

FLUTTER APPLICATION

PARA

Management System

생각을 정리하고, 행동을 이끌어내는
두 번째 뇌 (Second Brain)

Projects

Areas

Resources

Archive

문서 버전: v1.0
작성일: 2026년 2월 18일

상태: 계획 단계

목차

1 프로젝트 개요 (Project Overview)	4
1.1 PARA란?	4
1.2 앱 비전	4
1.3 타겟 플랫폼	4
2 핵심 기능 (Core Features)	5
2.1 MVP (Minimum Viable Product) — Phase 1	5
2.1.1 PARA 카테고리 관리	5
2.1.2 Project (프로젝트) 기능	5
2.1.3 Area (영역) 기능	5
2.1.4 Resource (자료) 기능	5
2.1.5 Archive (보관함) 기능	5
2.1.6 검색 & 필터	5
2.2 Enhanced Features — Phase 2	6
2.3 Advanced Features — Phase 3	6
3 기술 스택 (Tech Stack)	7
3.1 Frontend	7
3.2 데이터 저장	7
3.3 주요 패키지	7
4 아키텍처 (Architecture)	9
4.1 프로젝트 구조	9
4.2 아키텍처 레이어	9
5 데이터 모델 (Data Models)	10
5.1 ERD (Entity Relationship)	10
5.2 모델 클래스 예시	10
6 UI/UX 설계 (Design Concept)	12
6.1 디자인 컨셉	12
6.2 색상 팔레트	12
6.3 전체 레이아웃 (Desktop)	12
6.4 인터랙션 & 애니메이션	12
6.5 네비게이션 패턴	13
7 개발 로드맵 (Development Roadmap)	14
7.1 Phase 1: Foundation 1~2주	14
7.1.1 Week 1 — 프로젝트 셋업 & 기본 구조	14
7.1.2 Week 2 — 핵심 CRUD	14
7.2 Phase 2: Enhancement 3~4주	14
7.2.1 Week 3 — UI 고도화	14
7.2.2 Week 4 — 연결 & 통계	14
7.3 Phase 3: Polish 5~6주	14
7.3.1 Week 5 — 데스크탑 최적화	14
7.3.2 Week 6 — 완성도	14

8 고려사항 & 아이디어	15
8.1 차별화 포인트	15
8.2 주의사항	15
8.3 향후 확장 가능성	15
9 개발 시작 전 체크리스트	15

프로젝트 개요 (Project Overview)

PARA란?

Tiago Forte가 고안한 **디지털 정보 관리 시스템**으로, 모든 정보를 4가지 카테고리로 분류합니다.

카테고리	설명	특성
Projects	명확한 목표와 기한이 있는 단기 노력	시작일/마감일, 진행률, 할 일 목록
Areas	지속적으로 유지해야 하는 책임 영역	건강, 재정, 커리어 등 지속적 관리
Resources	관심 있는 주제나 참고 자료	태그, 북마크, 메모, 링크
Archive	비활성화된 항목들의 저장소	완료된 프로젝트, 더 이상 관리 안 하는 영역

앱 비전

VISION

"생각을 정리하고, 행동을 이끌어내는 두 번째 뇌 (Second Brain)"

핵심 가치:

- 심플하지만 강력한 — 복잡하지 않으면서 필요한 기능은 모두 제공
- 빠른 입력 — 떠오르는 생각을 즉시 캡처
- 유연한 이동 — 항목 간 카테고리 전환이 자유로움
- 아름다운 UI — 다크/라이트 테마, 모던한 디자인

타겟 플랫폼

우선순위	플랫폼
1차	Windows Desktop (Flutter Desktop)
2차 확장	Android / iOS (Flutter Mobile)
3차 확장	Web (Flutter Web)

핵심 기능 (Core Features)

MVP (Minimum Viable Product) — Phase 1

PARA 카테고리 관리

- 4개 카테고리(P/A/R/A) 대시보드
- 각 카테고리 내 항목 CRUD (생성/읽기/수정/삭제)
- 항목 간 카테고리 이동 (드래그 앤 드롭 또는 메뉴)
- 아카이브 자동 제안 (오래된 항목)

Project (프로젝트) 기능

- 프로젝트 생성 (제목, 설명, 시작일, 마감일)
- 하위 태스크 목록 (체크리스트)
- 진행률 자동 계산 (완료된 태스크 / 전체 태스크)
- 상태 관리: 진행중 대기중 완료 보관됨
- 마감일 임박 알림

Area (영역) 기능

- 영역 생성 (제목, 설명, 아이콘)
- 관련 프로젝트 연결
- 영역별 메모/노트 작성
- 유지 기준(Standard) 설정 (예: "매주 운동 3회")

Resource (자료) 기능

- 자료 생성 (제목, 내용, 태그)
- 태그 기반 분류
- 마크다운 메모 지원
- URL 링크 저장 및 프리뷰

Archive (보관함) 기능

- 보관된 항목 목록
- 보관 날짜 기록
- 복원 기능 (보관함 → 원래 카테고리)
- 영구 삭제 (확인 후)

검색 & 필터

- 전역 검색 (모든 카테고리에서 검색)
- 태그 필터링
- 날짜 범위 필터
- 상태별 필터

Enhanced Features — Phase 2

대시보드 & 통계

- 오늘의 요약
- 주간/월간 활동 히트맵
- 카테고리별 항목 수 차트
- 최근 활동 타임라인

항목 간 연결 (Linking)

- 프로젝트 ↔ 영역 연결
- 프로젝트 ↔ 자료 연결
- 양방향 링크 (Backlinks)
- 연결 그래프 시각화

Quick Capture (빠른 입력)

- 시스템 트레이 아이콘
- 글로벌 단축키로 빠른 메모
- Inbox (분류 전 임시 저장소)
- 나중에 적절한 카테고리로 분류

커스터마이징

- 다크/라이트 테마
- 커스텀 컬러 팔레트
- 아이콘 커스터마이징
- 레이아웃 (그리드/리스트/카드)

Advanced Features — Phase 3

- 동기화 & 백업 — 로컬 JSON/SQLite 백업, Google Drive / OneDrive 연동, 내보내기/가져오기
- 첨부파일 지원 — 이미지·파일 첨부, 스크린샷 캡처 후 바로 첨부
- 스마트 기능 — 주간 리뷰(Weekly Review) 알림, 아카이브 제안, 태그 추천, 검색어 자동완성

기술 스택 (Tech Stack)

Frontend

영역	기술	비고
Framework	Flutter 3.x (Stable)	
State Mgmt	Riverpod 2.x	추천
Routing	GoRouter	
UI	Material 3 (Material You)	
Animation	Flutter Animate / Lottie	
Icons	Material Icons + Lucide	
Fonts	Pretendard (한글), Inter (영문)	Google Fonts

데이터 저장

로컬 저장소 (Phase 1 — 오프라인 우선)

- **SQLite** (drift 패키지) — 구조화된 데이터
- **SharedPreferences** — 앱 설정
- **파일 시스템** — 첨부파일 저장

클라우드 동기화 (Phase 3 — 선택적)

- Supabase 또는 Firebase
- Google Drive API

주요 패키지

dependencies:

```
# State Management
flutter_riverpod: ^2.0.0
riverpod_annotation: ^2.0.0

# Database
drift: ^2.0.0          # SQLite ORM
sqlite3_flutter_libs:   # SQLite 라이브러리

# Routing
go_router: ^14.0.0

# UI / UX
flutter_animate: ^4.0.0 # 애니메이션
google_fonts: ^6.0.0    # 폰트
fl_chart: ^0.70.0       # 차트
flutter_markdown: ^0.7.0 # 마크다운 렌더링
```

```
# Utilities
uuid: ^4.0.0          # 고유 ID 생성
intl: ^0.19.0         # 날짜/시간 포맷
url_launcher: ^6.0.0  # URL 열기
path_provider: ^2.0.0 # 파일 경로

# Windows 전용
window_manager: ^0.4.0 # 윈도우 크기 관리
system_tray: ^2.0.0   # 시스템 트레이
hotkey_manager: ^0.2.0 # 글로벌 단축키
```

아키텍처 (Architecture)

프로젝트 구조

```

lib/
├── main.dart                      # 앱 진입점
├── app.dart                        # MaterialApp 설정
└── core/
    ├── constants/                 # 색상, 크기, 문자열
    ├── theme/                      # 다크/라이트 테마
    ├── router/                     # GoRouter 설정
    ├── utils/                      # 유ти리티 함수
    └── extensions/                # 확장 메서드
├── data/
    ├── database/                  # Drift DB, Tables, DAOs
    ├── models/                     # 데이터 모델
    └── repositories/              # 리포지토리 패턴
├── providers/                     # Riverpod 프로바이더
└── features/
    ├── dashboard/                # 대시보드 (홈)
    ├── projects/                  # 프로젝트 관리
    ├── areas/                     # 영역 관리
    ├── resources/                 # 자료 관리
    ├── archive/                   # 보관함
    ├── search/                    # 전역 검색
    ├── inbox/                     # 빠른 입력
    └── settings/                  # 앱 설정
└── shared/                        # 공유 위젯 & 레이아웃

```

아키텍처 레이어

UI Layer	Dashboard Screen, Projects Screen, Areas Screen, Resources Screen, ...
Provider Layer	State Notifiers / AsyncNotifiers / Providers (Riverpod)
Repository Layer	Repositories — 데이터 소스 추상화, 비즈니스 로직 처리
Data Layer	Drift DB (SQLite) · SharedPreferences (설정) · File System (첨부파일)

표 1: 4-Layer Architecture

데이터 모델 (Data Models)

ERD (Entity Relationship)

Projects	Areas	Resources
id (PK) title description status area_id (FK) start_date due_date progress created_at updated_at archived_at	id (PK) title description icon standard created_at updated_at archived_at	id (PK) title content (MD) url item_type created_at updated_at archived_at

표 2: 주요 엔티티

Tasks
id (PK)
project_id (FK)
title
is_completed
order_index
created_at

표 3: Tasks

Tags / Item_Tags
id (PK) name color created_at
item_id · item_type · tag_id

표 4: Tags & Junction

Links
id (PK)
from_id
from_type
to_id
to_type
created_at

표 5: Links

모델 클래스 예시

```
class Project {
  final String id;
  final String title;
  final String? description;
  final ProjectStatus status;
  final String? areaId;
  final DateTime? startDate;
  final DateTime? dueDate;
  final double progress;      // 0.0 ~ 1.0
  final DateTime createdAt;
  final DateTime updatedAt;
  final DateTime? archivedAt;
  final List<Task> tasks;
  final List<Tag> tags;
```

```
}
```

```
enum ProjectStatus {  
    active,      // 진행중  
    onHold,      // 대기중  
    completed,   // 완료  
    archived,    // 보관됨  
}
```

UI/UX 설계 (Design Concept)

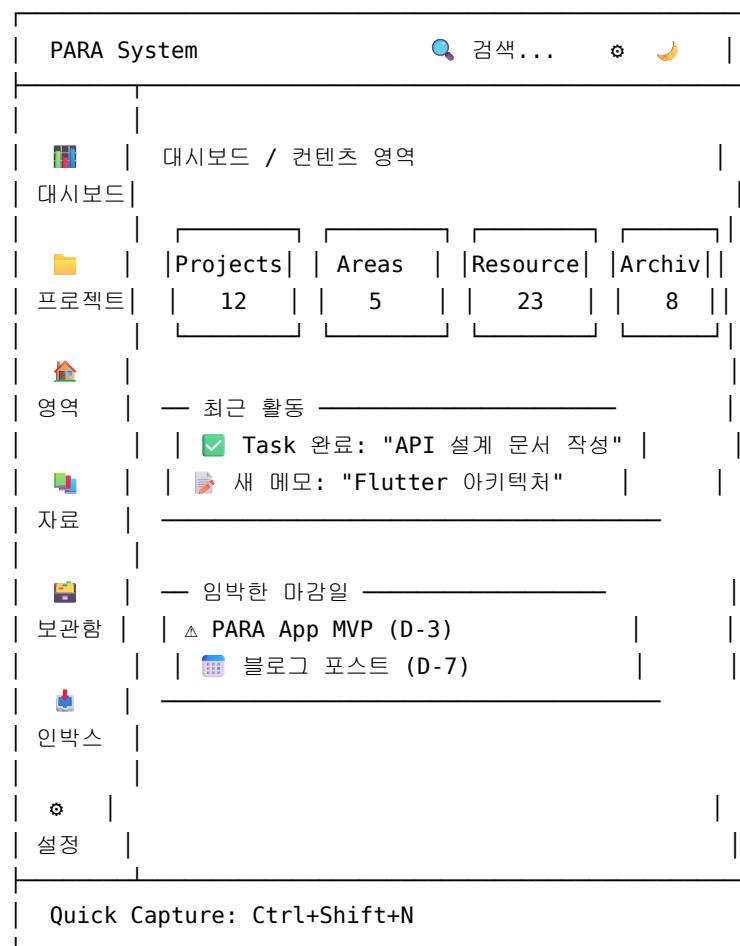
디자인 컨셉

- 스타일: Glassmorphism + Material 3
- 다크 모드 기본 (라이트 모드 전환 가능)

색상 팔레트



전체 레이아웃 (Desktop)



인터랙션 & 애니메이션

- 카드 호버 시 약간의 lift 효과 (elevation 변화)
- 카테고리 전환 시 슬라이드 트랜지션
- 체크박스 완료 시 confetti/pulse 애니메이션
- 드래그 앤 드롭으로 카테고리 간 항목 이동
- 검색 시 실시간 하이라이트

네비게이션 패턴

플랫폼	패턴
Desktop	NavigationRail (좌측 사이드바)
Mobile	BottomNavigationBar (하단 탭)
반응형	화면 크기에 따라 자동 전환

개발 로드맵 (Development Roadmap)

Phase 1: Foundation 1~2주

Week 1 — 프로젝트 세팅 & 기본 구조

- Flutter 프로젝트 생성
- 패키지 설치 및 설정
- 폴더 구조 세팅
- 테마 시스템 구축 (다크/라이트)
- 라우팅 설정 (GoRouter)
- 메인 레이아웃 (NavigationRail + Content)
- SQLite 데이터베이스 스키마 정의

Week 2 — 핵심 CRUD

- Project CRUD (생성/읽기/수정/삭제)
- Area CRUD
- Resource CRUD
- Archive 기능 (이동/복원)
- Task 체크리스트 (프로젝트 하위)
- 기본 검색 기능

Phase 2: Enhancement 3~4주

Week 3 — UI 고도화

- 대시보드 화면 구현
- 카드 뷰 / 리스트 뷰 전환
- 애니메이션 추가
- 태그 시스템 구현
- 마크다운 에디터 (Resources)
- 진행률 시각화

Week 4 — 연결 & 통계

- 항목 간 링크 기능
- 통계 차트 (fl_chart)
- 활동 타임라인
- 마감일 알림
- Inbox / Quick Capture

Phase 3: Polish 5~6주

Week 5 — 데스크탑 최적화

- 시스템 트레이 아이콘
- 글로벌 단축키
- 윈도우 크기 관리
- 키보드 단축키 전체 구현
- 성능 최적화

Week 6 — 완성도

- 데이터 내보내기/가져오기
- 백업/복원 기능
- 온보딩 화면
- 에러 핸들링 강화
- 테스트 작성

고려사항 & 아이디어

차별화 포인트

#	기능	설명
1	Weekly Review 도우미	매주 일요일 알림 → 각 프로젝트 리뷰 프롬프트
2	시간 추적 연동	프로젝트별 소요 시간 타이머
3	템플릿 시스템	자주 만드는 프로젝트 유형을 템플릿으로 저장
4	Zettelkasten 하이브리드	Resource에 양방향 링크 → 지식 그래프
5	AI 분류 제안	제목/내용 기반으로 카테고리 자동 추천 (로컬 ML)
6	뽀모도로 연동	프로젝트/태스크에 뽀모도로 타이머 내장

주의사항

주의사항

- **오프라인 우선** — 인터넷 없이도 완벽하게 동작해야 함
- **데이터 안전** — 로컬 데이터 손실 방지를 위한 자동 백업
- **성능** — 항목이 수백~수천 개여도 빠르게 동작
- **マイ그레이션** — DB 스키마 변경 시 기존 데이터 보존
- **접근성** — 키보드만으로 모든 기능 사용 가능

향후 확장 가능성

- **플러그인 시스템** — 사용자가 기능 확장 가능
- **Notion/Obsidian 연동** — 데이터 가져오기/내보내기
- **GitHub Issues 연동** — 개발 프로젝트 동기화
- **캘린더 뷰** — 프로젝트/마감일을 캘린더로 시각화
- **모바일 위젯** — Android/iOS 홈 화면 위젯

개발 시작 전 체크리스트

- Flutter SDK 설치 확인 (최신 Stable)
- Windows Desktop 개발 환경 설정
- 디자인 시안 확정 (Figma 등)
- Git 저장소 초기화
- CI/CD 파이프라인 설정 (선택)
- 코딩 컨벤션 문서화

문서 작성일: 2026-02-18 · 버전: v1.0 · 상태: 계획 단계