***Layer Relationships***

1. **Controllers** -> call **Application Layer** (or Service Layer if Application Layer is not needed)
2. **Application Layer** -> calls **Service Layer** and **Mapping Layer**
3. **Service Layer** -> calls **Business Logic Layer**, **Repository Layer**, and **Unit of Work**
4. **Mapping Layer** -> maps DTOs to and from entities
5. **Infrastructure Layer** -> accessed as needed for external dependencies

***Typical Architecture Summary***

Here’s how a typical layered architecture might look with these additional layers:

1. Presentation Layer (Controllers in Web API)
2. Application Layer
3. Service Layer
4. Business Logic Layer
5. Validation Layer
6. Repository Layer
7. Unit of Work Layer
8. Mapping Layer
9. Infrastructure Layer (for external dependencies)

***Example of Repository layer***