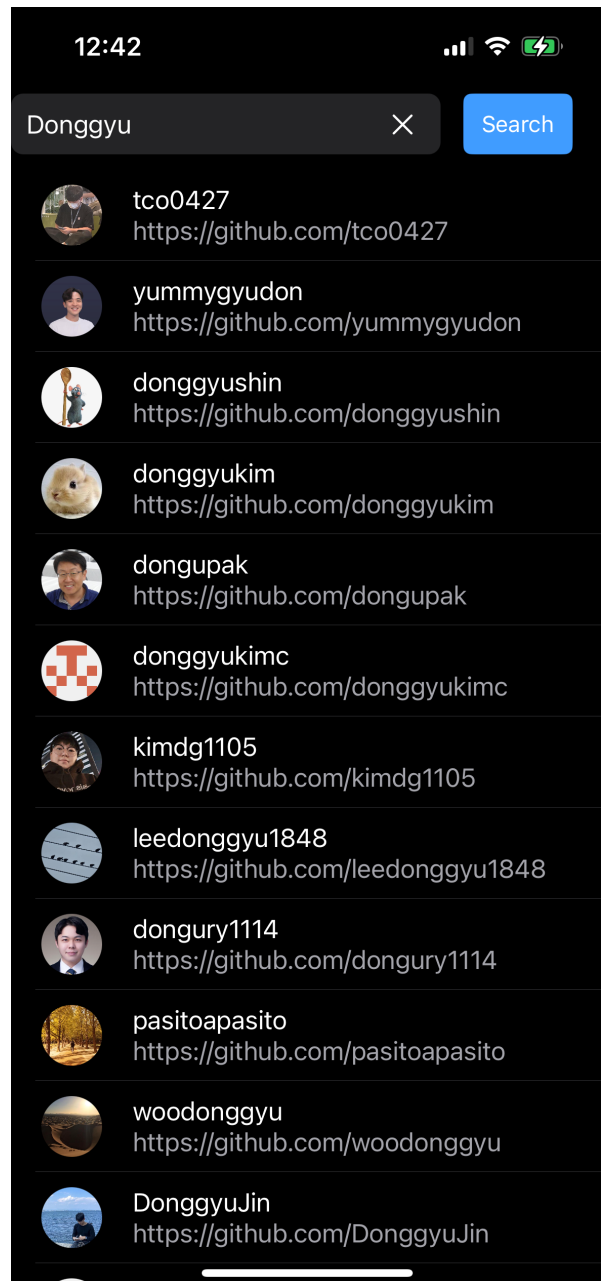
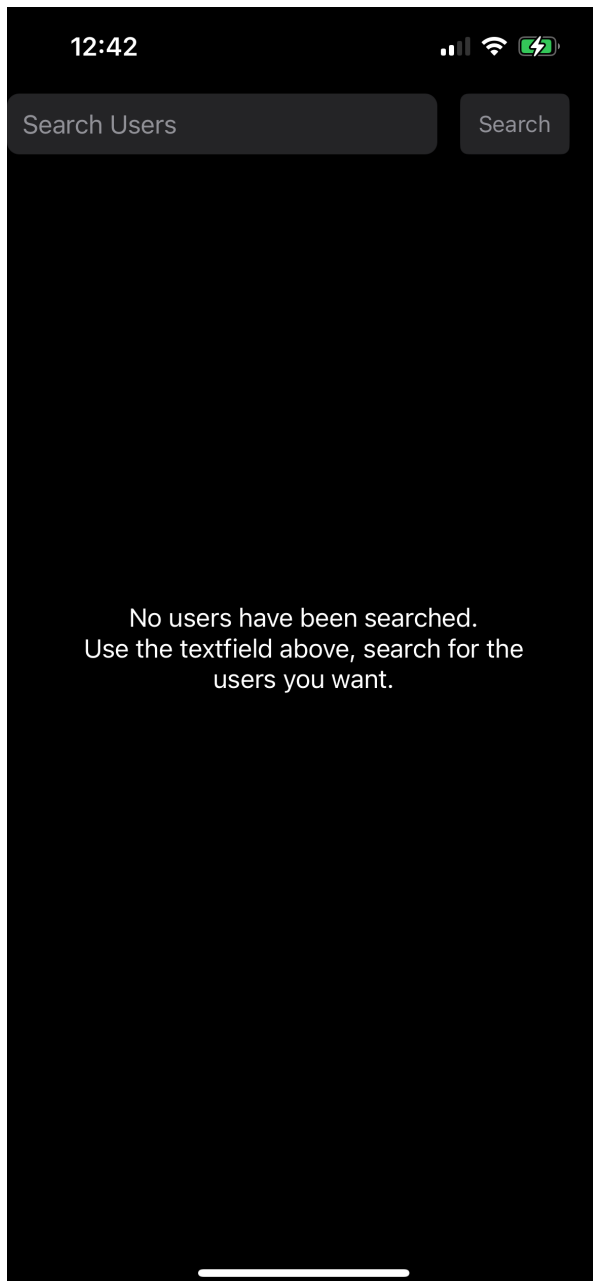
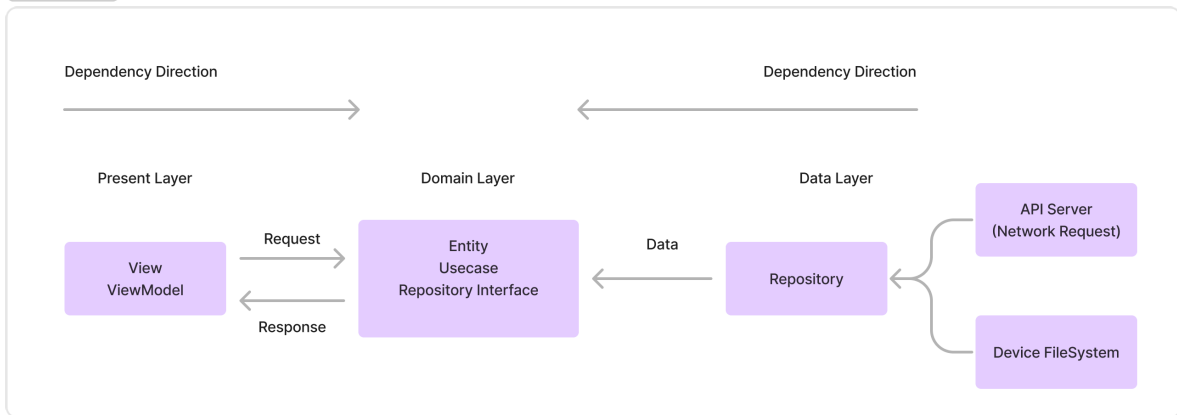


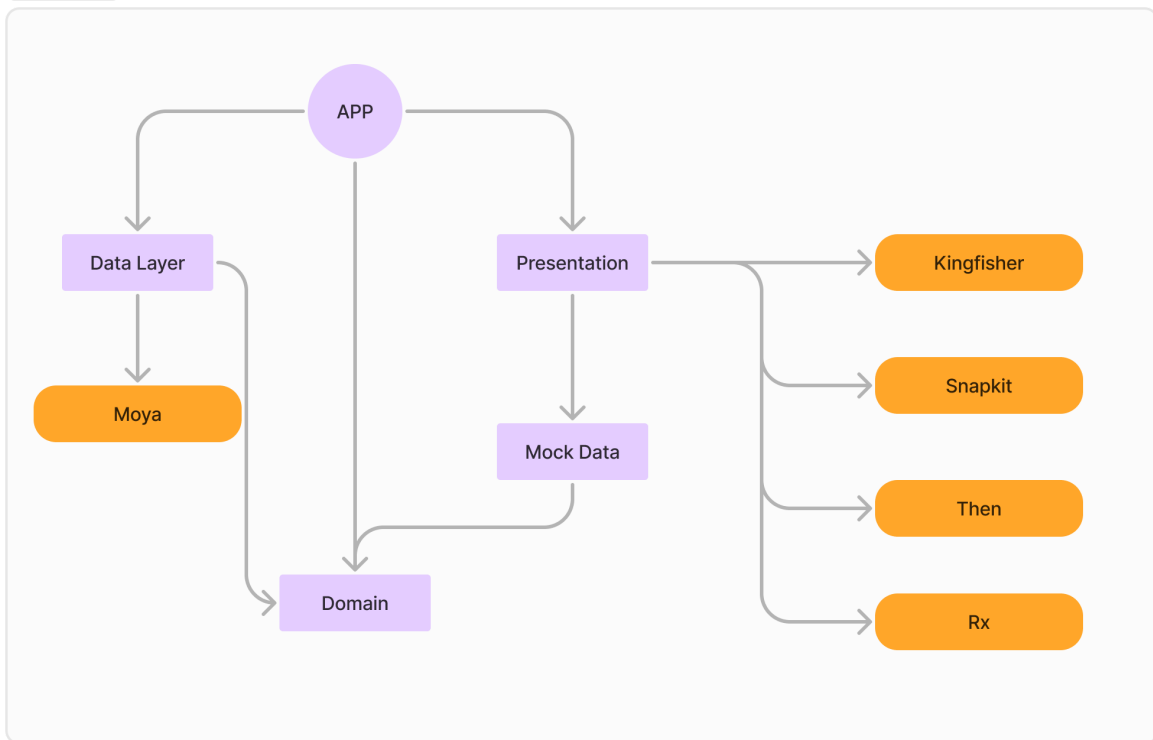
Search App



Architecture



Structure



Domain Layer

애플리케이션의 비즈니스 로직과 규칙을 포함하는 레이어입니다. 이 레이어는 애플리케이션의 핵심 기능과 도메인 모델을 정의합니다.

Role

- 비즈니스 로직을 처리합니다.
- 애플리케이션의 핵심 규칙과 정책을 정의합니다.
- Usecase를 통해 비즈니스 작업을 수행합니다.
- 인터페이스를 통해 **Data Layer** 와 상호작용합니다.

Components

- Model: User 가 있습니다.
- Usecase: 비즈니스 로직을 구현합니다. LogoutUsecase, SearchUserUsecase 등이 있습니다.
- Repository Interface: 데이터 레이어와의 통신을 추상화합니다. AuthRepository, UserRepository 등이 있습니다.

Data Layer

애플리케이션의 데이터 소스와의 상호작용을 담당하는 레이어입니다. 데이터베이스, 네트워크 API, 캐시 등 외부 데이터 소스와의 통신이 포함됩니다.

Role

- 데이터 소스로부터 데이터를 가져오고 저장하는 작업을 담당합니다.
- Repository 패턴을 사용하여 데이터 소스의 구체적인 구현을합니다.
- 네트워크 요청을 처리하고, API 응답을 파싱합니다.
- 로컬 데이터베이스(이 프로젝트에서는 키체인)와의 상호작용을 관리합니다.

Component

- Repository Implementations: 데이터 소스의 구현체. UserRepositoryImpl, AuthRepositoryImpl 등이 있습니다.
- Moya API Client: 네트워크 요청을 처리하는 클라이언트로 MoyaGithub, MoyaGithubAPI 등이 있습니다.
- TokenKeychainManager: Github access token 을 저장하고 관리하는 매니저입니다.

Presentation Layer

사용자 인터페이스와 상호작용을 담당합니다. 데이터를 화면에 표시하고 사용자의 입력을 처리합니다.

Role

- 사용자 인터페이스를 구성합니다.
- 사용자 입력을 처리하고, 그에 따라 애플리케이션의 상태를 업데이트합니다.
- ViewModel을 사용하여 뷰와 비즈니스 로직을 연결하고 UI 상태를 관리합니다.

Components

- ViewController/View: 사용자 인터페이스를 구성합니다.
- ViewModel: 뷰와 비즈니스 로직을 연결합니다.

MockData

개발 및 테스트 단계에서 활용하기 위한, 통제 가능한 데이터 셋팅을 해둡니다. Xcode15 Feature인 Preview 기능을 사용 가능하게 해주고, 테스트 코드에서도 활용되어질 수 있습니다.

Protocols

```
public protocol AuthRepository {
    var accessToken: AnyPublisher<String?, Never> { get }

    func requestAccessToken(code: String) async throws
    func logout()
}

public protocol UserRepository {
    func get(query: String, page: Int, perPage: Int) async throws -> ([User], totalCount: Int)
}
```

- AuthRepository : 유저의 인증과 관련된 데이터 소스와의 상호작용을 추상화하였습니다.
- UserRepository: 유저 데이터와 관련된 데이터 소스와의 상호작용을 추상화하였습니다.

데이터 소스와의 상호작용을 추상화하여 실제 데이터소스계층과의 격리를 통해 MockData 를 이용한 Preview 기능 및 테스트블한 코드 작성이 가능해집니다.