

Desing Pattern – Use Case Mapping – QuakeAlert

Table of Contents

1. Use Case & Pattern Matrix
2. Justifications for Each Mapping
3. Integration in Architecture
4. Task Matrix

1. Use Case & Pattern Matrix

Use Case

Design Pattern

Receive Earthquake Alert	Observer
Configure Notification	Strategy
View Safety Tips	Command

2. Justification for Each Mapping

1. Receive Earthquake Alert – Observer Pattern

- **Why:** The app must react to external earthquake events and notify users instantly.
- **Benefit:** Observer allows the system to react to state changes (e.g., a new earthquake detected) without tightly coupling components.
- **Example:** The mobile client “observes” the data source and pushes notifications as soon as a new relevant event occurs.

2. Configure Notification – Strategy Pattern

- **Why:** Users may want to change how/when they receive notifications (e.g., only for quakes above a certain magnitude).
- **Benefit:** Strategy enables interchangeable notification behaviour without modifying the core logic.
- **Example:** Each notification preference (e.g., "silent mode", "vibrate", "only daytime") can be implemented as a separate strategy class or function.

3. View Safety Tips – Command Pattern

- **Why:** Each action (showing a new safety tip) can be triggered independently, possibly reused elsewhere.
- **Benefit:** Encapsulates operations as objects, making actions modular and triggerable from buttons, timers, or events.
- **Example:** “Show Tip” buttons execute a Command that fetches and displays safety content.

3. Integration in Architecture

- Observer: Notification Manager + Earthquake Fetcher
- Strategy: Settings UI + Notification Logic
- Command: Safety Info Screen + Action Dispatcher

These patterns help the system stay modular, testable, and extensible particularly important for a real-time, event-driven app like QuakeAlert.

4.Task Matrix

Task

Responsible Member

Pattern selection	Altar
Strategy & Command justification	Talha
Observer mapping	Samet
Integration with architecture section	Yakup
Document review and final formatting	Altar