

Ehiane's Design Table

Principle/Pattern name	Classes that are involved	Additional comment
Inheritance	User, Student, Donor, Guest	User serves as the base class, and Student, Donor, and Guest inherit shared properties and methods, promoting code reusability and extensibility.
Factory Method	UserFactory, User, Student, Donor, Guest	The UserFactory class creates specific user types (Student, Donor, Guest) dynamically based on input, centralizing object creation logic.
Single Responsibility	AuthenticationService, XMLDatabase	Each class has a focused responsibility: AuthenticationService handles login and registration, while XMLDatabase manages data persistence in the XML file.
Dependency Inversion	AuthenticationService, User	AuthenticationService depends on the abstraction User, not concrete implementations (Student, Donor, etc.), ensuring flexibility and scalability.
Association	Club, Student	Student objects can join or leave Club objects through membership associations, reflecting their relationship in the real world.
Composite	Club, Project, Student	Club contains a collection of Project and Student objects, enabling hierarchical data representation and management.
Command Pattern	TransactionService, Donation, Donor	TransactionService acts as the invoker to process commands (e.g., donations), ensuring modularity in handling financial transactions.
Encapsulation	All Classes	Encapsulation is applied to restrict direct access to class members, ensuring data integrity and promoting abstraction through properties.
Observer	Scholarship, Student	Scholarship tracks its applicants and awarded students, allowing dynamic updates when the application's state changes (e.g., a student applies or gets awarded).
Data Mapper	XMLDatabase, User, Club, Project, Scholarship	The XMLDatabase class abstracts XML file operations, mapping data between the XML structure and in-memory objects seamlessly.