

Technical Report: Design Patterns in Health Information Network

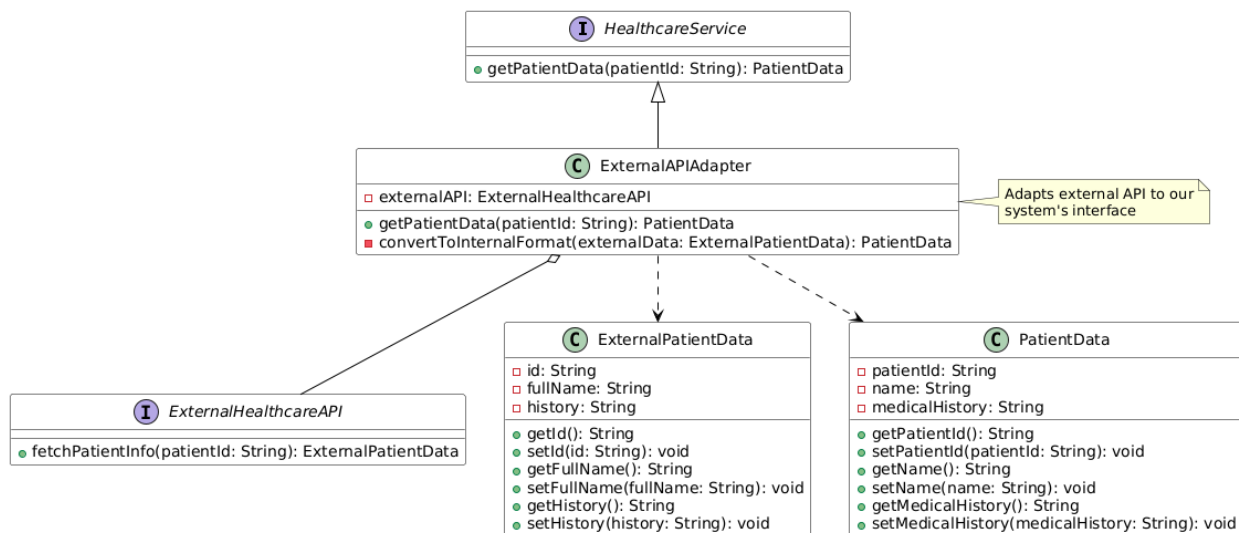
1. Adapter Pattern

The Adapter pattern is used to allow incompatible interfaces to work together. In our healthcare system, we implemented this pattern to integrate external healthcare APIs with our internal system.

Implementation Details

- **Purpose:** To bridge the gap between external healthcare APIs and our internal system's interface
- **Components:**
 - ExternalAPIAdapter: Adapts external API calls to our system's interface
 - HealthcareService: Our system's interface
 - ExternalHealthcareAPI: External API interface
 - PatientData and ExternalPatientData: Data transfer objects

UML Diagram



2. Bridge Pattern

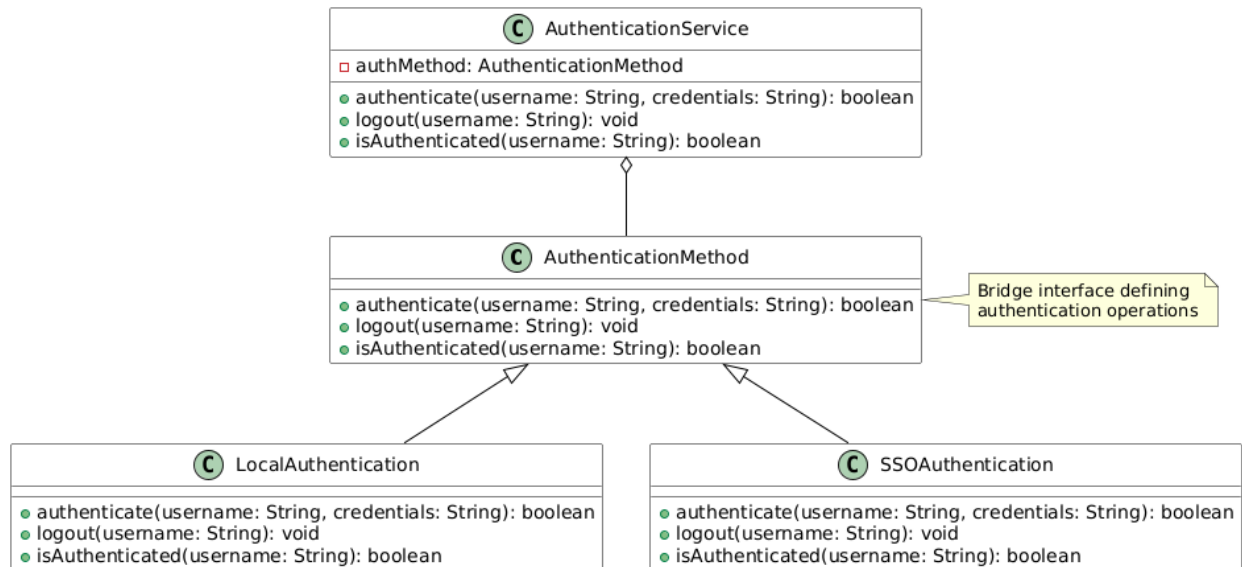
The Bridge pattern is used to separate abstraction from implementation, allowing both to vary independently.

Implementation Details

- **Purpose:** To decouple the abstraction of healthcare services from their implementation
- **Components:**
 - Abstraction: Defines the high-level interface

- Implementation: Defines the low-level interface
- Concrete Implementations: Specific implementations of healthcare services

UML Diagram



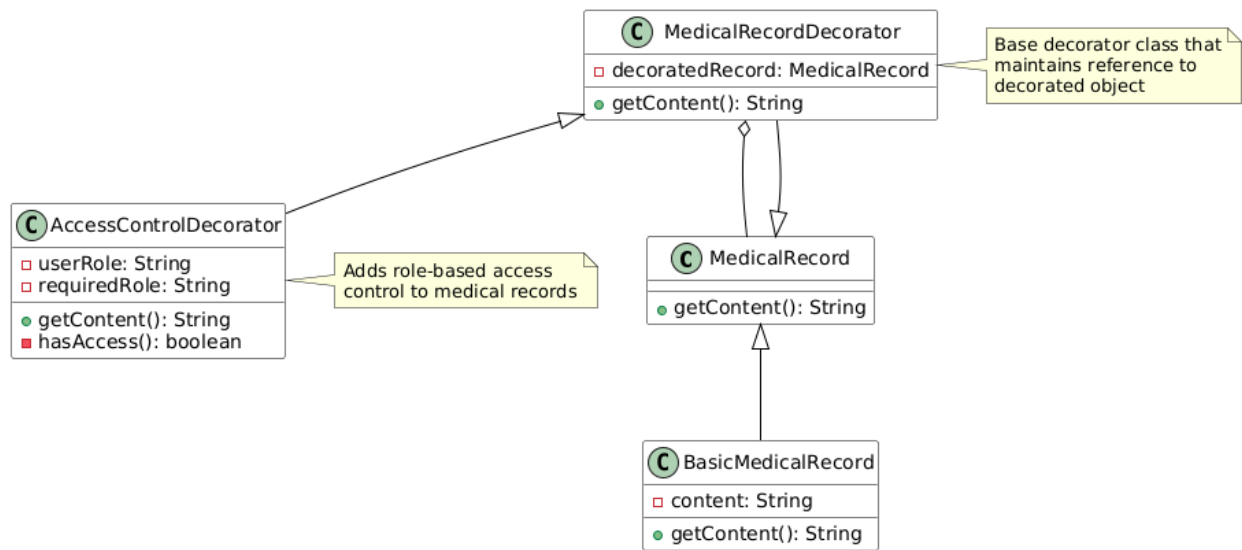
3. Decorator Pattern

The Decorator pattern is used to add responsibilities to objects dynamically without affecting other objects.

Implementation Details

- **Purpose:** To add additional functionality to healthcare services at runtime
- **Components:**
 - Base Component: Core healthcare service
 - Decorators: Additional service layers
 - Concrete Decorators: Specific enhancements

UML Diagram



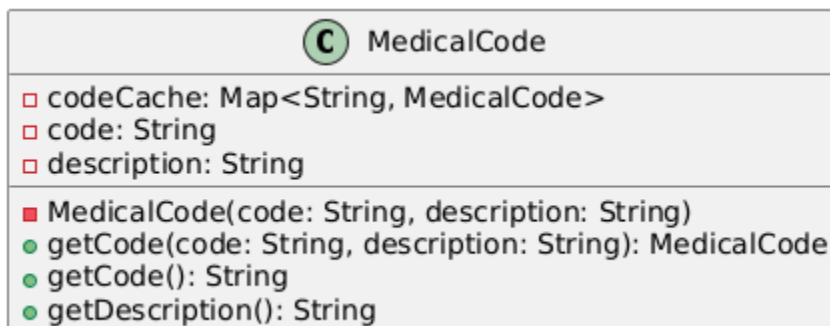
4. Flyweight Pattern

The Flyweight pattern is used to minimize memory usage by sharing as much data as possible with similar objects.

Implementation Details

- **Purpose:** To optimize memory usage when dealing with large numbers of similar healthcare records
- **Components:**
 - Flyweight: Shared data structure
 - Flyweight Factory: Manages and creates flyweight objects
 - Client: Uses flyweight objects

UML Diagram



Conclusion

These design patterns have been implemented to address specific challenges in our healthcare information system:

- **Adapter:** Enables integration with external systems
- **Bridge:** Provides flexibility in service implementation
- **Decorator:** Allows dynamic enhancement of services
- **Flyweight:** Optimizes memory usage for large datasets

Each pattern contributes to making the system more maintainable, scalable, and efficient while following object-oriented design principles.