



Object Design Document

Riferimento	NC10_SDD_ver.1.0
Versione	1.0
Data	09/01/2026
Destinatario	Prof. Carmine Gravino
Presentato da	NC10 Team

Revision History

Data	Versione	Descrizione	Autori
01/12/2025	0.1	Prima Stesura	CG
20/12/2025	0.2	Revisione	CG
09/01/2026	1.0	Revisione Finale	CG, AP, BR ,RO

Team members

Nome	Ruolo nel progetto	Acronimo	Informazioni di contatto
Antonio Plumitallo	Team Member	AP	a.plumitallo@studenti.unisa.it
Riccardo Odierna	Team Member	RO	r.odierna1@studenti.unisa.it
Biagio Radino	Team Member	BR	b.radino@studenti.unisa.it
Christian Antonio Genovese	Team Member	CG	c.genovese18@studenti.unisa.it

Pattern 1 - Proxy per immagini catalogo

Obiettivo

L'adozione del pattern **Proxy** ha l'obiettivo di migliorare le prestazioni del sistema e ridurre il carico di rete, evitando il caricamento immediato di tutte le immagini associate ai libri presenti nel catalogo.

In particolare, l'immagine reale viene recuperata solo quando effettivamente necessaria, ad esempio nel momento in cui l'utente accede alla scheda di dettaglio di un libro o quando l'immagine entra nell'area visibile dell'interfaccia (viewport).

Questa scelta consente di:

- ridurre i tempi di risposta nelle operazioni di browsing del catalogo;
- limitare l'utilizzo di banda e risorse di sistema;
- migliorare l'esperienza utente nelle fasi di navigazione iniziale.

Implementazione nel sistema

Nel sottosistema Gestione Catalogo viene definita un'interfaccia comune, ad esempio **ImmagineLibro**, che espone operazioni per l'accesso all'immagine, come **mostra()**.

L'interfaccia è implementata da due classi:

- **ImmagineLibroReale**, che incapsula l'accesso allo storage fisico delle immagini (filesystem o database);
- **ImmagineLibroProxy**, che mantiene esclusivamente i metadati necessari (ad esempio identificativo del libro e path dell'immagine) e un riferimento inizialmente nullo all'oggetto reale.

Alla prima invocazione di un metodo dell'interfaccia, il proxy provvede a istanziare e caricare l'oggetto **ImmagineLibroReale**, delegando successivamente a quest'ultimo tutte le chiamate future.

In questo modo, il caricamento dell'immagine viene posticipato fino al momento in cui risulta realmente necessario.

Pattern 2 - Facade per i sottosistemi applicativi

Obiettivo

Il pattern **Facade** viene adottato per fornire un punto di accesso unificato e semplificato alle funzionalità offerte dai principali sottosistemi applicativi, quali:

- Gestione Catalogo;
- Gestione Carrello;
- Gestione Ordini;
- Gestione Recensioni;
- Registrazione e Autenticazione.

L'obiettivo principale è nascondere la complessità interna dell'architettura (controller, service, DAO) al livello di presentazione, riducendo l'accoppiamento e migliorando la manutenibilità del sistema.

Implementazione nel sistema

Viene introdotta una o più classi Facade, ad esempio **ShopFacade**, che espongono metodi ad alto livello come:

- `cercaLibri(...)`
- `aggiungiAlCarrello(...)`
- `checkout(...)`
- `inserisciRecensione(...)`

Internamente, la Facade coordina le chiamate ai service dei singoli sottosistemi (ad esempio **GestioneCatalogoService**, **CarrelloService**, **GestioneOrdiniService**, **GestioneRecensioniService**, **GestioneAutenticazioneService**), occupandosi della gestione delle sequenze di operazioni necessarie al completamento delle richieste.

In questo modo, i controller del livello di presentazione interagiscono con un'interfaccia uniforme e semplificata, senza dipendere direttamente dai dettagli implementativi dei sottosistemi.