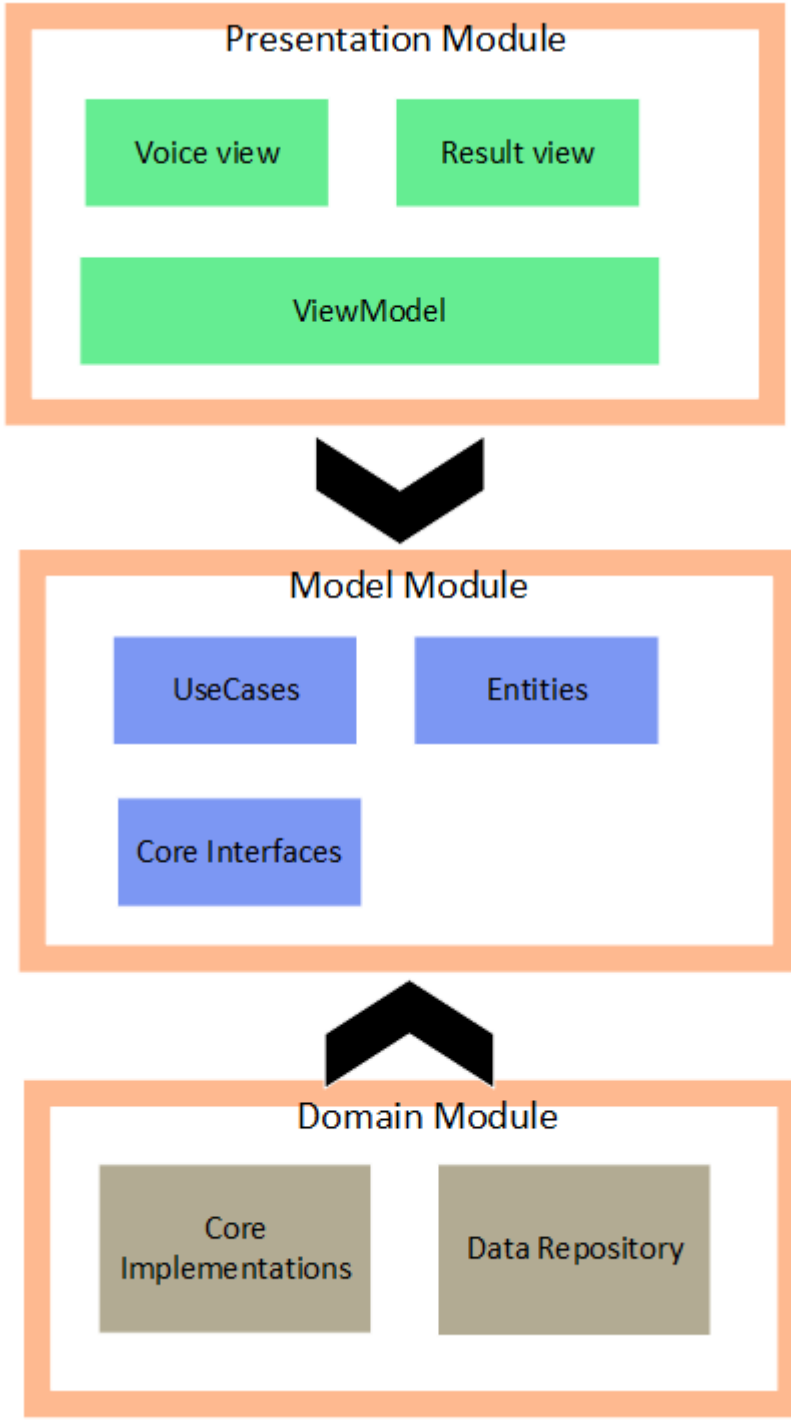


**Presentation Module**은 사용자 인터페이스(UI)와 관련된 모든 것을 포함합니다. 이 모듈은 사용자가 애플리케이션과 상호작용하는 부분을 담당합니다.

- **Voice view**: 음성 입력을 처리하고 사용자에게 음성 피드백을 제공하는 UI 요소입니다. 예를 들어, 음성 명령을 인식하고 결과를 화면에 표시하는 역할을 합니다.
- **Result view**: 사용자가 요청한 작업의 결과를 표시하는 UI 요소입니다. 예를 들어, 검색 결과나 계산 결과를 화면에 보여줍니다.
- **ViewModel**: UI와 비즈니스 로직 사이의 중간자 역할을 합니다. 데이터 바인딩을 통해 UI에 데이터를 제공하고, 사용자 입력을 처리하여 비즈니스 로직에 전달합니다.

**Model Module**은 애플리케이션의 비즈니스 로직과 관련된 모든 것을 포함합니다. 이 모듈은 애플리케이션의 핵심 기능을 담당합니다.

- **UseCases**: 특정 비즈니스 로직을 수행하는 클래스입니다. 예를 들어, 사용자가 로그인하거나 데이터를 검색하는 기능을 담당합니다. UseCase는 애플리케이션의 동작을 정의하고, 엔티티와 상호작용합니다.
- **Entities**: 애플리케이션의 핵심 데이터 구조를 정의합니다. 예를 들어, 사용자 정보나 제품 정보 같은 데이터 모델이 여기에 해당합니다. 엔티티는 비즈니스 규칙을 포함하고 있습니다.
- **Core Interfaces**: UseCase와 엔티티가 상호작용하는 데 필요한 인터페이스를 정의합니다. 이를 통해 모듈 간의 결합도를 낮추고, 코드의 유연성과 테스트 용이성을 높입니다.

**Domain Module**은 애플리케이션의 데이터 접근과 관련된 모든 것을 포함합니다. 이 모듈은 데이터베이스나 외부 API와의 상호작용을 담당합니다.

- **Core Implementations**: Core Interfaces를 구현한 클래스들입니다. 예를 들어, 데이터베이스에서 데이터를 가져오거나 저장하는 로직을 포함합니다.
- **Data Repository**: 데이터 소스와 상호작용하는 클래스입니다. 예를 들어, 데이터베이스나 네트워크에서 데이터를 가져와서 UseCase에 제공하는 역할을 합니다. Repository 패턴을 사용하여 데이터 접근 로직을 추상화하고, 데이터 소스가 변경되더라도 비즈니스 로직에 영향을 주지 않도록 합니다.