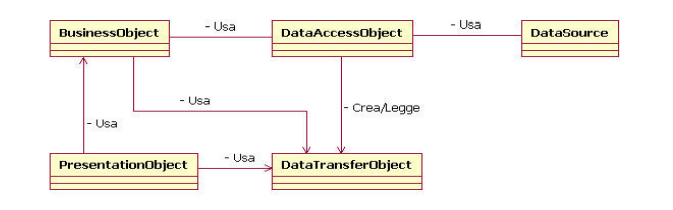
**4 Design Pattern con Class Diagram**

**4.1 Data Access Object Pattern (DAO)**

**Nome e classificazione:** Data Access Object Pattern (DAO) ed è un design pattern architetturale.

**Scopo:** La sua funzione è quella di permettere di astrarre e incapsulare tutti gli accessi al data source.

**Applicabilità:** L’oggetto DAO gestisce la connessione con il data source per estrarre e/o immagazzinare i dati. Gli oggetti DAO implementano i meccanismi di accesso richiesti per lavorare con il data source.

**Class diagram:**

**Partecipanti:** I partecipanti al pattern sono:

* BusinessObject: rappresenta il client del dato. È l’oggetto che richiede l’accesso al data source per ottenere o immagazzinare informazioni.
* DataAccessObject: è l’oggetto principale di questo pattern.

Il DataAccessObject astrae l’implementazione dell’accesso ai dati per il BusinessObject per rendere l’accesso trasparente e indipendente dalla tecnologia utilizzata.

* DataSource: è il data source con la sua implementazione attuale, ovvero il nostro database MySQL.
* TransferObject: rappresenta il Transfer Object usato per trasportare i dati. Il DataAccessObject può usare il Transfer Object per ritornare i dati al client. Il DataAccessObject può anche ricevere dati dal client tramite un Transfer Object per aggiornare il data source.

**Conseguenze:** Il DAO nasconde del tutto ai suoi client i dettagli implementativi di accesso al data source. Poiché l’interfaccia esposta dal DAO non cambia quando il sottostante data source cambia implementazione, questo pattern permette al Data Access Object di adattarsi a differenti schemi implementativi senza che questo abbia alcun effetto sui client o sui componenti del business layer.

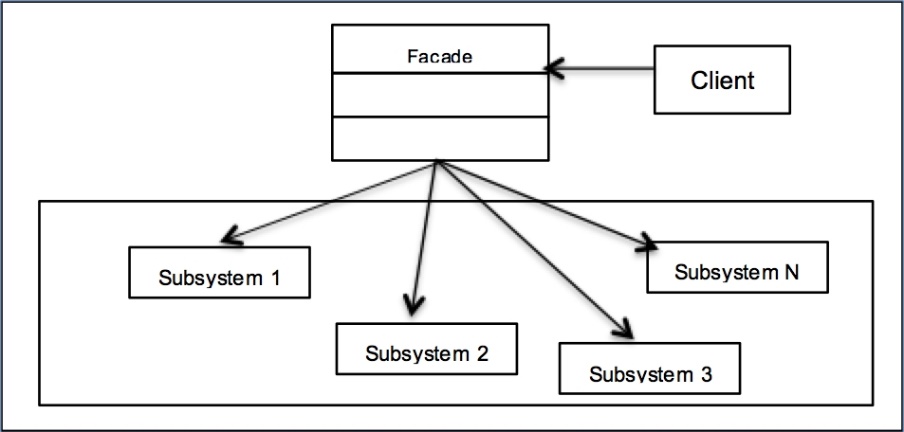
**Utilizzo:** all’interno del nostro prodotto software, verrà utilizzato per la gestione delle connessioni e dei dati del database MySQL.

**4.2 Observer Pattern**

**Nome e classificazione:** Observer Pattern ed è un design pattern comportamentale.

**Scopo:** Questo pattern definisce una relazione tra un oggetto e un insieme di oggetti interessati a essere notificati quando il primo oggetto cambia di stato.

**Applicabilità:** Quando si partiziona un sistema in un insieme di oggetti cooperanti è importante mantenere gli oggetti consistenti tra loro quando cambiano di stato. Spesso l'oggetto che cambia di stato deve essere indipendente dagli oggetti interessati al cambiamento permettendo che il loro numero e identità possano variare dinamicamente. Mantenere riferimenti espliciti agli oggetti interessati produce un design complesso e poco flessibile.

**Class Diagram:**

**Partecipanti:** I partecipanti al pattern sono:

* Observer è l'interfaccia che specifica il tipo di notifiche di eventi di interesse;
* ConcreteObserver implementa Observer e specifica il comportamento dei metodi in risposta agli eventi;
* Subject è la classe che cambia di stato e avverte gli Observers quando tali cambiamenti occorrono. Deve offrire due metodi per rispettivamente aggiungere e rimuovere gli Observer dalla lista degli interessati.

**Conseguenze:** l’Observer pattern, garantisce che quando un oggetto cambia stato, un numero aperto di oggetti dipendenti vengano aggiornati automaticamente, oltre al fatto che deve essere possibile che un oggetto possa notificare un numero aperto di altri oggetti. L’Observer pattern può causare però perdite di memoria, noto come “problema dell’ascoltatore scaduto”, perché nell’implementazione di base si richiede sia la registrazione esplicita, sia la cancellazione esplicita, come nel Dispose Pattern. Questo

perché il soggetto contiene forti riferimenti agli observer, mantenendoli in vita.

**Utilizzo:** all’interno della nostra piattaforma verrà utilizzato nella comunicazione dell’inserimento di una nuova news a tutti gli utenti e nella comunicazione di una segnalazione di una pubblicazione all’autore della stessa.

**5 Glossario**

**MySQL:** MySQL è un database relazionale (RDBMS) multithread open source, sviluppato nel 1996 da una società di consulenza svedese, la TcX, che aveva bisogno di un database veloce e che richiedeva poche risorse, pur dovendo gestire notevoli quantità di dati. La diffusione di MySQL è dovuta principalmente alla sua natura open source e gratuita, oltre alle sue doti di velocità e flessibilità.

**DBMS:** Un DBMS è sostanzialmente uno strato software che si frappone fra l’utente ed i dati veri e propri. Grazie a questo strato intermedio l’utente e le applicazioni non accedono ai dati così come sono memorizzati effettivamente, cioè alla loro rappresentazione fisica, ma ne vedono solamente una rappresentazione logica. Ciò permette un elevato grado di indipendenza fra le applicazioni e la memorizzazione fisica dei dati.

**RDBMS:** Il termine relational database management system (RDBMS) (sistema per la gestione di basi di dati relazionali) indica un database management system basato sul modello relazionale. Un database relazionale è costituito da tabelle, ognuna delle quali è composta da righe identificate da un codice univoco denominato chiave. Le tabelle che compongono il database non sono del tutto indipendenti tra loro ma relazionate da legami logici.

**Java:** Java è un linguaggio di programmazione ad alto livello, orientato agli oggetti e a tipizzazione statica, che si appoggia sull'omonima piattaforma software di esecuzione, specificamente progettato per essere il più possibile indipendente dalla piattaforma hardware di esecuzione.

**HTML:** HTML (HyperText Markup Language) è un linguaggio di markup. Oggi è utilizzato principalmente per il disaccoppiamento della struttura logica di una pagina web e la sua rappresentazione, gestita tramite gli stili CSS per adattarsi alle nuove esigenze di comunicazione e pubblicazione all'interno di Internet.

**Javascript:** In informatica JavaScript è un linguaggio di scripting orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione Web lato client per la creazione, in siti web e applicazioni web, di effetti dinamici interattivi tramite funzioni di script invocate da eventi innescati a loro volta in vari modi dall'utente sulla pagina web in uso.

**CSS:** Il CSS, in informatica, è un linguaggio usato per definire la formattazione di documenti HTML, XHTML e XML, ad esempio i siti web e relative pagine web.

**CamelCase**: la notazione a cammello, o in inglese CamelCase, è la pratica nata durante gli anni 70 di

scrivere parole composte o frasi unendo tutte le parole tra loro, ma lasciando le loro iniziali maiuscole. Si può distinguere in un lowerCamelCase, in cui la prima lettera della prima parola viene lasciata minuscola, o in UpperCamelCase, in cui la prima lettera della prima parola è maiuscola.

**Bootstrap:** Bootstrap è una raccolta di strumenti liberi per la creazione di siti e applicazioni per il Web. Essa contiene modelli di progettazione basati su HTML e CSS, sia per la tipografia, che per le varie componenti dell'interfaccia, come moduli, pulsanti e navigazione, così come alcune estensioni opzionali di JavaScript.