

**Singleton Pattern:** Chosen for ensuring that only one instance of a class, such as a centralized hurricane data management system, exists throughout the application. It ensures global access and control over actions like data updates or alerts distribution, crucial for maintaining consistent state across the application. It will be implemented by making the class's constructor private, and providing a public static method to get the instance of the class.

**Observer Pattern:** Selected for managing updates between the system and its users, especially for real-time hurricane tracking and alert notifications. It facilitates communication between the system (subject) and users or other systems (observers) by notifying all registered observers when a change occurs, like a new hurricane detection. Implementation involves creating a subject interface to attach and detach observers, and an observer interface that updates responding to subject's state change.