**Design Patterns Usage in Admission System**

**Introduction**

The Admission System incorporates several design patterns to manage its functionalities efficiently. This document outlines the classes involved in the system and how they utilize different design patterns.

**Design Patterns Implemented**

1. **Chain of Responsibility (AdmissionHandler)**
   * **Usage:** Enables the handling of admission processes by multiple handlers in a chain.
   * **Classes:** AdmissionHandler, DMCMarksHandler
2. **Strategy (EligibilityCheck)**
   * **Usage:** Provides different eligibility check strategies tailored for each discipline.
   * **Classes:** EligibilityCheck, DMCMarksEligibilityCheck
3. **Factory Method (AdmissionProcessFactory)**
   * **Usage:** Creates specific admission-related objects without exposing creation logic.
   * **Classes:** AdmissionProcessFactory, ComputerScienceAdmissionProcessFactory, OtherDisciplinesAdmissionProcessFactory
4. **Observer (Observer)**
   * **Usage:** Enables notification to multiple observers about admission system events.
   * **Classes:** Observer
5. **Decorator (Quota)**
   * **Usage:** Adds quota functionality to the Admission System without modifying its core code.
   * **Classes:** Quota

**Class-Pattern Relationships**

AdmissionSystem

* **Patterns Used:** Chain of Responsibility, Strategy, Factory Method, Observer, Decorator
* **Explanation:** The Admission System orchestrates various patterns to handle admission processes, eligibility checks, creation of admission-related objects, event notification, and quota management.

Student

* **Pattern Used:** Observer
* **Explanation:** Observes the Admission System, receiving notifications about system events.

Other Classes

* **Patterns Used:** Chain of Responsibility (AdmissionHandler), Strategy (EligibilityCheck), Factory Method (AdmissionProcessFactory), Decorator (Quota)
* **Explanation:** Describe how these classes are involved in the respective design patterns.