되었다. 이에 따라 구글 플레이 스토어 출시에 필요한 정책 준수, 가이드라인 분석, 개인정보 및 보안 요소 점검, 내부 검토 등 다양한 외부 요구 사항이 새롭게 포함되었다. 이러한 작업들은 직접적인 기능 개발과는 별도로 상당한 시간과 자원을 소모하게 되었으며, 최종적으로 앱 배포를 위한 법적·기술적 적합성 확보가 하나의 주요 과제가 되었다

2. MCP 모듈

프로젝트 초기부터 제기된 현대적 기술 요소 부족에 대한 비판은, 단순히 편집기 수준을 넘어서 이제 상용화되는 AI 기반의 사용자 요구를 반영한 것이었다. 이를 수용하여 본 프로젝트에는 MCP(Module Context Protocol) 모듈이 추가되었다. 이 모듈은 생성형 인공지능과의 연동을 통해, 사용자가 SQL 쿼리를 직접 작성하지 않더라도 AI가 자료 구조를 파악하고, 자동으로 쿼리를 생성하거나 수정 제안을 할 수 있는 인터페이스를 제공한다. MCP는 단순 자동화 도우미를 넘어, 사용자와 인공지능이 함께 작업을 구성할 수 있는 차세대 인터랙션 구조의 기반으로 작용할 예정이다.

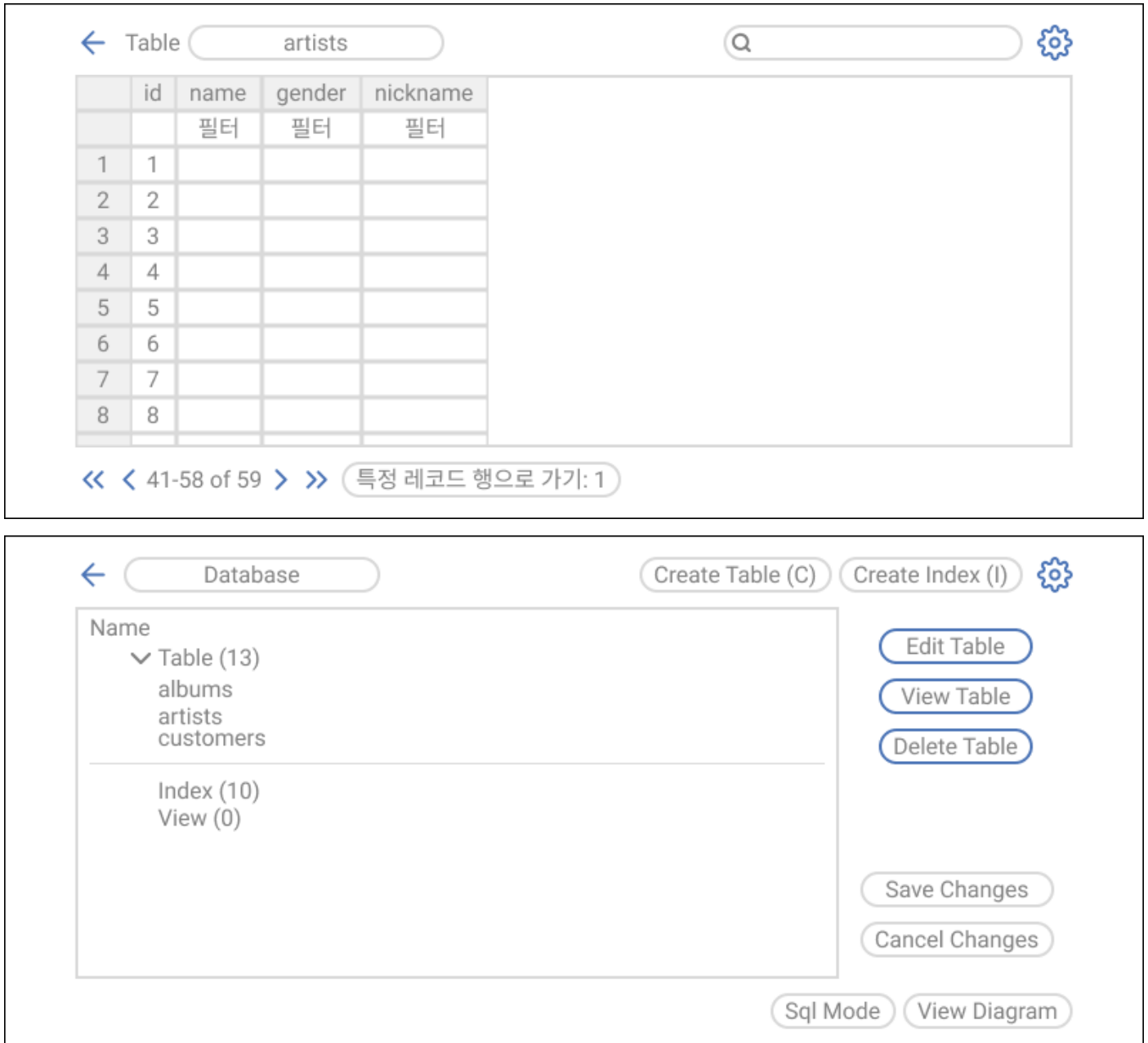
6. 실제로 구현된 기능

1. 디자인
2. 뷰어 에디터
3. 다이어그램
4. MCP 모듈

1. 디자인

디자인 구성은 프로젝트 초기부터 반복적인 수정과 외부 피드백을 통해 개선되었다. 특히 시각적 가독성과 터치 기반 조작의 직관성을 높이기 위해 여러 차례 테스트를 진행하였으며, 외부 인력에게서 테스트를 통한 디자인 검토와 보완 작업도 병행되었다.(디자인과 경영학과 등 여러 분야의 인력과 피드백 전달됨) 덕분에 전체적인 UI/UX는 개발 일정에 지장을 주지 않으면서도 적절한 시기에 안정된 형태로 구축될 수 있었다.

초기 구상





PengsQL 

test.db



user_base.db



다이어그램 생성

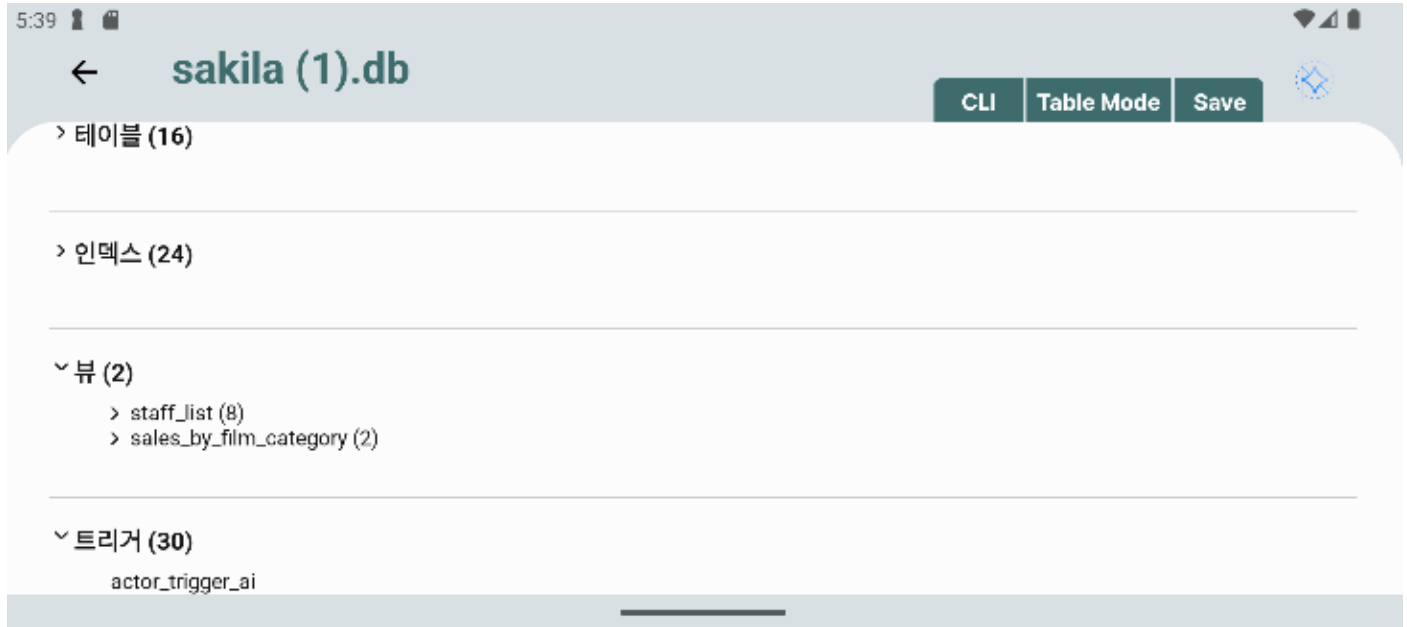
DB 생성



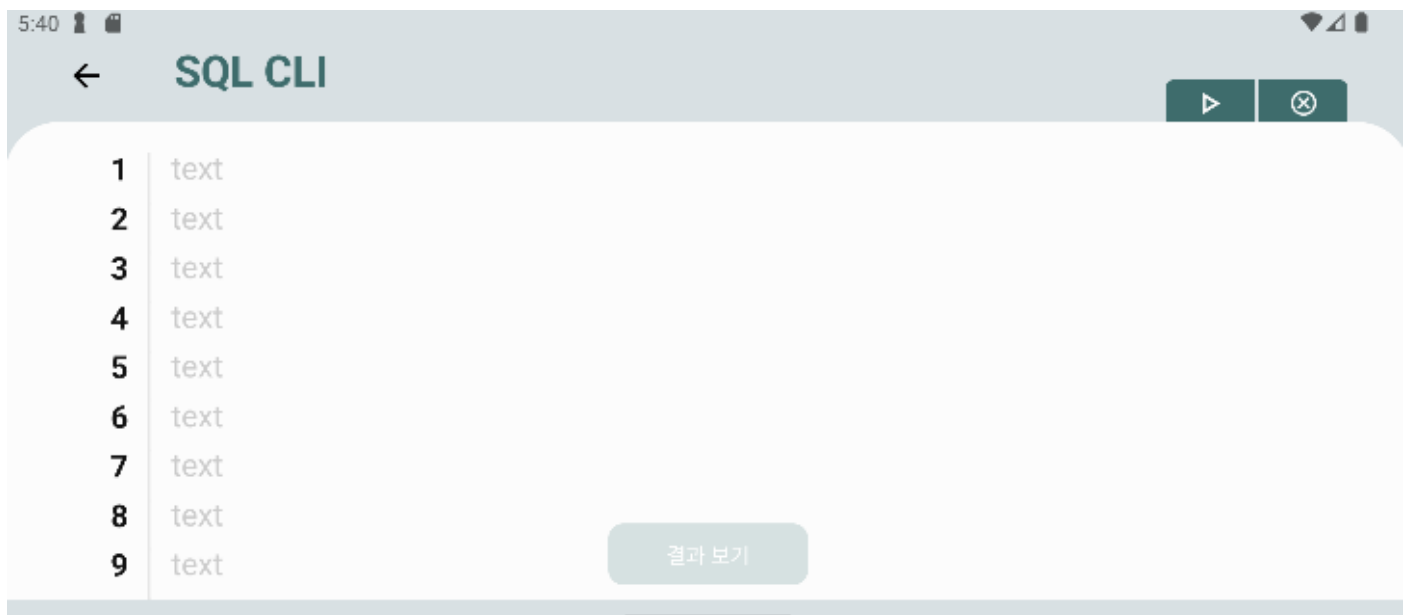
2. 뷰어 에디터

뷰어·에디터는 전체적으로 네 가지 주 기능으로 구성되었으며, 각각의 역할은 다음과 같이 구분된다.

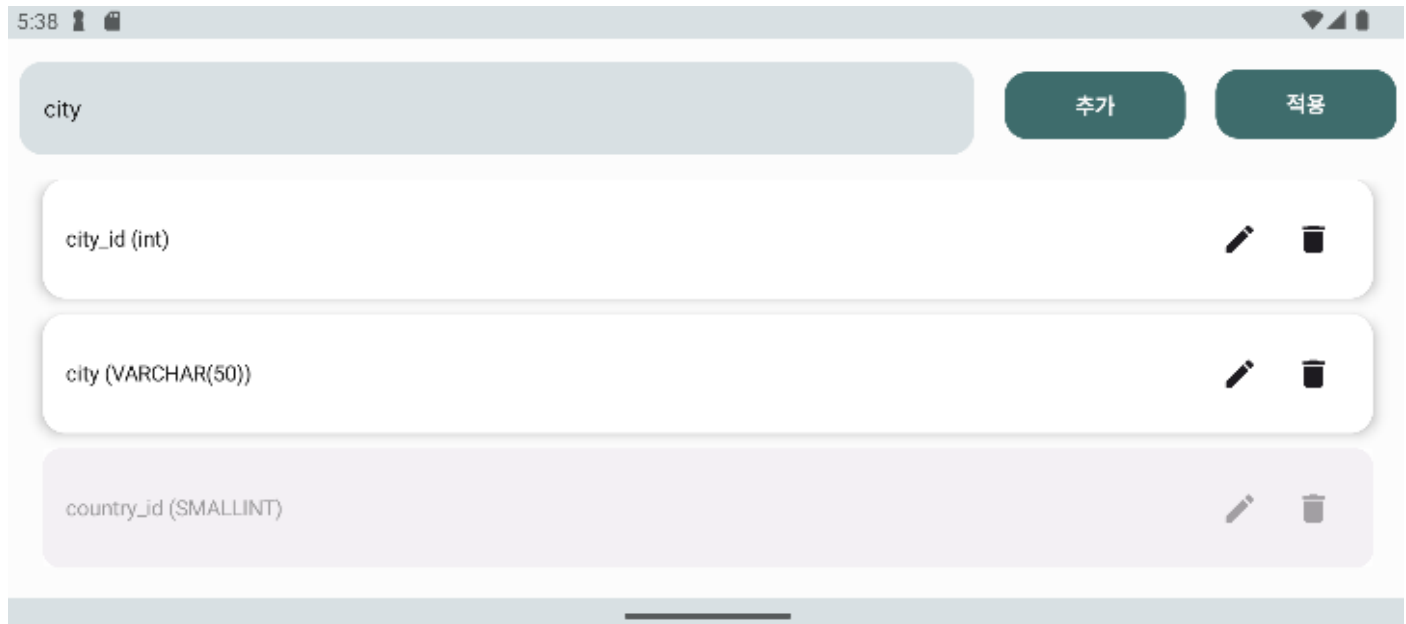
1. DB 메인 인터페이스: 현재 연결된 SQLite DB 내 자원을 탐색하고, 기본적인 구조 확인 및 내부 수정 작업이 가능한 공간이다.



2. SQL CLI: 사용자가 직접 SQL 쿼리를 입력하고 실행 결과를 확인할 수 있는 창으로, 쿼리 기반 작업을 선호하는 숙련된 사용자에게 최적화된 기능이다.



3. 테이블 수정 뷰: DB 내 테이블의 자료형과 구조를 시각적으로 수정할 수 있는 화면으로, GUI 기반 설계를 중시하는 사용자를 위한 기능이다.



4. 테이블 리스트 뷰: 테이블에 저장된 실제 레코드를 페이지 단위로 조회하고, 필터링 및 정렬이 가능한 리스트 형태로 제공된다.

5:41

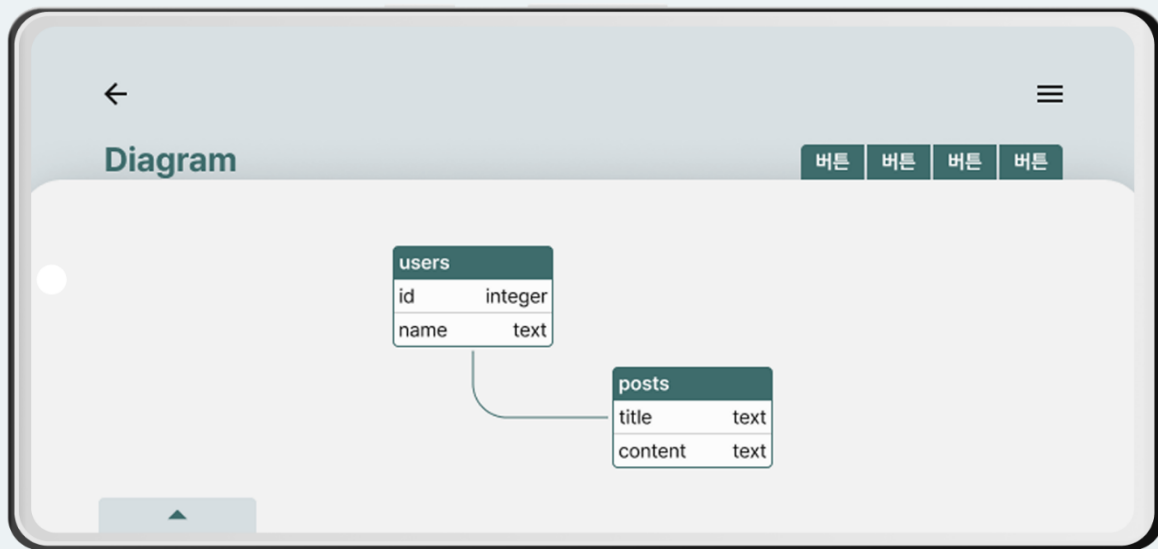
← country

	country_id	country	last_updat
	Filter	Filter	Filter
1	1	Afghanistan	2021-03-06 15:51:49
2	2	Algeria	2021-03-06 15:51:49
3	3	American Samoa	2021-03-06 15:51:49
4	4	Angola	2021-03-06 15:51:49
5	5	Anguilla	2021-03-06 15:51:49
6	6	Argentina	2021-03-06 15:51:49

이들 기능은 독립적이면서도 상호 연결되어 있어, 다양한 수준의 사용자가 목적에 맞게 작업을 수행할 수 있도록 구성되어 있다.

3. 다이어그램

DB 다이어그램 기능은 실시간 갱신이 가능한 시각화 모듈로 구현되었다. 사용자가 DB 구조를 변경할 경우, 그 변화가 즉시 다이어그램에 반영되도록 설계되어(2초 업데이트) 높은 반응성과 명확한 구조 인지가 가능하다. 특히 이 기능은 선언형 언어 기반의 다이어그램 구축과, 이를 해석하는 내부 파서(Parser) 엔진 구현을 통해 실현되었다. 결과적으로 PangSQL은 단순한 시각화가 아닌, 구조 설계와 추적을 병행할 수 있는 다이어그램 툴을 완성하게 되었다.



4. MCP 모듈

MCP(Module Context Protocol) 모듈은 생성형 AI와 PangSQL의 직접적인 연동을 목표로 구현되었으며, 기본적으로 서버 자원 및 네트워크 통신이 필요하다. 해당 모듈은 사용자가 연결한 DB 내에서 쿼리 구조를 분석하고, AI가 이를 바탕으로 쿼리를 자동 작성하거나 수정 제안을 제공할 수 있도록 설계되었다. 이를 위해 뷰어·에디터 내부 로직이 일부 확장되었으며, MCP는 에디터의 고급 기능 확장 모듈로서 작동한다.



7. 개발 방법론

1. 기존에 설정된 개발 방법론
2. 중간에 설정된 개발 방법론

1. 기존에 설정된 개발 방법론

자바-코틀린 언어 체인

프로젝트의 주요 기능 구현에는 Java, 프론트엔드 및 연동, 사용자 입력 관련 요소에는 Kotlin을 활용하는 언어 체인 구조가 적용되었다. 이는 각 개발자의 언어적 강점을 고려한 선택으로, Java는 기능 중심의 안정적인 백엔드 로직 구현에, Kotlin은 안드로이드 환경에서의 UI 구성에 적용하였다.

강력한 모듈화 및 책임 레이어 구조

개발 초기부터 모든 기능은 가능하면 독립된 모듈로 분리하는 원칙이 설정되었다. 기능 모듈은 기본적으로 싱글톤 구조를 취하며, 핵심 기능은 가장 아래 기반 레이어(L1)에, 그 위로는 **핵심 기능 레이어(L2)와 기능 모듈 레이어(L3), 그리고 종속 레이어(L4)가 존재한다. 모든 문제는 종속 레이어라는 구상과 함께 다수의 상위 모듈과 연결되며, 책임 문제 추적과 디버깅 효율을 극대화했다.

- L1: 시스템 기반 베이스 레이어
- L2: 핵심 기능 모듈
- L3: UI, 기능 등 중간 기능 모듈
- L4: 외부 연동, 표현 계층 등 종속성 중심 모듈

이러한 레이어 구조는 전체 프로젝트에 명확한 책임 구역을 부여하고, 코드 의존성을 관리하기 쉽게 하였다.

프로토타입 개발 전략

진행 중 디자인, 연동 모듈 등 외부 요소의 정체로 개발이 지연되지 않도록, 각 모듈 개발자는 일정 시점 이후 '일단 구현'을 목표로 프로토타입 개발에 착수하였다. 이 전략은 개발 정체 상황을 피하고, 가능한 한 많은 기능적 시도를 실험할 수 있는 환경을 제공하였다. 실제로 프로토타입 개발 단계에서는 정식 구조가 확정되기 전에도 적극적인 UI/UX 설계 실험과 기능 조합 검토가 이루어졌다.

2. 중간에 설정된 개발 방법론

중간 단계에서 진행된 프로젝트 점검을 통해, 기존에 설정된 방법론이 실제 개발에서 어떻게 작동했는지를 분석하였다. 그 결과, 다음 네 가지 측면에서 방법론의 실효성이 입증되었다.

1. 프로토타입 개발법의 활성화: 실제로 디자인과 기획이 늦어진 상황에서도 기능 구현이 병행되어 전체 프로젝트 속도에 긍정적인 영향을 주었다.
2. 레이어 기반 책임 분리 구조: 문제가 발생하더라도 어느 모듈의 문제인지 명확히 드러나므로 디버깅과 유지보수 효율성이 높았다.
3. 모듈화 기반 분산 개발: 각 개발자들이 독립적으로 기능을 개발하고 병합하는 구조가 유효하게 작동하였다.
4. 요청 기반 회의 소집 체계: 필요 시 주개발자 간의 소규모 회의가 즉시 소집될 수 있는 구조로, 신속한 피드백과 조율이 가능하였다.

이러한 결과에 따라, 다음 사항을 공식적인 개발 절차에 포함하기로 결정하였다.

- 현재의 회의 구조 유지: 기능 주개발자가 필요 시 인력 지원을 요청하고, 기능 단위의 완료 시점을 명확히 추적하는 구조를 계속 유지한다.
- 스크럼 기반 보고 체계 도입: '이전까지 한 것, 지금 하고 있는 것, 이후 진행할 것'의 항목을 중심으로, 문제 사항과 수정 사항을 기록하는 공용 노트 기반의 스크럼 보고서를 추가하였다. 이를 통해 개발 흐름을 전체 팀이 한눈에 파악하고, 협업 시 의사소통을 명확히 할 수 있게 되었다.

8. 일정과 출시

1. 일정

2. 출시 상태

1. 일정

이번 프로젝트는 초기 단계에서부터 프로토타입 개발 전략을 도입하여, 기획의 완성도를 기다리지 않고 핵심 기능 개발을 선제적으로 착수하였다. 이러한 접근은 전체 일정의 흐름을 견고히 유지하는 데 크게 기여하였으며, 중간 시험 기간에는 각 기능별 문서화를 통해 진행 상황을 구조화하고 명시화하여 일정 관리의 안정성을 확보할 수 있었다.

프로젝트의 전체적인 구현은 중간 이후 약 3주 차에 접어들면서 대부분의 기능이 완성 단계로 넘어갔고, 기말을 앞둔 시점에서는 기능 개발이 완료되어 팀원들이 시험 준비에 집중할 수 있도록 인력을 분산할 수 있었다. 결과적으로 프로젝트는 기능 완성도와 개발 일정 모두에서 안정적인 흐름을 유지하며 마무리 단계로 진입하였다.

스크럼 보고서 일정 보고서(사진)

분류	항목	2025/03				2025/04				2025/05				2025/06			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
메인 화면	디자인 (구현)																
	디자인 (설계)																
	새로운 데이터베이스																
	새로운 다이어그램																
	파일 업로드																
	파일 사용 (다이어그램 파일, db 파일)																
	설정 페이지 구현																
	파일 삭제																
다이어그램	디자인 (설계)																
	디자인 (구현)																
	컴파일러 구현																
에디터 메인	디자인 (구현)																
	디자인 (설계)																
	필드 출력 기능 (전체)																
	보기 기능 (테이블, 뷰, 인덱스, 트리거)																
	삭제 기능 (전체)																
	수정 기능 (인덱스, 트리거)																
	화면 업데이트																
	저장 기능																
	롤백 기능																
	디자인 (구현)																
	디자인 (설계)																
	필드 페이지 출력 기능 (테이블, 뷰)																

https://docs.google.com/spreadsheets/d/1s_hKZwVdrr0tTCd5SsgTeneQ-BlrSfDSiO25MkLbHx0/edit?gid=1134581641#gid=1134581641

2. 출시 상태

본 프로젝트는 초기부터 6월 내 프로덕션 라이브를 목표로 설정하고 있었으며, 기능 구현과 병행하여 실제 배포 가능성을 전제로 하는 작업 흐름을 유지해왔다. 그러나 구글 계정 인증 절차 및 첫 배포 과정에서 필요한 각종 절차 확인이 예상보다 지연되면서, 실제 출시 일정은 당초 목표보다 다소 유동적이 되었다.

이러한 상황을 고려하여, 개발 일정은 6월 내로 종료하기로 계획되어 있었고, 실제로 시험 주간 이전까지 기능 개발은 대부분 완료되었다. 다만 계정 인증 문제가 6월 17일에서야 해결되었으며, 현재는 내부 검토 상태에 들어간 상황이다. 큰 문제가 발견되지 않는다면, 출시 발표 이후 곧 진행될 것으로 예상된다.

최신 버전

여러 트랙에서 모든 버전의 개요를 확인하세요. [더보기](#)

필터 추가

버전 검색

최신 버전

버전	최신 버전	트랙	버전 상태	최근 업데이트 날짜	국가/지역	설치한 사용자 수	
평관-V1	1	내부 테스트	내부 테스터에게 제공된 전체 출시	2025년 6월 19일 오전 11:09	-	0.00%	→
1 (1.0)	1	비공개 테스트 - Alpha	검토 중 전체 출시	2025년 6월 19일 오전 11:17	1 /177	0.00%	→

19에 수정된 버전 재검토 중인 상태

9. 최종 평가

최종 평가

이번 프로젝트는 초기 설정된 개발 방법론과 프로토타입 개발 전략을 중심으로 진행되었으며, 결과적으로 프로젝트가 정해진 일정 내에서 기능 완성도에 도달하는 데 성공할 수 있었다. 초반부터 방법론이 명확히 설정되어 있었기에 프로젝트의 중심을 잃지 않고, 각 기능을 목표 시점 내에 구현할 수 있었던 것은 큰 성과였다.

물론, 도입한 방법론이 완벽했던 것은 아니다. 초기 설계 단계에서 상정했던 계층(Layer) 구조는 실제 개발이 진행되며, 초기 구상에서 바라본다면 사실상 무너졌고, 레이어 간 구분은 흐릿해졌다. 그럼에도 불구하고 기능 단위를 모듈화하여 책임을 최대한 분리하려는 구조적 시도는 유지되었고, 일정 수준의 구조 정리는 지켜졌다. 물론 여전히 일부 예외 처리나 구조는 개발자의 기억과 주석에 의존해야 하는 한계도 분명 존재한다.

이 프로젝트는 여러 개발 방식과 협업 방식을 실험한 첫 번째 종합 프로젝트이기도 하다. 외부 피드백 수용, UML 기반의 설계 문서 작성, 출시 가이드라인의 준수 등, 다양한 자원과 인력을 동원하고 이를 실제로 프로젝트에 통합한 경험은 매우 색달랐다. 단순 기능 구현을 넘어서, 프로젝트 전체의 방향성을 일관되게 유지하며 현대적인 요구와 기술 흐름을 반영하는 데에도 성공했다고 평가할 수 있다.

출시 준비 또한 본 프로젝트에서 처음 시도된 과정이었다. 구글 개발자 인증 절차와 같은 예상치 못한 외부 변수로 인해 일정이 지연되었고, 출시 직전 앱 패키지 이름(`com.example` 등)에 대한 제약으로 인해 사실상 전체 프로젝트 리팩토링이 요구되는 문제도 발생하였다. 특히 Kotlin에 대한 자동 리팩토링 도구의 인식 부족으로 인해, 대부분의 변경을 수작업으로 처리해야 했다. 하지만 이러한 경험을 통해 향후 앱 개발 시에는 보다 철저한 사전 준비와 도구 선택의 중요성을 학습할 수 있었다.

물론 프로젝트는 여전히 보완 가능한 부분이 존재하며, 향후 발전시킬 여지도 충분하다. 그럼에도 불구하고 정해진 시간 내에서 구현된 본 프로젝트는 기능 구현, 방법론 실험, 출시 준비라는 모든 면에서 성공적인 결과를 도출했다고 평가할 수 있다. 이는 단지 기능이나 프로젝트의 완성이 아니라, 하나의 소프트웨어를 실제로 기획하고, 구현하는 DevOps를 체험했다고 평가하겠다.