In the development of this software, I followed common **Java coding conventions** to ensure readability, maintainability, and consistency.

1. Class and Interface Naming

- Class names use **PascalCase** (each word capitalized):
  E.g., Website, User, MonitorService, EmailChannel.
- Interface names also use PascalCase and often describe a capability:
  E.g., Observer, Subject, WebsiteComparisonStrategy.

2. Package Naming

- Package names are all **lowercase** and structured by functionality:
  E.g., com.etatar.websitemonitoring.model, strategy, observer, notification.

3. Method Naming

- Methods use **camelCase** starting with a lowercase letter:
  E.g., checkForUpdates(), notifyObservers(), setComparisonStrategy().

4. Field and Variable Naming

- Fields and variables also use camelCase:
  E.g., lastContent, comparisonStrategy, observers.

5. Constants

- Constants (if any) use **UPPER_CASE** with underscores.
  (e.g., DEFAULT_TIMEOUT, not used in this project yet).

6. Braces and Indentation

- Opening braces go on the same line:

java

```
public void checkForUpdates() {
   // code here
}
```

- Proper indentation (4 spaces) is used consistently.

7. Code Separation and Modularity

- Each class has **a single responsibility**, and logic is divided into appropriate packages:
  - strategy for content comparison
  - observer for observer pattern
  - model for core data objects like Website and User

8. Comments

- Important methods or blocks include short comments where needed.
- JavaDoc could be added for public methods/classes in a full-scale application.