

## The Observer Design Pattern

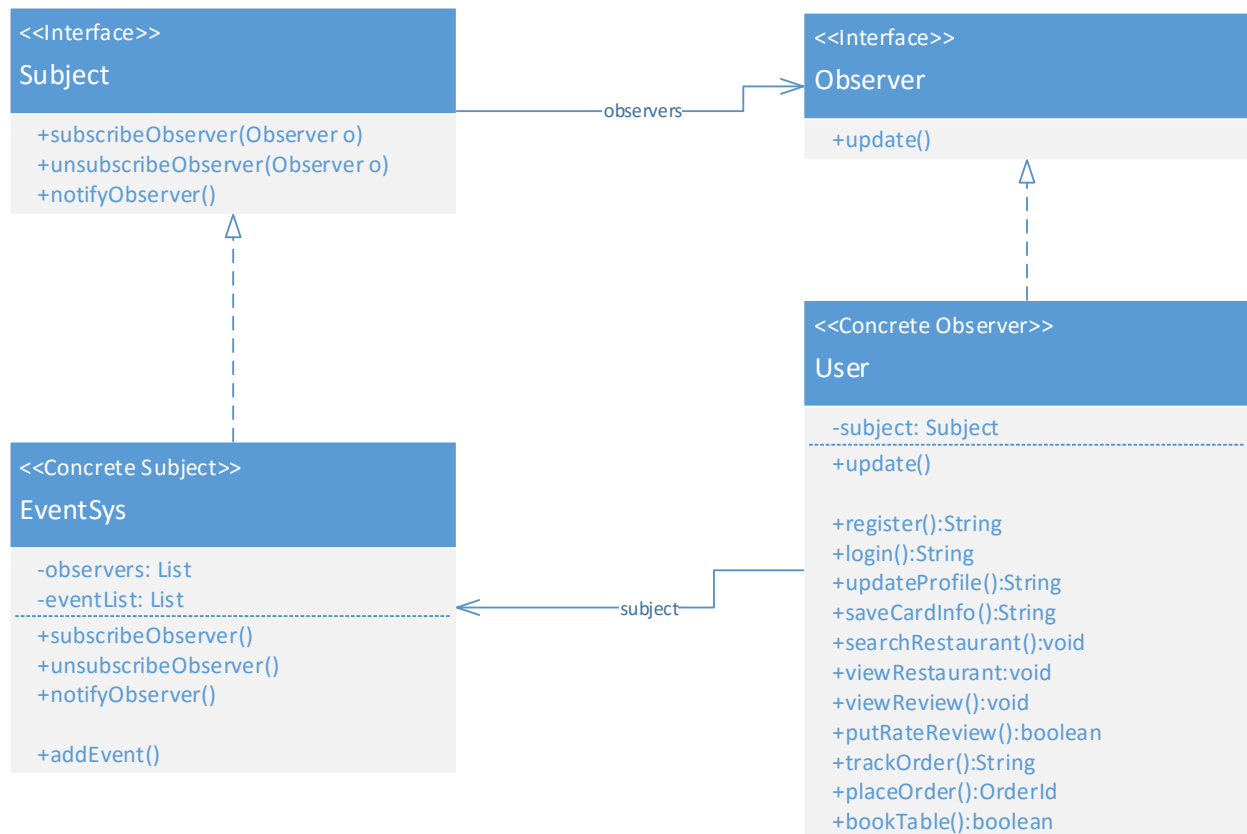
**Object Oriented Goal:** Favoring Composition over Inheritance, Programming to interface and not implementation, maintaining loosely coupled designs between the objects that interact.

**Definition:** The Observer Pattern defines a One-to-Many dependency between objects so that when one object changes state, all of its dependents are notified and updated automatically.

**Implemented on classes:**

- 1) Subject (Interface)
  - a) EventSys (Concrete Subject)
- 2) Observer (Interface)
  - a) User (Concrete Observer)

**Design Snapshot:**



**Design Explanation:**

Whenever a restaurant creates a new event, all the users subscribed with the restaurant must be notified of the events by email or text message. This requirement can be designed using the Observer Design Pattern. Following are the key points of the Observer pattern implemented in our design model:

- The Subject interface acts as a supertype of the concrete subject classes. Users use this interface to subscribe as observers and also to remove themselves from being observers.
- The EventSys class is the concrete subject which implements the Subject interface. Thus, this class has subscribe and unsubscribe methods which are used by observers to subscribe with the particular event notifications. This class also has a notifyObserver() method which is used to notify the Users if any event is added by the restaurant. In addition to the methods implemented from the Subject interface, this class also has its own methods like addEvent(), getEvent(), setEvent() and others which are specific to the class.
- The Observer interface is an interface which needs to be implemented by the Users who want to receive notifications from any event. It has one method update(), which has to be implemented by all the concrete observers, which in our case are the Users.
- The User class implements the Observer interface and thus becomes a concrete observer. The User class needs to implement the update() method. The update method will be called whenever any new event is available to send the notification to the User.
- Every User object subscribes with the concrete subject (EventSys) by calling the subscribe method through the Subject interface. The User can also call unsubscribe if the User no more wants any notifications from the restaurant.