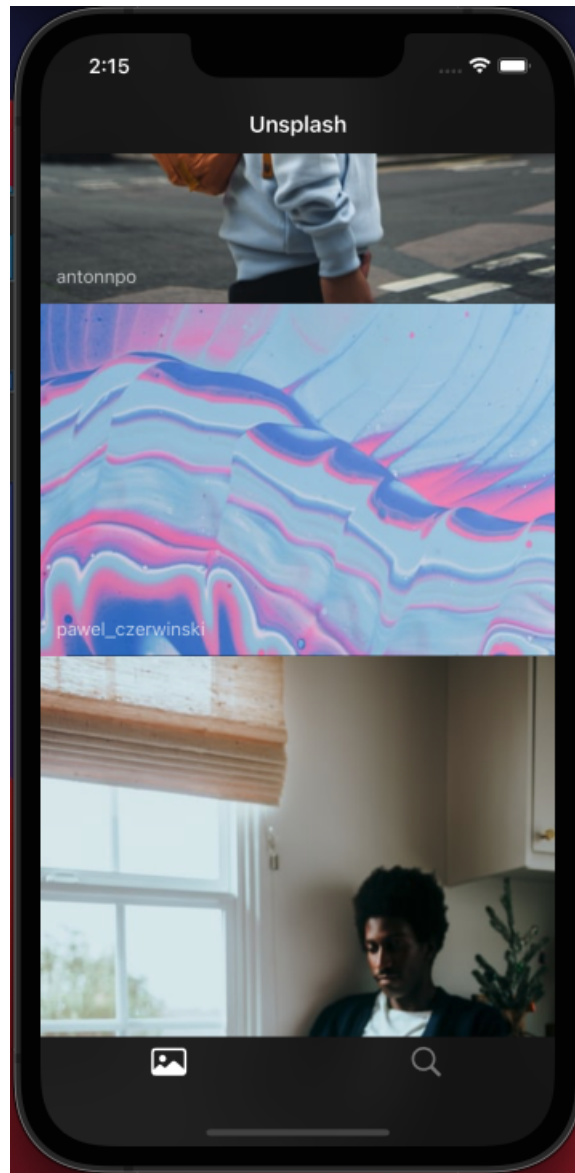


김종권_설계_Unsplash_App

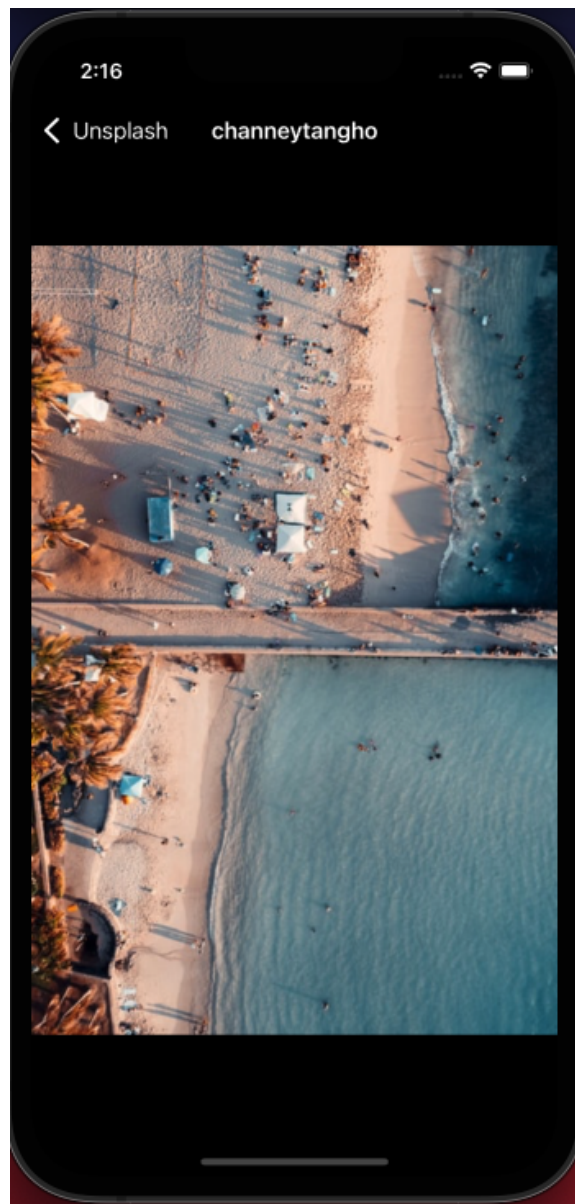
기능 구현

기능1. 사진 목록



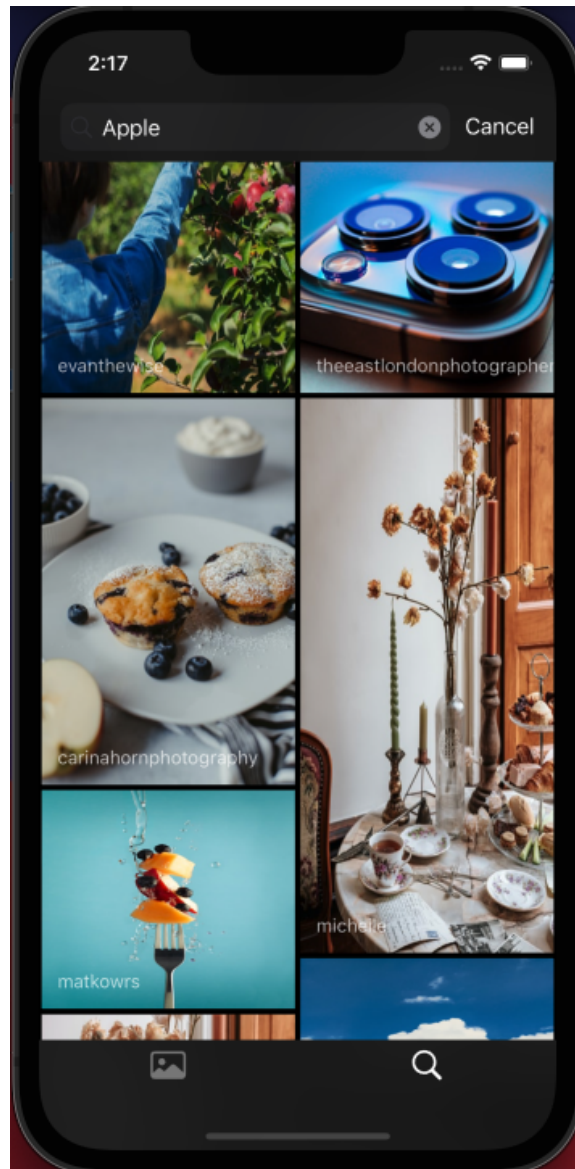
- ✓ 스크롤에 따라 사진이 자동으로 로드
- ✓ 사진을 터치하면 해당 사진이 상세 화면으로 표출

기능2. 사진 상세보기



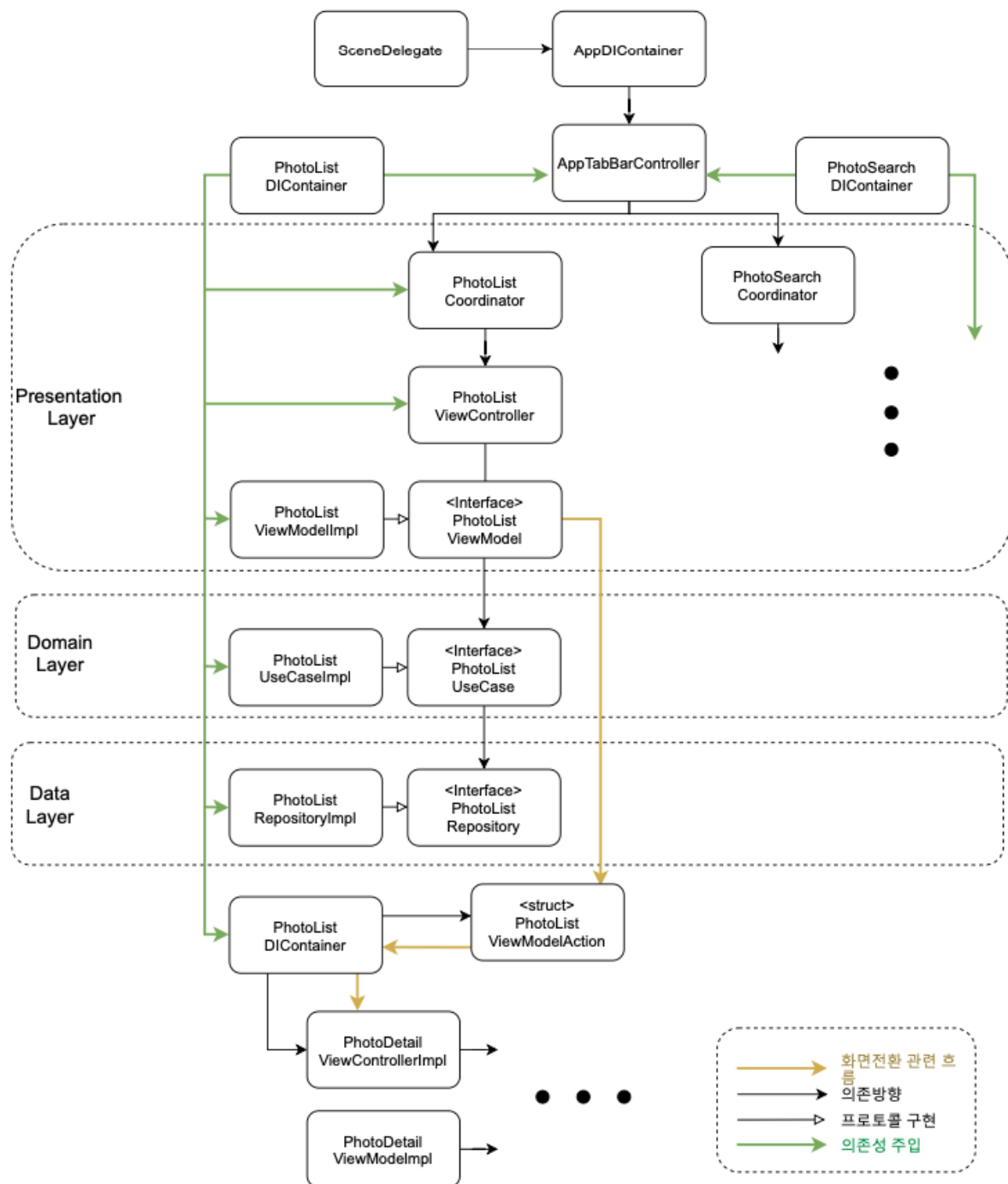
- ✓ 목록에서 선택한 사진 표시
- ✓ 좌우로 스와이프하여 사진 목록의 이전/다음 이미지 표시
- ✓ 닫기 시 사진 목록으로 되돌아가고 상세에서 마지막으로 본 사진이 목록에 바로 표시

기능3. 사진 검색



- ✓ 키워드로 사진 검색
- ✓ 검색 후 동작은 기능1, 2와 동일

앱 아키텍처 (MVVM-C)



1) Presentation Layer

- ViewController, ViewModel, ViewModelAction
- 플랫폼 의존성이 없고 사용자의 입력에 대한 행동만을 정의

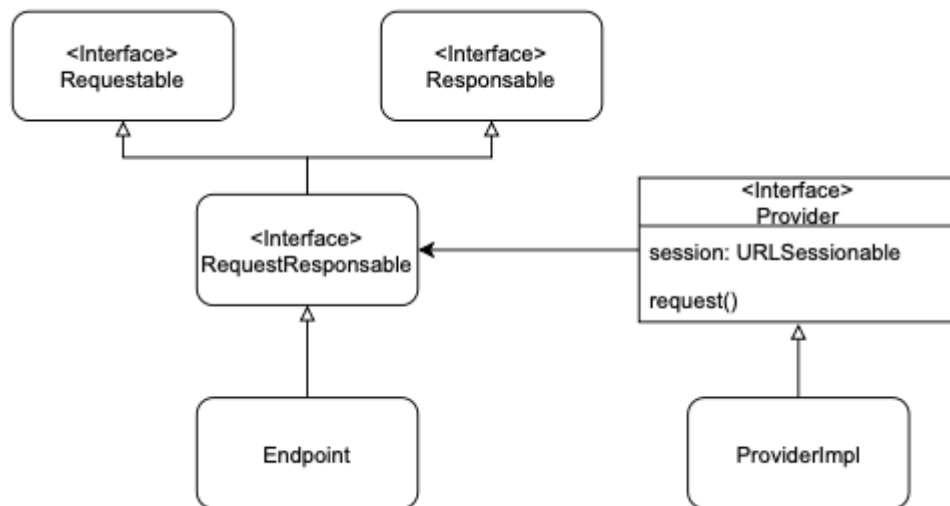
2) Domain Layer

- UseCase
- 가장 중요한 비즈니스 로직이 있는 모듈이기 때문에, 되도록 외부에 의존하지 않고 만약 의존한다면 Protocol에 의존하여 의존성을 역전시키는 코드로 적용
(ex - Domain Layer에서 Data Layer 의존: UseCase가 인터페이스 Repository에 의존)

3) Data Layer

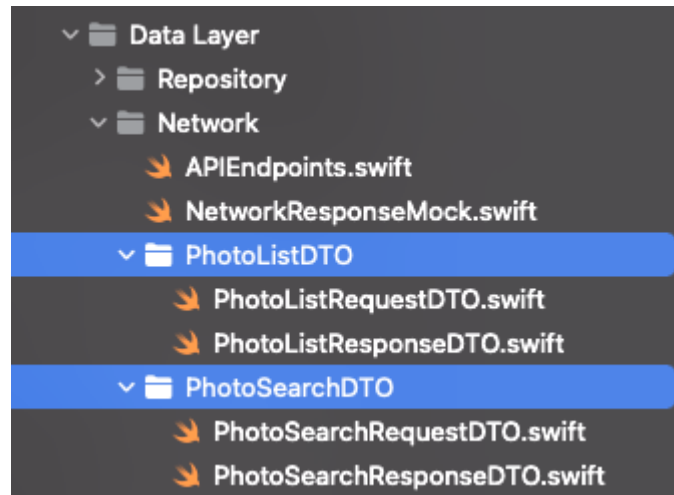
- RepositoryImpl
- Network, DB, Storage등에 접근하여 데이터 입출력을 담당하는 곳

네트워크 설계



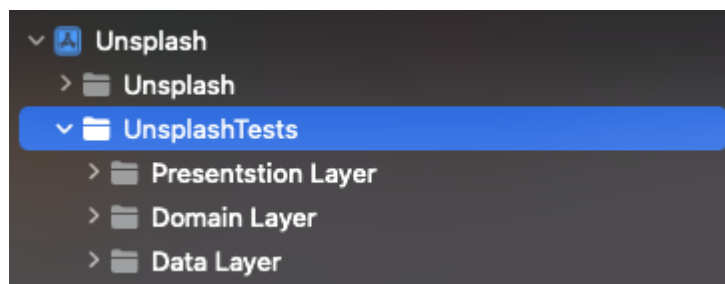
- **Responsible**: associativetype Response만 선언되어 있으며, Endpoint객체를 만들 때 Response 타입을 지정하여, Provider의 request에서 제네릭 추론 타입을 따로 넘기지 않아도 Endpoint하나만 있으면 추론이 되도록 설계
- **Endpoint**: Requestable, Responsible의 성격을 가지고 있으며, 네트워크에 접근하는 Provider의 파라미터로 사용되는 컴포넌트
- **URLSessionable**: URLSession을 wrapping한 프로토콜로 사용되며, URLSession의 `dataTask()`메소드를 testable하도록 만드는 컴포넌트

DTO 모델

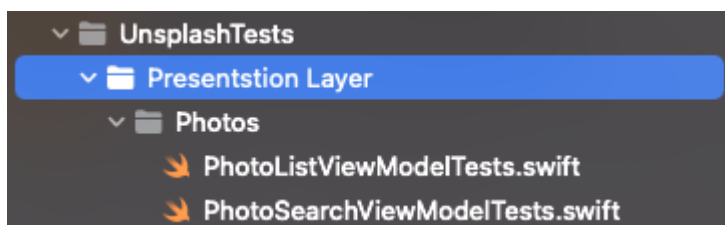


- Network Request, Response에서만 사용되는 객체이고, Network에 연관되어 있으므로 Data Layer 그룹 하위에 존재
- Domain Layer에서 Data Layer로의 인터페이스인 RepositoryImpl에서 받아서 Domain Layer로 넘겨줄때는 Domain의 Entity객체로 변환하여 UseCase에 반환

유닛 테스트

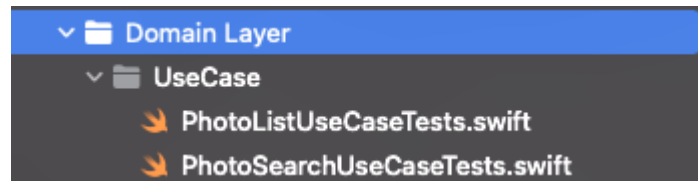


1) Presentation Layer



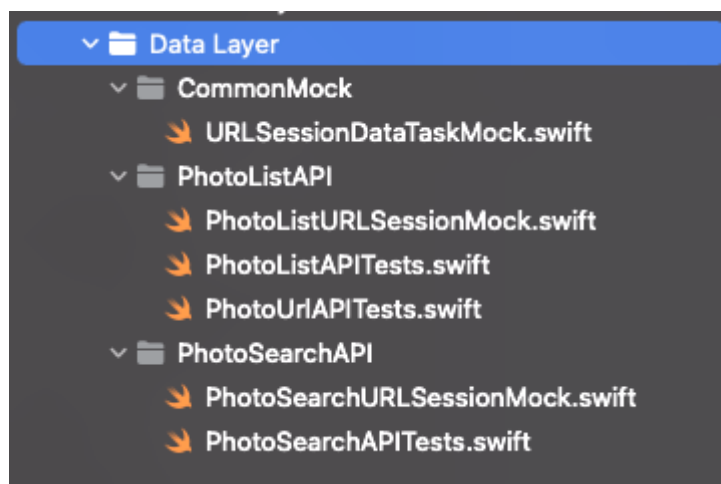
- **_ViewModelTests**: UseCase를 Mock으로 놓고 View의 Input이 주어졌을 때, ViewModel의 Output 테스트

2) Domain Layer



- **_UseCaseTests**: Network에 접근하는 Repository를 Mock으로 만들어서 UseCase에 주입한 후, UseCase를 실행했을 때 실행하면 기대한 응답값이 맞는지 테스트

3) Data Layer



- **URLSessionDataTaskMock**: URLSessionDataTask객체에서 resume()이 호출 될 때, call back을 실행해보기 위한 URLSessionDataTask의 서브 클래스
- **_URLSessionMock**: 네트워크에 URLSessionable을 준수하는 클래스이며 dataTask()메소드를 재정의하여 HTTPURLResponse의 데이터를 mocking하는 용도
- **_APITests**: mock용도의 특정 endpoint로 request했을 때 기대하는 response가 내려 오는지에 대한 테스트