

PomoTodo



김도연
도민준
신승재
심근웅

TEAM IEEE



ISTJ
현실주의자

심근웅
기록자



ENTP
변론가

도민준
지도자



ENFP
활동가

신승재
진행자



ENTJ
통솔자

김도연
설계사

기획 배경

프로젝트 목표 및 주제 설명



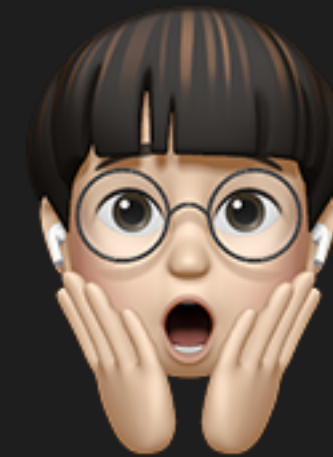
데이터 설계 및 핸들링을 해보고 싶습니다!

저는 커스텀 UI를 제작하고 다양한 UX 인터렉션을 처리해보고 싶어요!



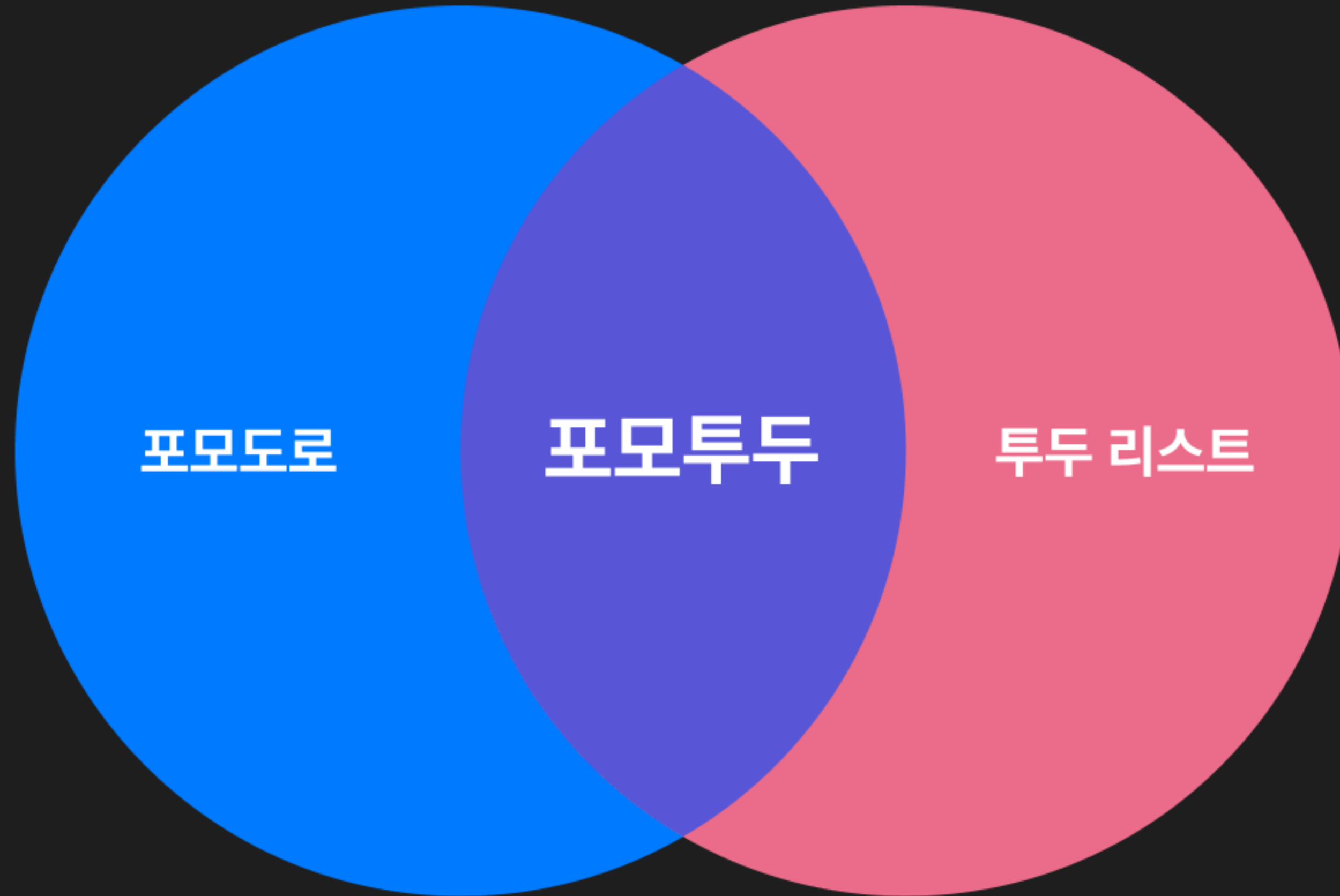
차트 UI를 구현해보고 싶어요!

팀 프로젝트의 전반적인 흐름을 경험해보고 싶습니다!



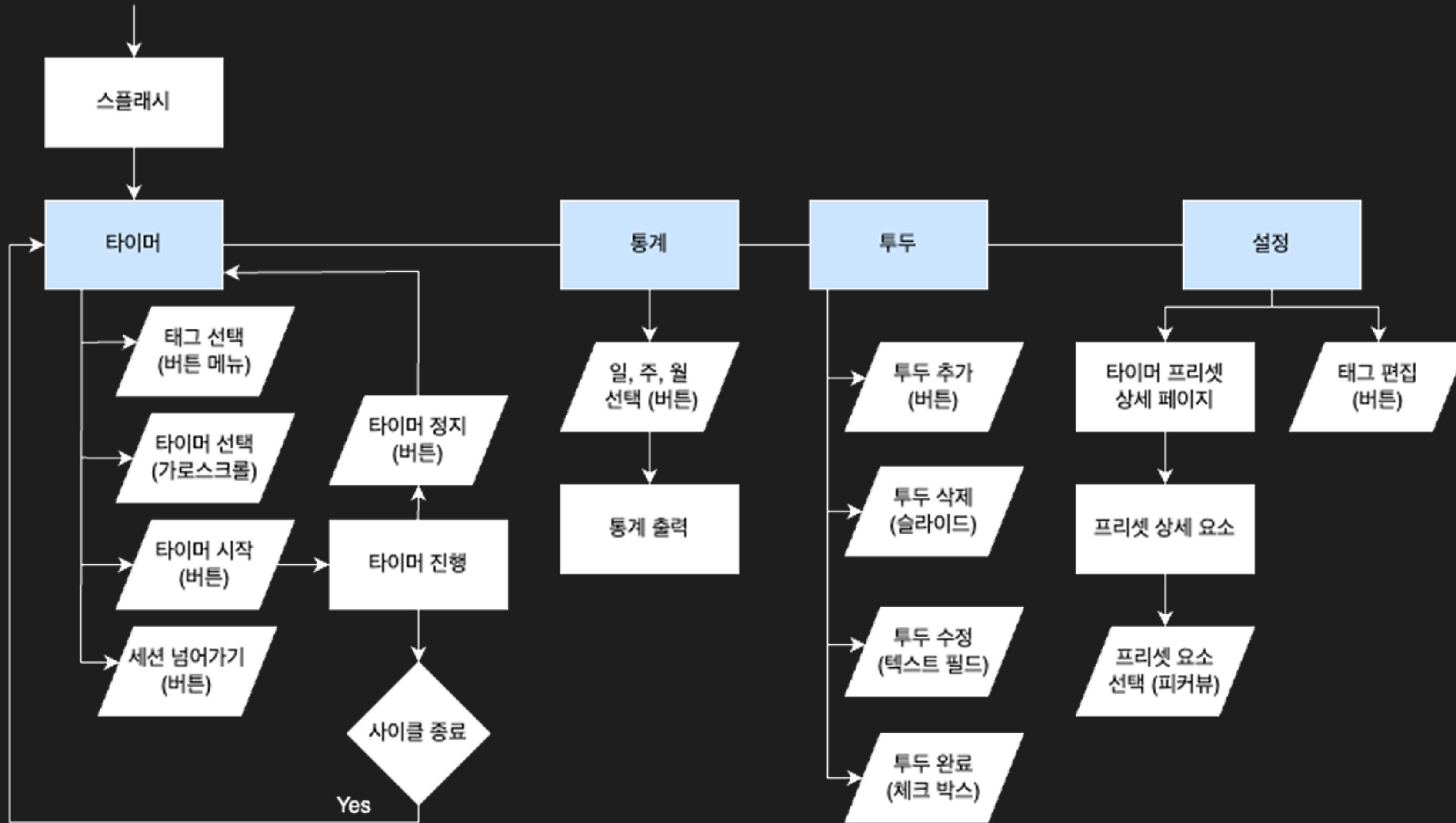
프로젝트 주제

프로젝트 목표 및 주제 설명



User Flow Chart

기획 과정



기능 정의서

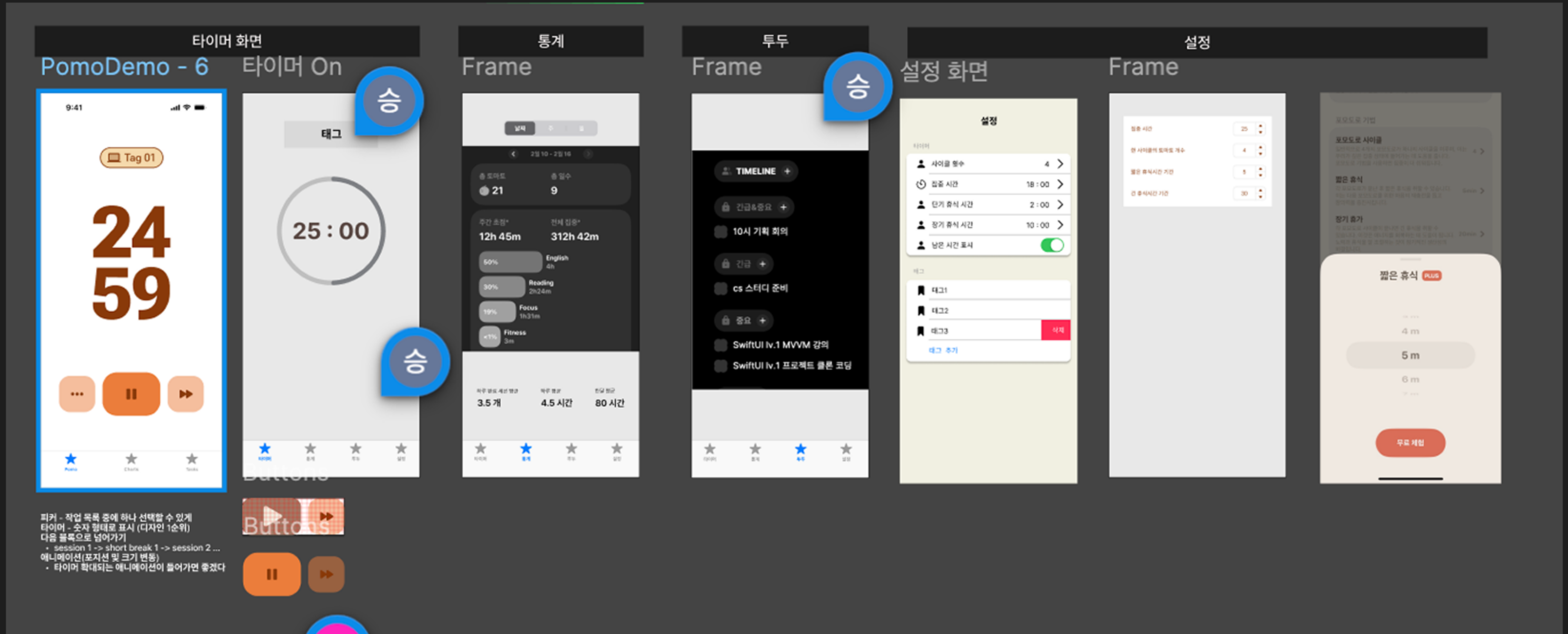
기획 과정

Aa 구분	☰ 주 기능	☰ 상세 기능	☰ 설명	☰ 개발 담당
1. 스플래시	1. 앱 실행		앱 로고 띄우기. 실행 로딩 화면 역할	송재
2. 탭바	2.1. 메뉴 접근	현재 보이는 화면 탭 아이템 활성화	활성화 탭 아이템과 비활성화 탭 아이템 로고/색상/사이즈 구분	도연 송재
3. 뽀모도로 타이머	3.1. 뽀모도로 타이머 비활성화(준비)	타이머 페이지 진입 시, 가장 기본 화면 - 태그 변경 버튼 - 타이머 세트 변경 - 타이머 재생 버튼 - 다음 세션으로 넘어가기 버튼 - 현재 사이클 진행 상황 표시	- 태그마다 색깔 지정되어 있음. 변경 버튼 클릭 시, 메뉴 드롭다운 스타일로 선택지 표시 - 현재 사이클 진행 상황과 집중/휴식 세션 구분 아이콘 동기화 - 타이머 재생 버튼 클릭시, 부드럽게 변환되는 애니메이션 진행 및 하단 탭바 메뉴 숨기기	도연 송재
	3.2. 뽀모도로 타이머 활성화	- 남은 시간 줄어듦에 따라 원형 진행바 및 디지털 시계 시간 감소 애니메이션 진행 - 타이머 정지 버튼 - 현재 사이클 진행 상황(휴식만) 표시	- 각 요소마다 지정된 컬러 존재(피그마에서 확인) - 타이머 정지 버튼 → 3.1 화면으로 돌아감(토마토 초기화 됨). 시간만 저장 - 타이머 종료(모든 시간이 지남) → 3.1 화면으로 돌아감(다음 세션 타이머로 변화)	도연 송재
4. 통계	4.1. 일간 통계	통계 페이지 진입 시, 가장 기본 화면 - 세그먼트 컨트롤 : 일 항목 활성화 - 날짜 이동 버튼 활성화/비활성화 : 데이터 없으면 이전/다음 버튼 비활성화	- 해당 일의 포모도로 및 세션 : 선택된 날짜에서 행한 포모도로 수 및 세션 - 누적 집중시간 : 어플 사용하면서 전체 누적 집중 시간 (일 / 주 / 월에 관계 없이 고정값) - 일간 집중 시간은 그 날 동안 태그별 집중시간의 총 합 - 차트 1. 태그 별 정렬 고정 2. 각 집중 시간의 퍼센트와, 집중 시간 값 나타냄 - 평균 세션, 평균 집중 시간 : 전체 시간 및 세션 / 사용한 날짜 수	민준 송재

	4.3. 월간 통계	- 세그먼트 컨트롤 : 월 항목 활성화
5. 투두리스트	5.1. 투두리스트 항목 조회	투두 페이지 진입 시, 가장 기본 화면
	5.2. 투두리스트 항목 생성	태그 옆 + 버튼 : 새로운 항목 추가할 수 있는 row 추가
	5.3. 투두리스트 항목 변경	투두리스트 항목 터치 : 항목 내용 수정
	5.4. 투두리스트 항목 삭제	투두리스트 좌측으로 슬라이드
6. 설정	6.1. 뽀모도로 설정 기본 페이지	설정 페이지 진입 시, 가장 기본 화면 - 뽀모도로 항목 조회 버튼 - 태그 편집 기능 활성화 - 투두리스트 태그 항목 선택
	6.2. 뽀모도로 타이머 항목 조회	뽀모도로 항목 터치 : 항목 내용 수정
	6.3. 뽀모도로 타이머 항목 편집	항목 값 피커

와이어 프레임

기획 과정



프로토타입

기획 과정

컴포넌트

Frame 1707481909

Button

Tab Item
Tab Bar - iPhone

Indigo

Light	Hex: #4A5568, RGB: 74, 85, 104
Light hover	Hex: #5A6677, RGB: 90, 102, 119
Light active	Hex: #6A7788, RGB: 106, 119, 136
Normal	Hex: #3A4558, RGB: 58, 69, 88
Normal hover	Hex: #4A5568, RGB: 74, 85, 104
Normal active	Hex: #5A6677, RGB: 90, 102, 119
Dark	Hex: #2A3548, RGB: 42, 53, 72
Dark hover	Hex: #3A4558, RGB: 58, 69, 88
Dark active	Hex: #4A5568, RGB: 74, 85, 104
Darker	Hex: #1A2538, RGB: 26, 37, 56

Blue

Light	Hex: #40A0D0, RGB: 64, 160, 208
Light hover	Hex: #50B0E0, RGB: 80, 176, 224
Light active	Hex: #60C0F0, RGB: 96, 192, 240
Normal	Hex: #3080C0, RGB: 48, 128, 192
Normal hover	Hex: #40A0D0, RGB: 64, 160, 208
Normal active	Hex: #50B0E0, RGB: 80, 176, 224
Dark	Hex: #2060A0, RGB: 32, 96, 160
Dark hover	Hex: #3080C0, RGB: 48, 128, 192
Dark active	Hex: #40A0D0, RGB: 64, 160, 208
Darker	Hex: #104060, RGB: 16, 64, 96

Cyan

Light	Hex: #40D0D0, RGB: 64, 208, 208
Light hover	Hex: #50E0E0, RGB: 80, 224, 224
Light active	Hex: #60F0F0, RGB: 96, 240, 240
Normal	Hex: #30A0A0, RGB: 48, 160, 160
Normal hover	Hex: #40C0C0, RGB: 64, 192, 192
Normal active	Hex: #50E0E0, RGB: 80, 224, 224
Dark	Hex: #208080, RGB: 32, 128, 128
Dark hover	Hex: #30A0A0, RGB: 48, 160, 160
Dark active	Hex: #40C0C0, RGB: 64, 192, 192
Darker	Hex: #106060, RGB: 16, 96, 96

Teal

Light	Hex: #40D0B0, RGB: 64, 208, 176
Light hover	Hex: #50E0C0, RGB: 80, 224, 192
Light active	Hex: #60F0D0, RGB: 96, 240, 208
Normal	Hex: #30A090, RGB: 48, 160, 144
Normal hover	Hex: #40C0A0, RGB: 64, 192, 160
Normal active	Hex: #50E0B0, RGB: 80, 224, 176
Dark	Hex: #208080, RGB: 32, 128, 128
Dark hover	Hex: #30A090, RGB: 48, 160, 144
Dark active	Hex: #40C0A0, RGB: 64, 192, 160
Darker	Hex: #106060, RGB: 16, 96, 96

Typography system

English	Korean
Semi Bold 36pt / 자간 48	세미 볼드 36pt -2.5%/ 행간 150%
Semi Bold 24pt / 자간 34	세미 볼드 30pt -2.5%/ 행간 150%
Medium 18pt / 자간 26	세미 볼드 24pt -2.5%/ 행간 150%
Regular 14pt / 자간 22	미디움 20pt -2.5%/ 행간 150%
Round 14	미디움 18pt -2.5%/ 행간 150%
	레귤러 14pt -2.5%/ 행간 150%

Design System

타이머 화면

타이머 기본 - 집중... 타이머 기본 - 휴식... 타이머 기본 - 태그... 타이머 활성화 타이머 활성화

통계 **투두** **설정**

통계 페이지F 투두F 설정 F 뽀모도로 설정 F 프리셋 세부 설정 F

디자인 시스템

Hi-Fi 프로토타입

역할 분담

개발 준비 과정

근웅



- 아이디어 정리
- 와이어프레임 제작
- 디자인 시스템 수립
- UI 디자인(하이파이) 제작
- 로고 제작

도연



민준



- Github 셋팅(.gitignore, .gitattribute)
- Issue, PR, Commit 컨벤션 정리
- Github Project 셋팅
- Github Action 브랜치 자동생성
- XCode 프로젝트 임포트

승재



프로젝트 관리

개발 과정



Figma

- 와이어프레임
- 디자인 시스템
- 최종 디자인
- 아이디어 공유



Github

- 원격 레포지토리 관리
- Github Project
- Actions을 통한 브랜치 자동 생성
- 코드 리뷰



Notion

- 전체 일정 관리
- 회의록 작성
- 개인 아이디어 공유



Discord

- 음성 회의
- Github 알림 연동
- 레퍼런스 공유
- 데일리 스크럼
- 질문 및 이슈, 버그 제보

컨벤션 & 코드 리뷰

개발 과정

The screenshot shows a GitHub Issues page with the search filter 'is:issue state:closed'. The page lists 12 issues, each with a title, author, and status. The issues are:

- [bugfix] Todo 버그 수정 (bug) - #52 - by Monfi98
- [bugfix] 통계vm 버그 수정 (bug) - #43 - by aado97
- [bugfix] 타이머 버그 수정 (bug) - #42 - by doyeonk429
- [feature] 화면 세로 모드, 라이트 모드 고정 및 스플래시 화면, 아이콘 추가 (feature) - #41 - by Monfi98
- [bugfix] 설정, 투두 테스트 이슈 수정 (bug) - #40 - by GeunwoongSim
- [refactor] StatisticsView 및 VM 리팩터링 (refactor) - #38 - by aado97
- [feature] TimerUseCase 함수 분리 (feature) - #33 - by Monfi98
- [refactor] TodoViewModel 리팩토링 (refactor) - #32 - by Monfi98
- [feature] 설정 데이터 연결 (feature) - #31 - by GeunwoongSim
- [feature] Timer Usecase 적용 (feature) - #30 - by doyeonk429
- [feature] 통계View 데이터 연결 (feature) - #29 - by aado97
- [feature] 설정 개발 (feature) - #24 - by GeunwoongSim

Issue

The screenshot shows a GitHub Pull Request page with the title '추가 설명'. The commit message is:

1. DIContainer 리팩토링

DIContainer에서 UseCase를 주입한 ViewModel을 생성하도록 리팩토링 하였습니다.
DIContainer의 정의는 다음과 같습니다

DI컨테이너(Dependency Injection Container)는 의존성 주입을 제공하는 컨테이너로, 객체의 생성과 제어를 담당한다.

보통 DIContainer는 register(등록), resolve(요청)을 하는 함수가 있어서, 필요할때 객체를 꺼내서 쓸 수 있으나, 저희 프로젝트에서는 usecase가 하나이므로 위 함수들이 불필요하다고 판단되어 객체를 생성하여 의존성 주입만을 담당하도록 만들었습니다.

```
@Observable
final class DIContainer {

    static let shared = DIContainer()

    private let repositoryProvider: RepositoryProvider

    init() {
        self.repositoryProvider = RepositoryProvider()
    }

    func makeToDoViewModel() -> ToDoViewModel {
        ToDoViewModel(
            pomoToDoUseCase: PomoToDoUseCaseImpl(
                pomoDayRepository: repositoryProvider.pomoDayRepository,
                appConfigRepository: repositoryProvider.appConfigRepository
            )
        )
    }
}
```

2. ToDoViewModel 리팩토링

ToDoViewModel은 어떻게 하면 코드리뷰를 할때 보기 쉬울까 고민해보고 작성해보았습니다.
제안하는 컨벤션은 다음과 같습니다!

```
final class ToDoViewModel: ObservableObject {
```

Pull Request

The screenshot shows a GitHub Code Review page with a comment from Monfi98:

Header부분 분리해주세요.
다음과 같이 분리할 수 있습니다

```
Section(header: HeaderView(headerText: tag.name))

struct ToDoView: View { ... }

fileprivate struct HeaderView: View {

    let headerText: String

    var body: some View {
        VStack {
            Text(headerText)
                .foregroundColor(Color.black)
                .fontWeight(.semibold)
                .font(.system(size: 20))
            Spacer()

            Button {
                // action
            } label: {
                Label("", systemImage: "plus")
            }

        }.frame(width: UIScreen.main.bounds.width-54)
    }
}
```

또한 Button도 아래처럼 클로저 형태로 쓰는 것이 가독성에 좋습니다.

```
Button {
    // code
} label: {
    // code
}
```

GeunwoongSim 3 days ago: 요런식으로도 분리가 가능하네요 클로저형태로 쓰도록하겠습니다!

Code Review

기술 스택

개발 과정

Swift



Haptic

UI



SwiftUI

Storage



Swift Data

Architecture



+ Clean Architecture



Animation

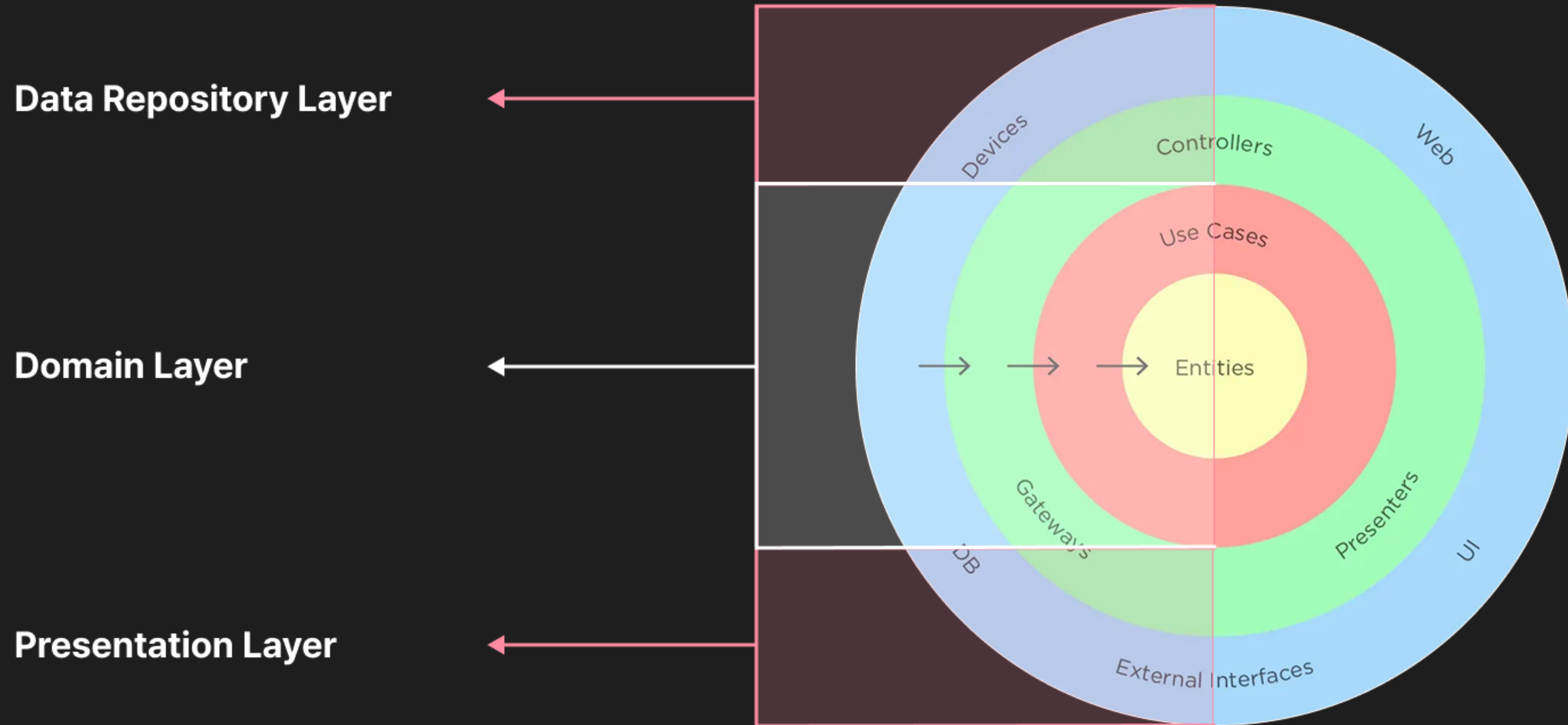
ERD

개발 과정



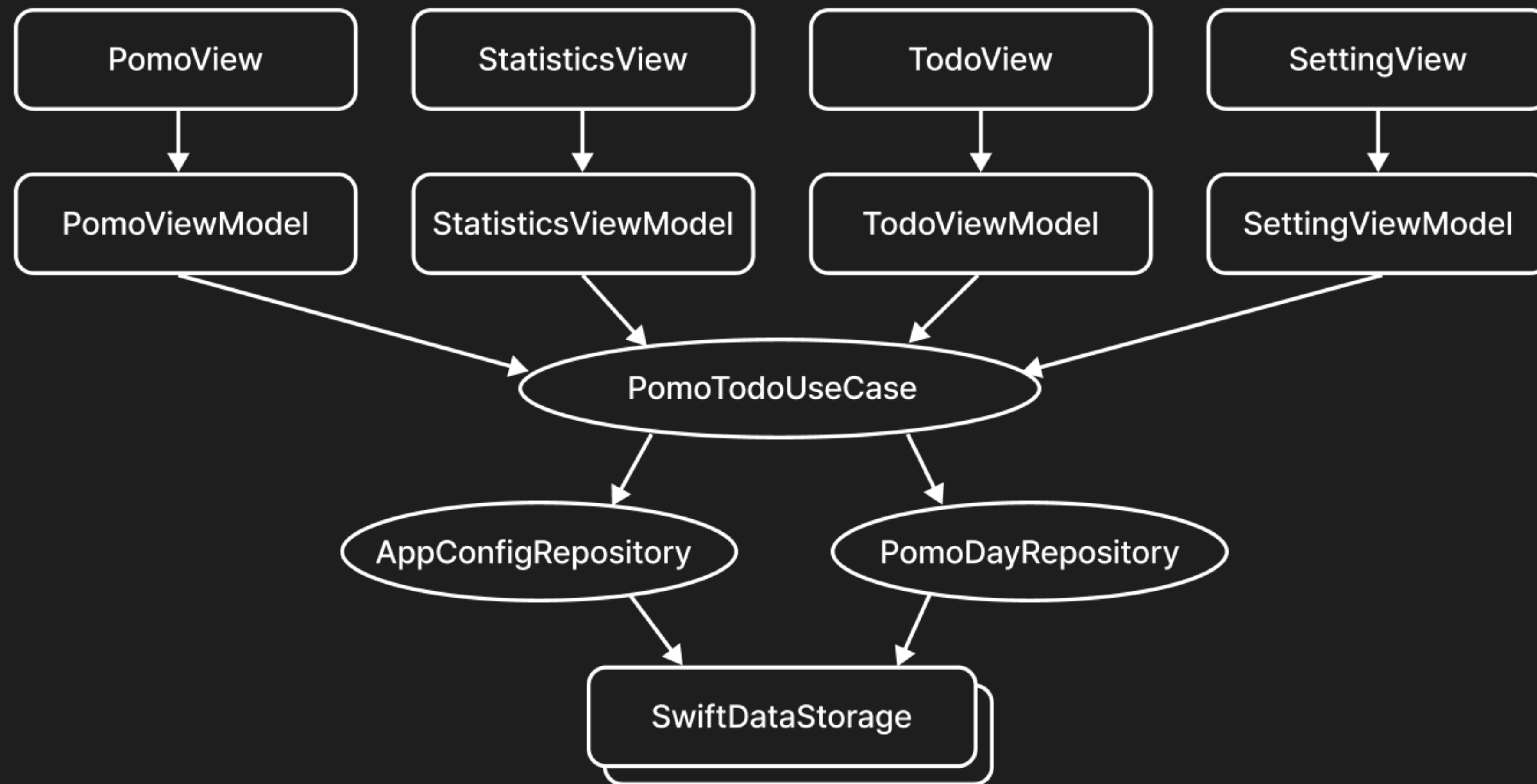
Clean Architecture

개발 과정



Project Architecture

개발 과정



개발 파트

개발 과정



도연

- 탭바 개발
- 포모도로 타이머 뷰 개발
- 포모도로 타이머 기능 개발
- 기기대응 Util 개발



민준

- StatisticsView UI 구현
- StatisticsViewModel 구현 (MVVM 패턴 적용)
- 데이터 로직 및 UseCase 적용
- 태그별 집중 시간 분석 로직 구현

개발 파트

개발 과정



승재

- Data Layer 설계 및 구현
- 비즈니스 로직 구현(UseCase)
- DTO 모델링
- SwiftData CRUD 구현
- 아키텍처 구조 설계



근웅

- 투두 뷰 개발
- 투두 기능 개발
- 설정 뷰 개발
- 설정 기능 개발

PomoTodo

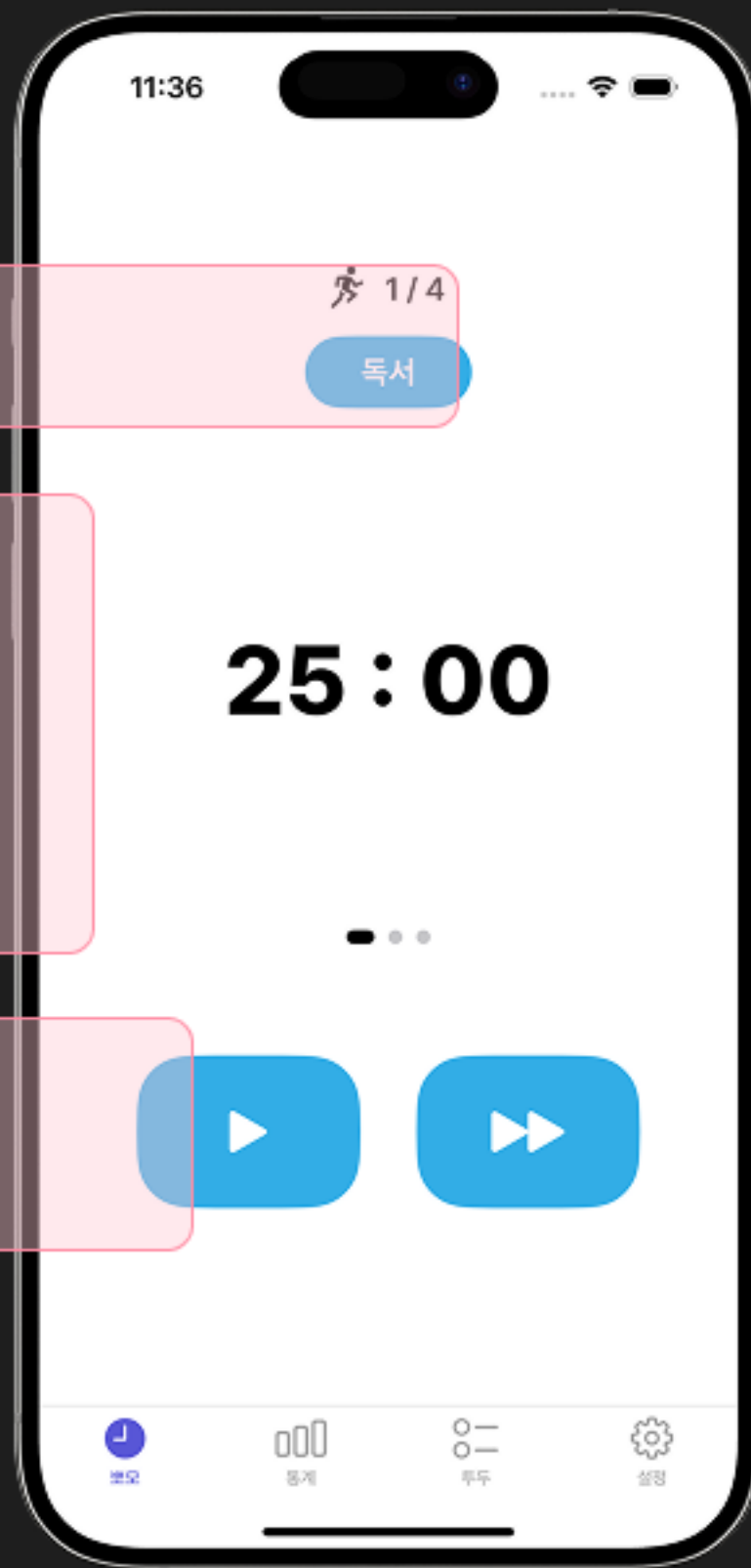
최종 개발물 소개



01 포모도로 타이머

최종 개발물 소개

포모도로 설정 (집중)

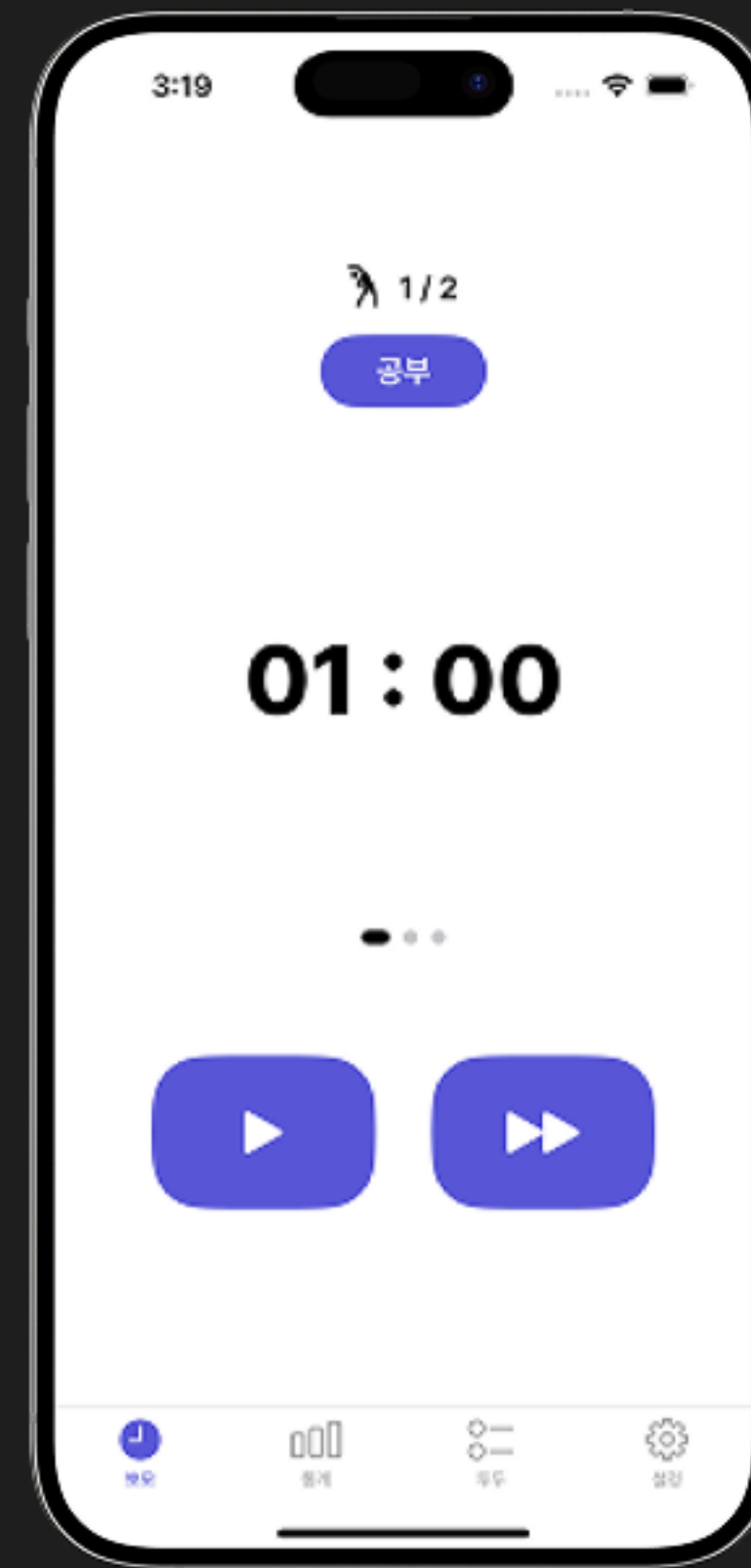


집중/휴식 모드,
포모도로 개수,
태그 표시

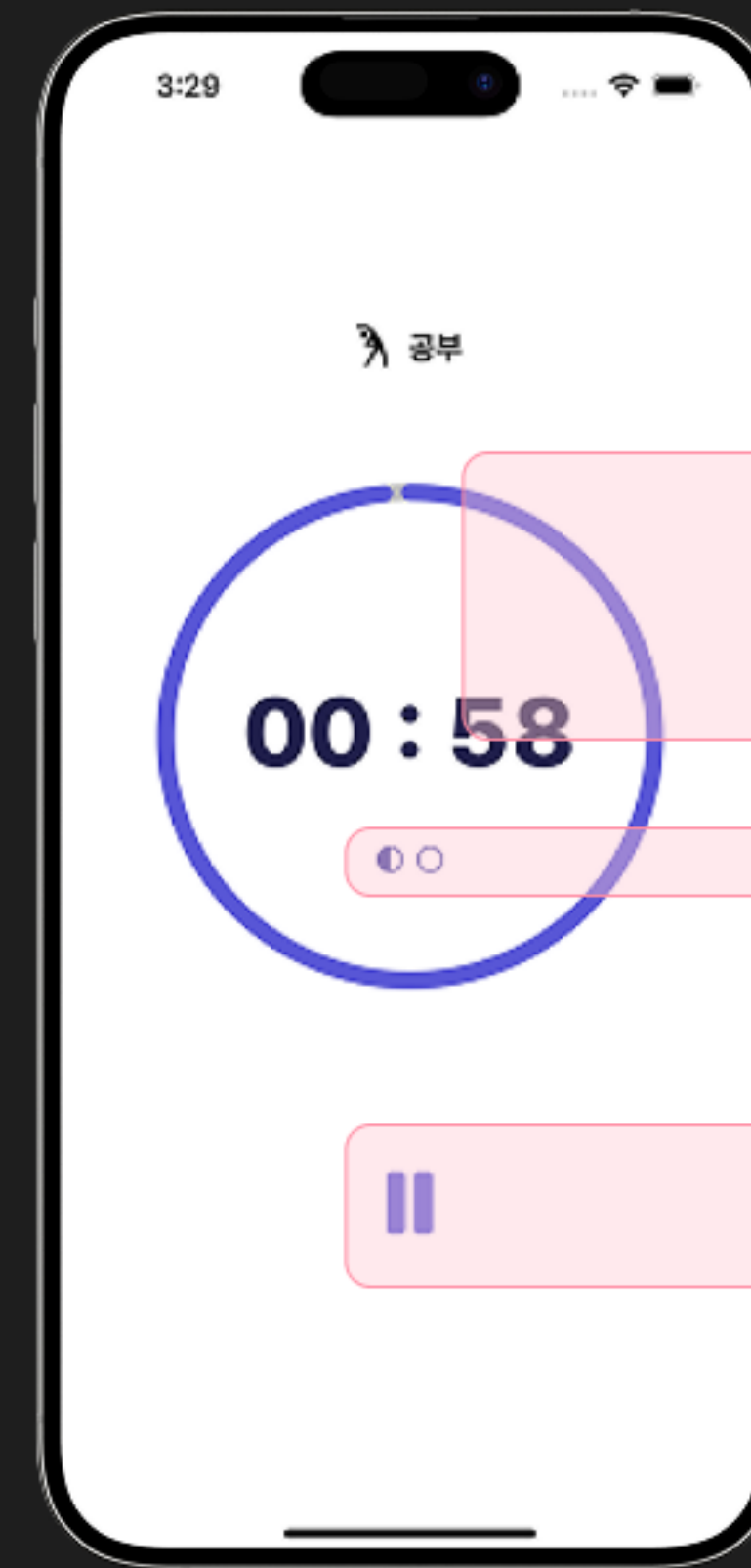
이번에 진행할
포모도로 표시

타이머 시작,
건너뛰기 버튼

포모도로 설정 (휴식)



포모도로 타이머 진행

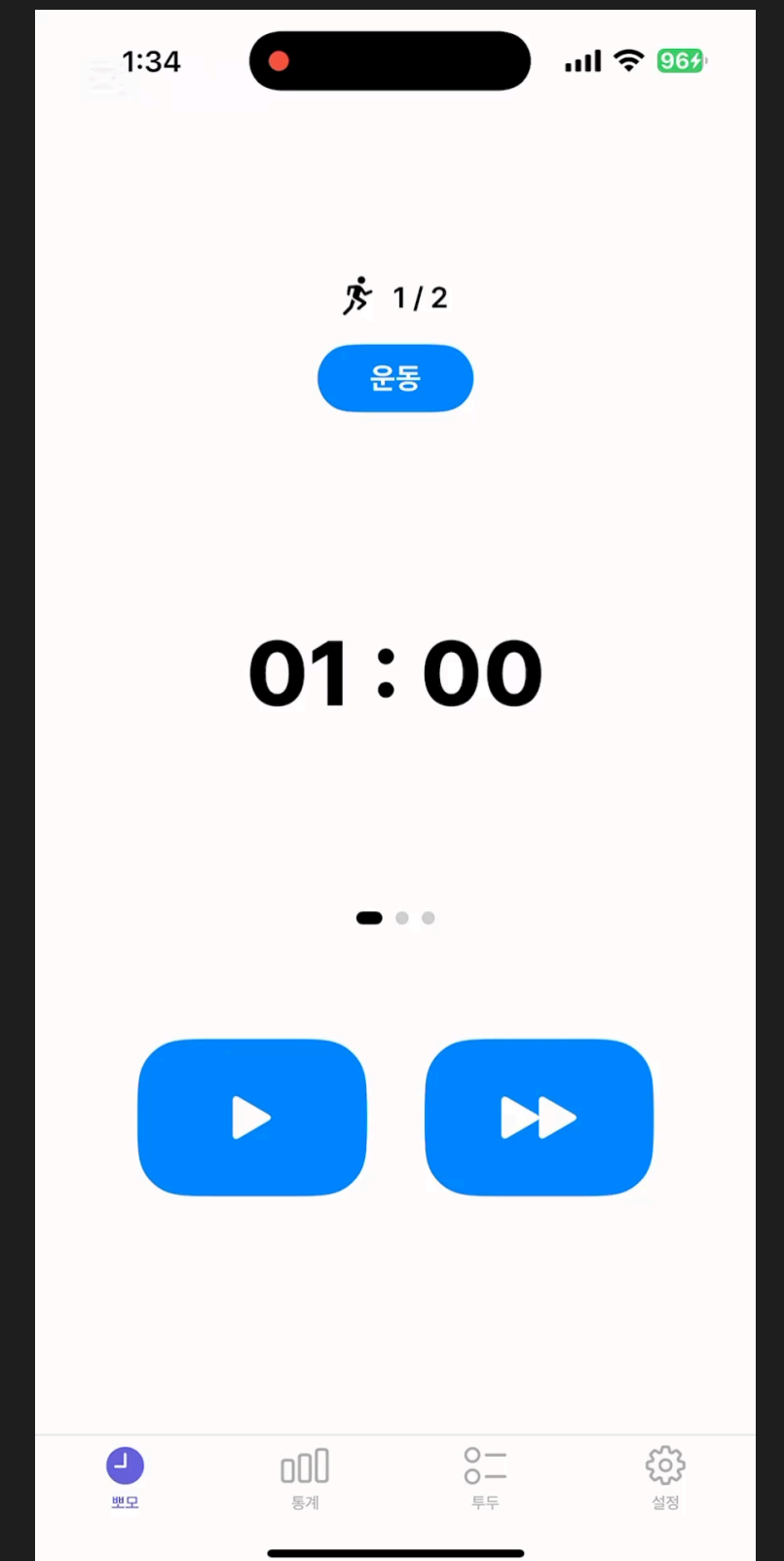


남은 시간 표시

포모도로
진행 현황 표시

타이머 정지 버튼

상세 시연 영상



02 포모도로 통계

최종 개발물 소개

통계 차트



일/주/월 통계 단위 필터링

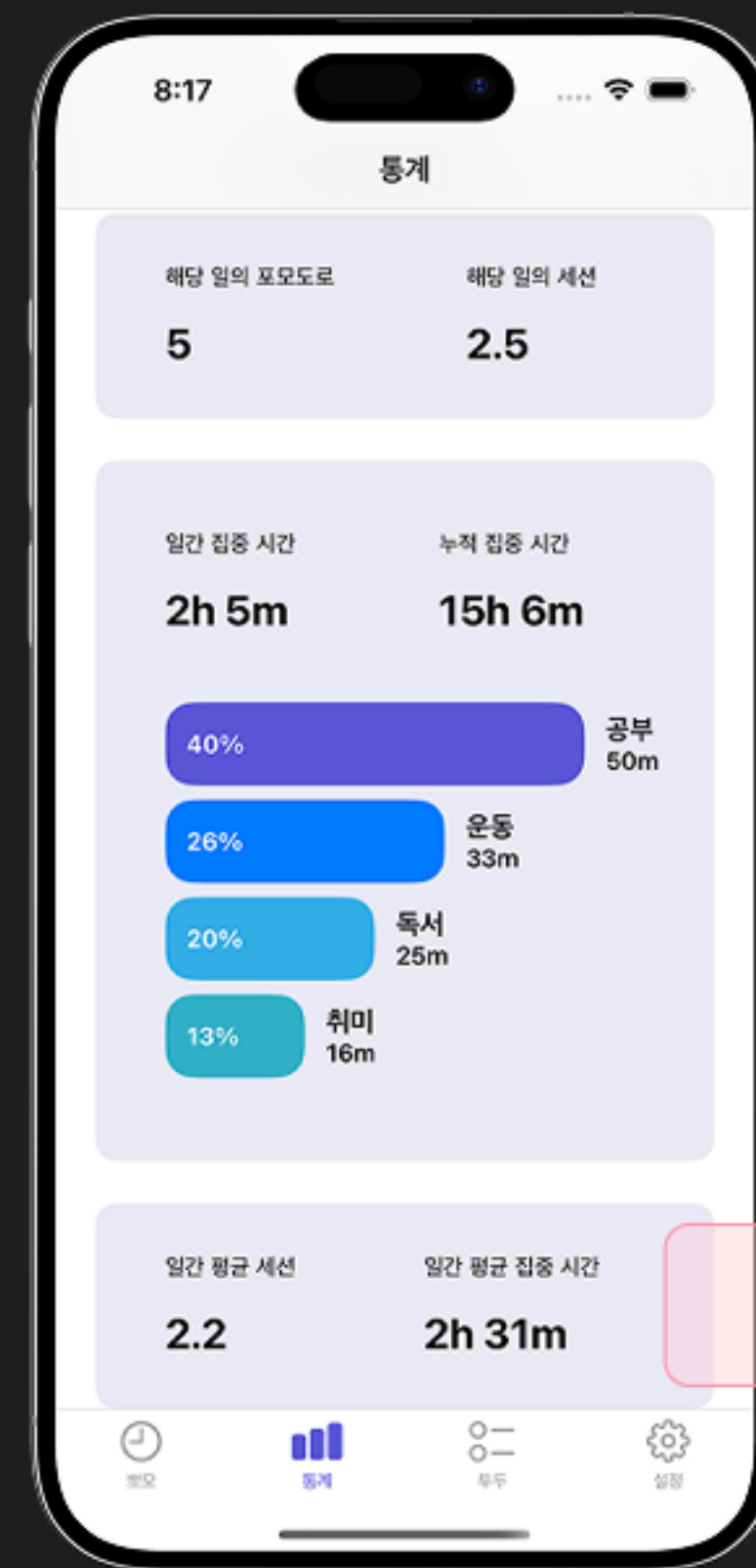
통계 기간 필터링

포모도로 및 세션 개수 표시

집중 시간 표시

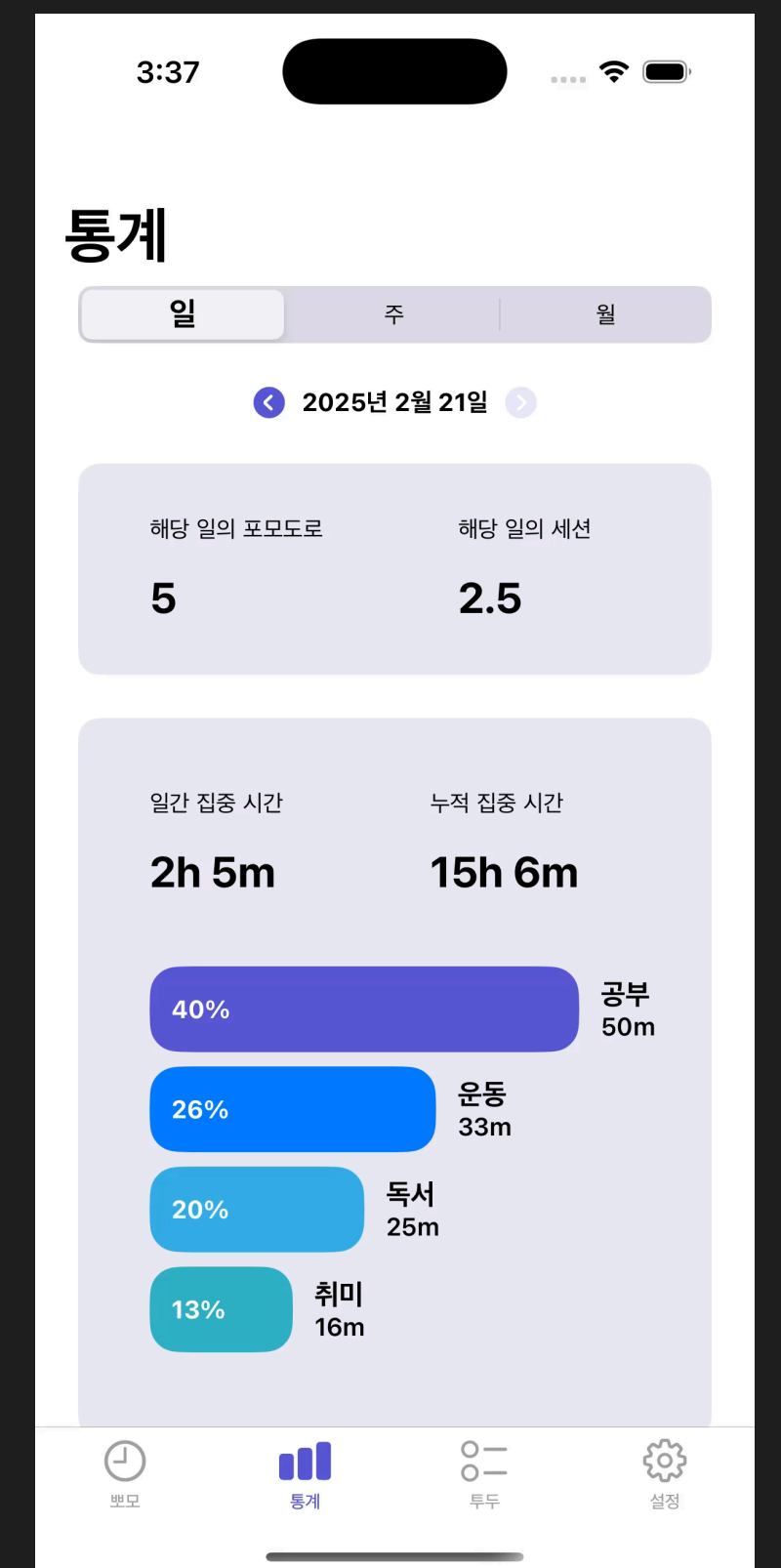
태그 별 집중 시간 표시

통계 차트 (평균)



기간별 평균 세션 및 집중시간

상세 시연 영상



시연 영상
데이터 일자

- 2024-11-28
- 2025-01-07
- 2025-01-22
- 2025-02-13
- 2025-02-20
- 2025-02-21

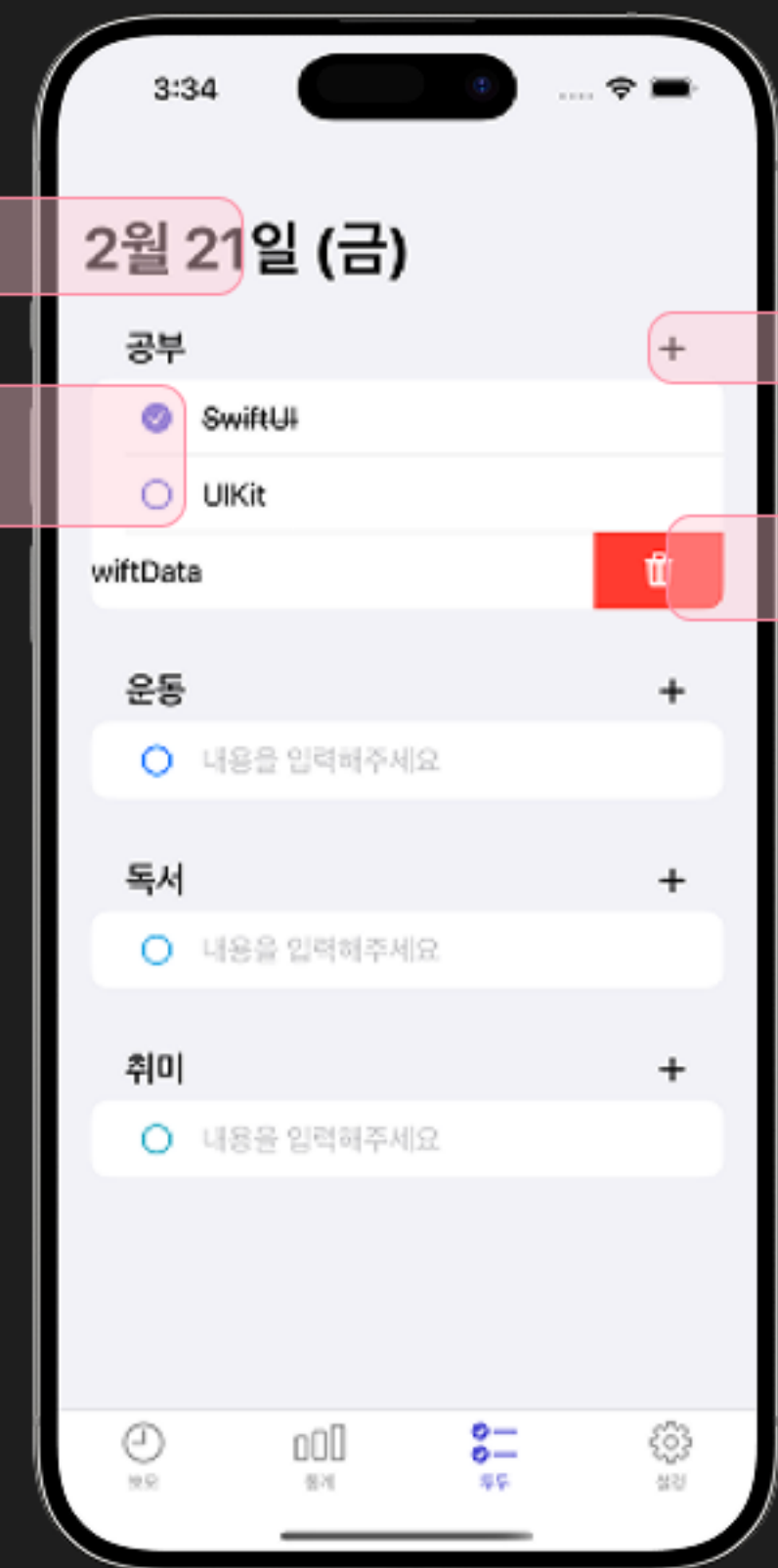
03 투두리스트

최종 개발물 소개

투두리스트 화면

오늘 날짜 표시

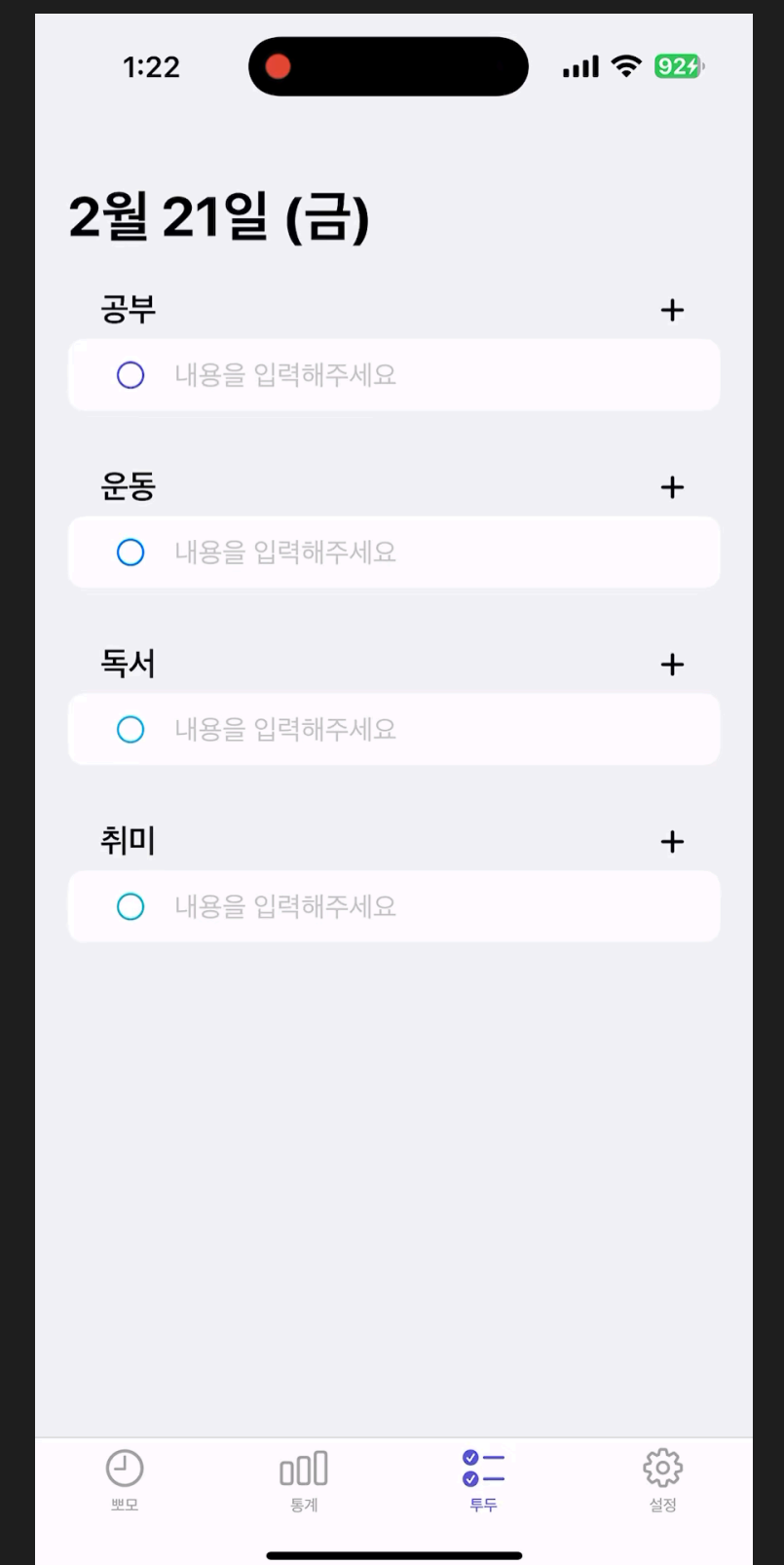
완료 여부 표시



새 항목 추가 버튼

삭제 제스처

상세 시연 영상



04 앱 설정

최종 개발물 소개

설정 기본 화면

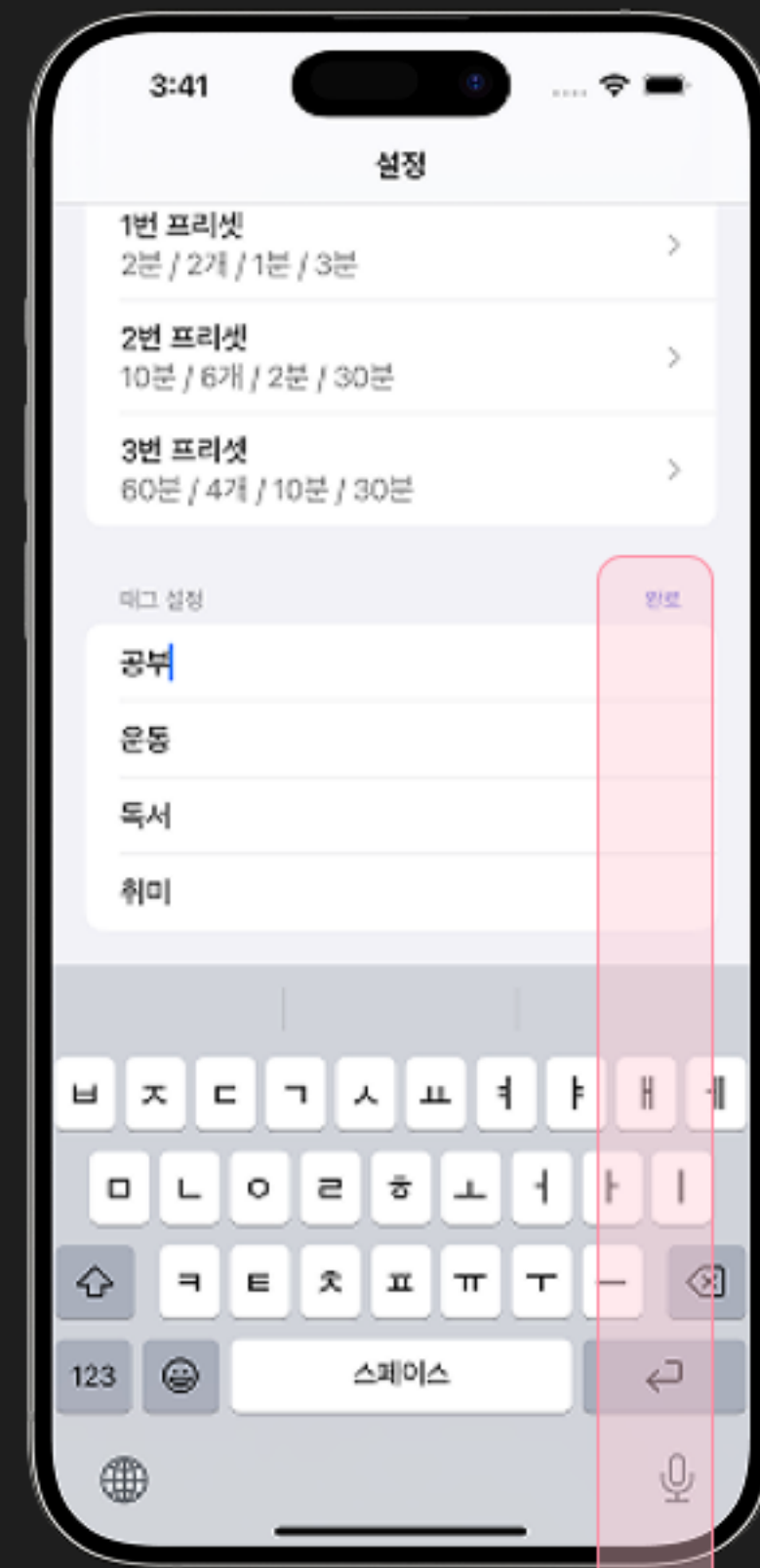


포모도로 타이머 프리셋 목록

현재 태그 목록

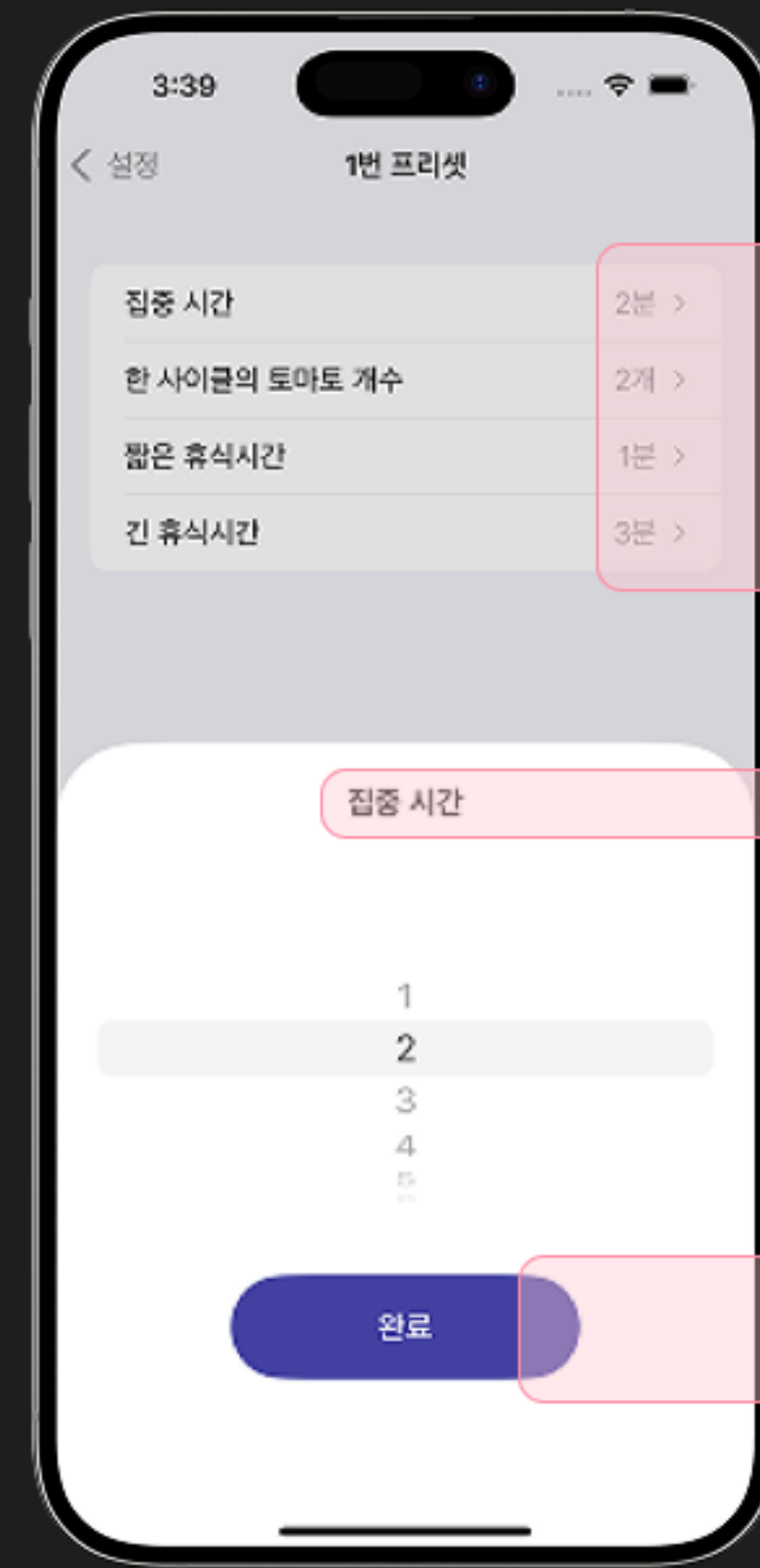
태그 편집 버튼

태그 변경



변경사항 저장 버튼

타이머 시간 변경

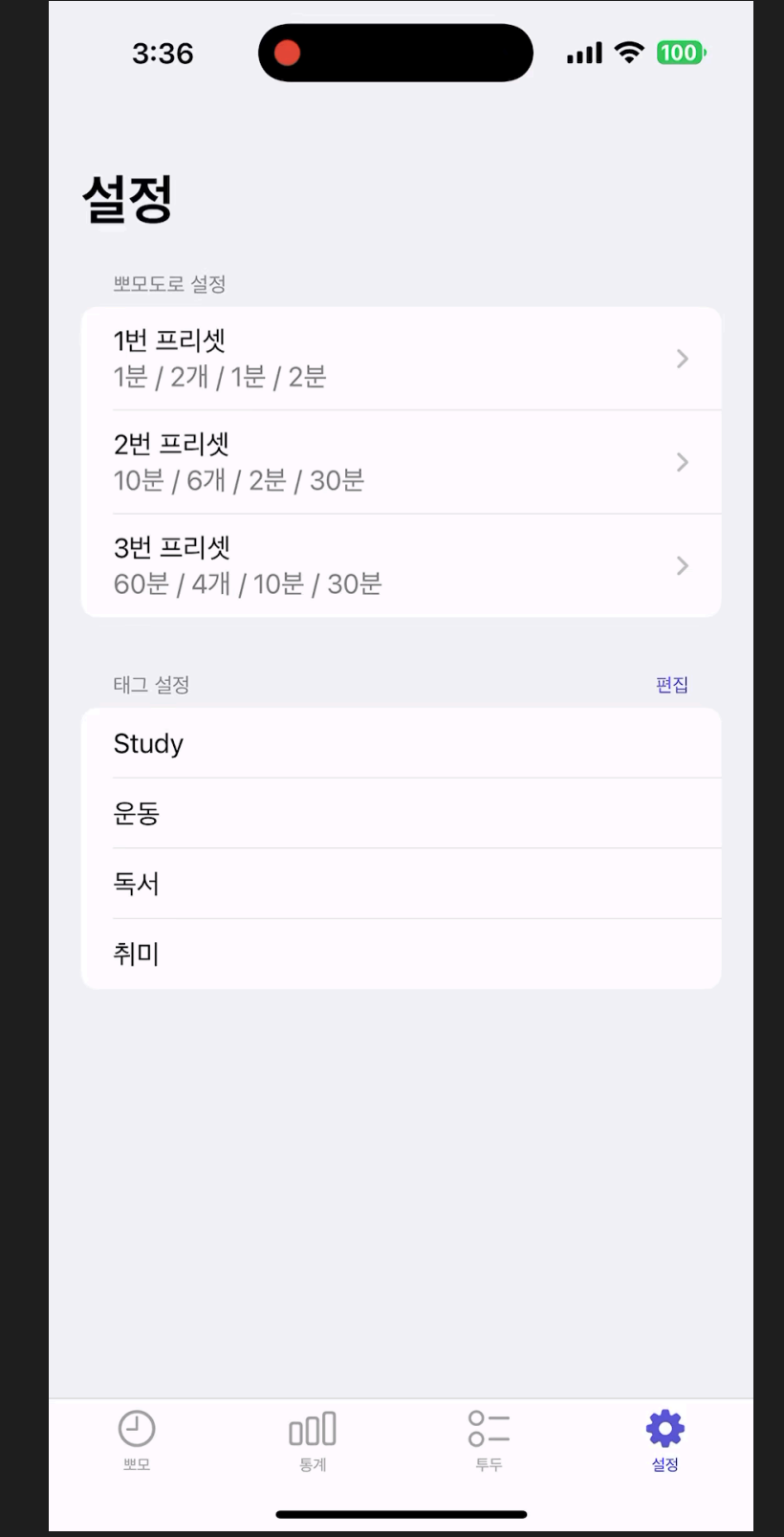


포모도로 타이머 설정 목록

편집 중인 항목 표시

변경 사항 저장 버튼

상세 시연 영상



프로젝트 회고

프로젝트 마무리



다들 밤늦게까지 너무너무 열심히 해주셔서 고맙고 덕분에 기한을 다 지키면서 프로젝트를 깔끔하게 마무리할 수 있었어요!! IEEE팀 최고입니다^^7



혼자서 개발할 때와 달리 협업을 하면서 기획 디자인 개발의 전형적인 흐름을 익힐 수 있어서 좋았습니다. 짧은 기간 마무리를 할 수 있게 열심히 노력한 IEEE팀은 G.O.A.T였습니다!

코드 리뷰를 통해 제대로 된 협업을 진행해 볼 수 있어서 좋은 경험이었고, SwiftUI에서 뷰와 데이터를 어떤 방식으로 다루는게 효율적인지 배울 수 있었습니다.



iOS 개발자 간의 협업은 처음이라 많은 부분에서 모자랐던 것 같습니다. 많이 배웠고, 다들 친절하게 알려주셔서 감사했습니다!



PomoTodo



김도연
도민준
신승재
심근웅