

Università degli studi di Napoli Parthenope

Dipartimento di Scienze e Tecnologie



System Design Document **ThetaSport**

Studenti:

Attilio Di Vicino
Lorenzo Pergamo
Mario Vista

Docenti:

Paola Barra
Mariacarla Staffa

Matricole:

0124002347
0124002278
0124002266

Anno Accademico 2022/2023

Indice

1	Introduzione	3
1.1	Scopo del sistema	3
1.1.1	Utente	3
1.1.2	Admin	3
1.2	Obiettivi di progettazione	3
1.3	Riferimenti	4
1.4	Panoramica	5
2	Sistema corrente	7
3	Sistema proposto	8
3.1	Panoramica	8
3.2	Decomposizione del sistema	8
3.3	Hardware/Software mapping	10
3.4	Gestione dei dati persistenti	11
3.5	Controllo accessi e sicurezza	13
3.6	Decisioni sul flusso di controllo globale	14
3.7	Condizioni limite	15
4	Servizi del sottosistema	17

Elenco delle figure

1.1	Diagrammi delle classi	6
3.1	HW/SW Mapping	11
3.2	EE/R Database	12
3.3	Modello Relazionale Database	13
3.4	Chain of Responsibility	14
4.1	Sottosistema	19

Capitolo 1

Introduzione

1.1 Scopo del sistema

Il sistema ThetaSport è un'applicazione di e-commerce progettata per offrire una piattaforma online completa per la vendita di prodotti sportivi. Lo scopo del sistema è fornire un'esperienza di acquisto comoda e personalizzata per gli utenti, consentendo loro di cercare, visualizzare e acquistare prodotti sportivi di alta qualità. Le funzionalità offerte dalla seguente piattaforma sono:

1.1.1 Utente

- Inserire i prodotti nel carrello della spesa
- Effettuare il pagamento. Il pagamento può avvenire secondo le modalità: contanti, carta di credito o bancomat.

1.1.2 Admin

- Inserire o cambiare un nuovo prodotto con le sue informazioni
- In base agli acquisti effettuati dagli utenti, proporre offerte specifiche
- In base ad una categoria visualizzare il numero di prodotti venduti

1.2 Obiettivi di progettazione

Il sistema si propone di raggiungere i seguenti obiettivi:

1. **Usabilità:** L'interfaccia utente dovrà essere intuitiva, facile da navigare e fornire una piacevole esperienza d'uso per gli utenti. La disposizione dei contenuti, le icone e le azioni dovranno essere ben organizzate per facilitare la ricerca dei prodotti e il completamento dell'acquisto.
2. **Prestazioni:** Il sistema dovrà avere prestazioni ottimali per consentire tempi di risposta rapidi durante la navigazione, la ricerca dei prodotti e il check-out. Le pagine dovranno caricarsi velocemente per ridurre i tempi di attesa degli utenti.
3. **Sicurezza:** La sicurezza delle informazioni degli utenti e delle transazioni finanziarie sarà una priorità, per prevenire accessi non autorizzati e proteggere i dati sensibili.
4. **Scalabilità:** Il sistema dovrà essere progettato per gestire un carico di utenti crescente senza perdere prestazioni o affidabilità. La scalabilità permetterà al sistema di adattarsi all'aumento del traffico e del numero di utenti senza subire interruzioni o rallentamenti significativi.

1.3 Riferimenti

Nel processo di progettazione dell'applicazione ThetaSport, ci siamo basati su diversi riferimenti e pratiche consolidate per garantire un design solido e ben strutturato. I principali riferimenti utilizzati includono:

1. **Architettura MVC** (Model-View-Controller): È stata adottata l'architettura MVC per separare chiaramente la logica dell'applicazione in tre componenti distinti - Model, View e Controller. Questo ci ha permesso di creare un'applicazione più modulare e facile da mantenere.
2. **Pattern Architetturale DAO** (Data Access Object): Il pattern DAO è stato utilizzato per separare la logica di accesso ai dati dal resto dell'applicazione. Questo ci ha consentito di accedere al database in modo efficiente e scalabile.
3. **Algoritmo TF-IDF**: Nell'applicazione è stato implementato l'algoritmo TF-IDF per l'analisi del comportamento degli utenti e la generazione di offerte personalizzate basate sugli acquisti storici.
4. **Principi S.O.L.I.D.**: Vengono inoltre sfruttati i principi S.O.L.I.D., ossia un insieme di regole di progettazione del software, per garantire che il codice sia leggibile, flessibile, privo di bug e di alta qua-

lità. L'adesione a questi principi ha reso il progetto più gestibile e collaborativo.

5. **Design Pattern:** Durante la fase di progettazione, sono stati applicati alcuni design pattern per risolvere problemi comuni e migliorare la struttura del codice. I design pattern utilizzati sono **Singleton**, **Abstract Factory**, **Visitor**, **Strategy** e **Chain of Responsibility**.

1.4 Panoramica

L'analisi del progetto ha portato alla seguente suddivisione in classi:



Figura 1.1: Diagrammi delle classi

Capitolo 2

Sistema corrente

Il sistema corrente di ThetaSport è stato sviluppato come un prototipo in Java utilizzando le tecnologie Servlet e Jakarta, con Tomcat come server web. Si tratta di una web application progettata e implementata da zero per soddisfare le esigenze specifiche del progetto.

Le tecnologie utilizzate includono:

1. **Java:** Il linguaggio di programmazione principale utilizzato per sviluppare l'applicazione. È stato richiesto l'utilizzo di Java per la sua portabilità, sicurezza e robustezza.
2. **Servlet:** Le Servlet sono stati utilizzate per gestire le richieste e le risposte HTTP. Sono parte della piattaforma Java EE e consentono di sviluppare applicazioni web dinamiche.
3. **Tomcat:** Tomcat è un server web e servlet container open-source sviluppato dalla Apache Software Foundation. È stato utilizzato per eseguire l'applicazione ThetaSport e gestire le richieste HTTP.

Dato che si tratta di un prototipo, l'applicazione potrebbe essere stata sviluppata per mostrare idee, funzionalità e logiche senza implementare completamente tutte le caratteristiche finali previste nel sistema completo. Di conseguenza, il prototipo iniziale potrebbe richiedere ulteriori miglioramenti per diventare una versione finale completa ed efficiente del sistema. La nostra web application è stata costruita da zero, il che significa che l'intero sistema è stato creato senza basarsi su nessuna applicazione preesistente. Questo approccio è noto come 'Greenfield Engineering', anche se durante la fase di progettazione abbiamo preso in considerazione alcune delle principali piattaforme di negozi online, come Amazon e Sportega.

Capitolo 3

Sistema proposto

3.1 Panoramica

Per il sistema proposto, come già detto in precedenza, si è scelto di adottare le architetture **MVC** (Model-View-Controller) e **DAO** (Data Access Object) per la progettazione della piattaforma. Queste architetture sono state selezionate in base alle esigenze del progetto, in particolare concentrandoci su due aspetti cruciali: sicurezza e scalabilità, insieme a un miglioramento nell'esperienza utente.

L'architettura **MVC** ci permetterà di separare in modo chiaro la logica dell'applicazione e la rappresentazione dell'interfaccia utente, consentendo una maggiore flessibilità nello sviluppo e una migliore manutenibilità del codice. Con questa architettura potremo gestire in modo efficace l'interazione tra i dati (**Model**), la presentazione dei dati all'utente (**View**) e il controllo delle operazioni (**Controller**).

L'architettura **DAO**, invece, ci consentirà di gestire in modo efficiente l'accesso ai dati nel database. Questo pattern ci aiuterà a separare la logica dell'applicazione dalla logica di accesso ai dati, semplificando le operazioni di lettura e scrittura dei dati e migliorando la sicurezza dell'applicazione.

3.2 Decomposizione del sistema

La decomposizione del sistema è il procedimento mediante il quale si frammenta un sistema complesso in parti più piccole e gestibili. Tale processo è finalizzato a semplificare la fase di progettazione e sviluppo. Questa suddivisione aiuta anche a favorire il riuso del codice e a migliorare la manutenibilità complessiva del sistema.

In questo caso la decomposizione può essere eseguita considerando diverse aree funzionali e moduli. Alcuni esempi di possibili componenti del sistema potrebbero includere:

- **Modulo di Autenticazione e Autorizzazione:** Questo modulo gestisce la registrazione degli utenti, l'autenticazione e l'autorizzazione per l'accesso alle diverse funzionalità dell'applicazione. Includerà funzionalità di gestione dei ruoli e dei privilegi degli utenti.
- **Modulo di Gestione del Catalogo Prodotti:** Questo modulo si occuperà della gestione dei prodotti disponibili sul sito. Sarà responsabile dell'aggiunta, modifica ed eliminazione dei prodotti, nonché della loro visualizzazione all'interno del catalogo.
- **Modulo di Gestione del Carrello:** Questo modulo si concentrerà sulla gestione del carrello degli acquisti. Consentirà agli utenti di aggiungere e rimuovere prodotti dal carrello, calcolare il totale degli acquisti e gestire le operazioni di check-out.
- **Modulo di Gestione degli Ordini:** Questo modulo si occuperà della gestione degli ordini effettuati dagli utenti. Registrerà i dettagli degli ordini, lo stato della spedizione e le informazioni di pagamento.
- **Modulo di Gestione delle Offerte Personalizzate:** Questo modulo sarà responsabile della generazione e dell'invio di offerte personalizzate agli utenti in base ai loro storico acquisti e preferenze.
- **Modulo di Gestione per la creazione di prodotti:** Questo modulo gestirà la creazione dei prodotti e li renderà acquistabili sul catalogo.
- **Modulo di Gestione dei pagamenti:** Questo modulo gestirà le modalità di pagamento per semplificare le operazioni legate a un metodo di pagamento specifico.

Ogni componente del sistema sarà progettato per essere autonomo e riusabile, in modo da poter essere utilizzato anche in altre parti dell'applicazione. La decomposizione del sistema permette di affrontare il processo di progettazione e sviluppo in modo più modulare e organizzato, migliorando la manutenibilità e scalabilità complessiva di ThetaSport.

3.3 Hardware/Software mapping

L'hardware/software mapping, anche noto come mappatura hardware/software, è il processo di collegamento e associazione delle componenti software del sistema con l'hardware fisico su cui verranno eseguite. Questo processo è cruciale durante la progettazione del sistema, poiché determina come il software sarà distribuito e eseguito sui diversi componenti hardware.

Nel caso di ThetaSport, la mappatura hardware/software potrebbe essere eseguita come segue:

1. **Web Server:** Il sistema utilizza un server web, come Apache, per ospitare l'applicazione web di ThetaSport. Il software dell'applicazione, inclusi i file HTML, CSS, JavaScript e le servlet Java, verranno eseguiti su questo server web.
2. **Application Server:** Un application server, potrebbe essere utilizzato per eseguire le servlet Java e gestire le richieste degli utenti. Questo server applicativo è responsabile di eseguire la logica di business dell'applicazione e di interagire con il database.
3. **Database Server:** Il sistema potrebbe utilizzare un database server, come MySQL o PostgreSQL, per archiviare e gestire i dati dell'applicazione. Il software del database gestirà le operazioni di scrittura e lettura dei dati dal database.
4. **Client-Side:** Sui dispositivi client, come i browser web, verrà eseguito il software client dell'applicazione. Questo includerà il rendering delle pagine web e l'esecuzione di script JavaScript per gestire l'interazione dell'utente con l'applicazione.

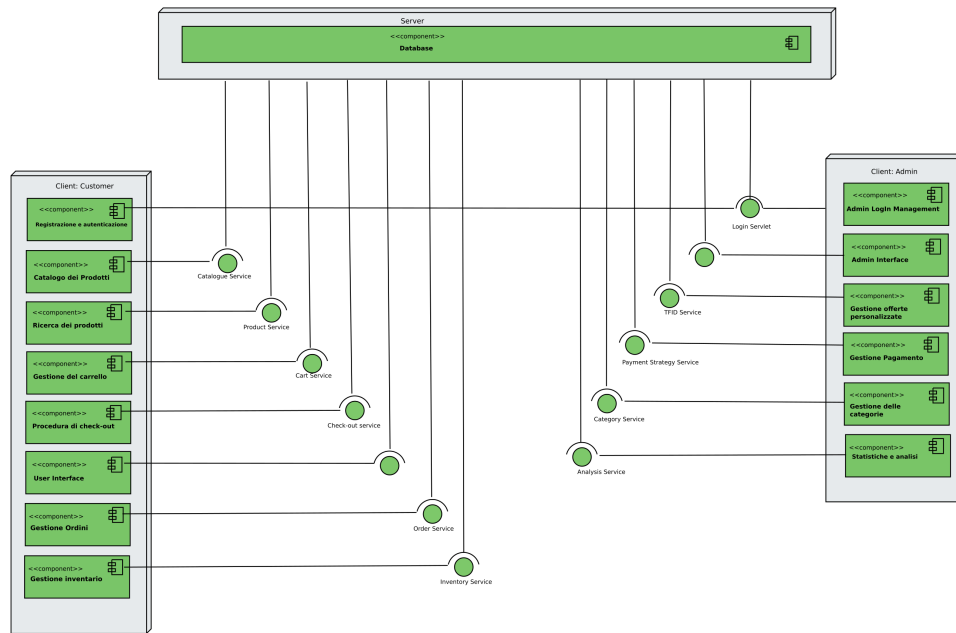


Figura 3.1: HW/SW Mapping

3.4 Gestione dei dati persistenti

La gestione dei dati persistenti è un aspetto fondamentale in un sistema come ThetaSport, che è un'applicazione web e-commerce. Per gestire i dati in modo persistente, possono essere utilizzate varie soluzioni, ma una delle opzioni comuni è l'utilizzo di un database relazionale. Di seguito è fornita una panoramica della gestione dei dati persistenti:

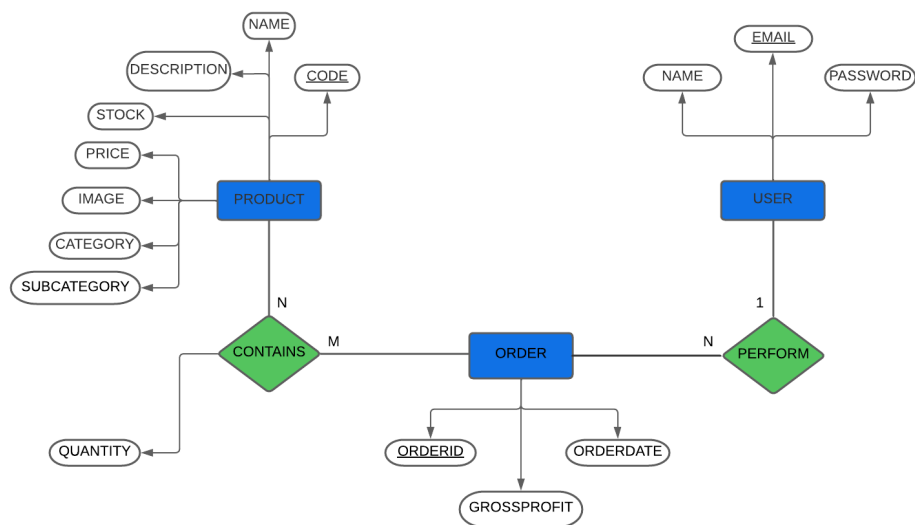


Figura 3.2: EE/R Database

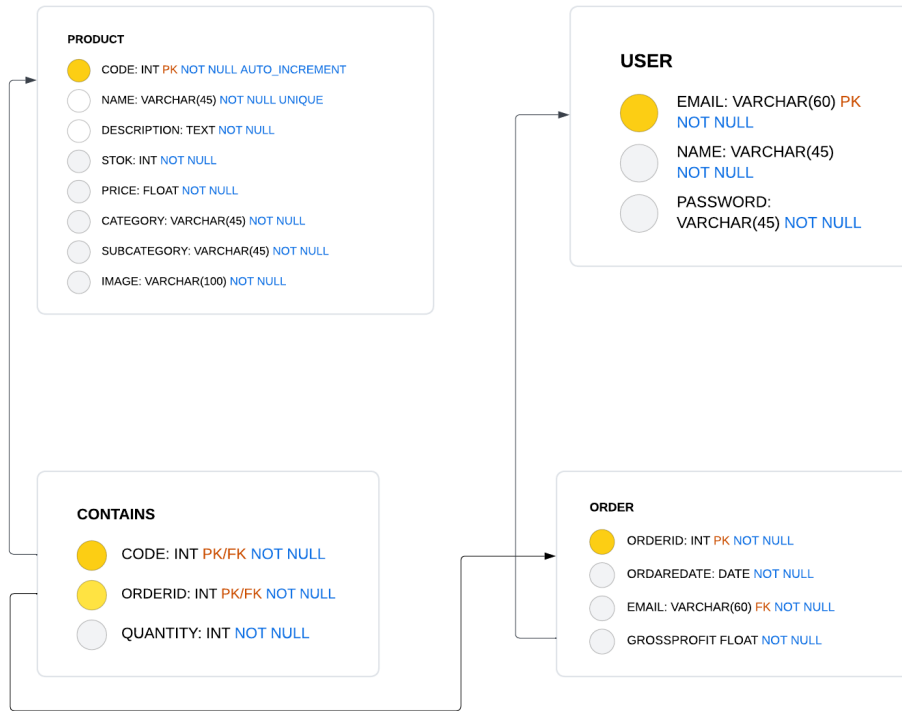


Figura 3.3: Modello Relazionale Database

3.5 Controllo accessi e sicurezza

Il controllo degli accessi e sicurezza garantiscono che solo gli utenti autorizzati possano usufruire di determinate funzionalità e dati sensibili. Di seguito, viene descritto come questi aspetti possono essere gestiti utilizzando sessioni, tramite il pattern Chain of Responsibility, con la verifica di password, email e ruolo degli utenti:

1. Gestione delle Sessioni:

- Quando un utente accede al sistema, una nuova sessione viene creata e associata a quell'utente. La sessione memorizza le informazioni di autenticazione e autorizzazione dell'utente.
- Durante la sessione, l'utente può accedere alle diverse funzionalità dell'applicazione senza dover inserire di nuovo le credenziali.

- Per garantire la sicurezza, la sessione ha una durata limitata e si annulla automaticamente dopo 30 minuti. Questo aiuta a prevenire accessi non autorizzati da parte di utenti che hanno lasciato il sistema aperto ma inattivo per un periodo prolungato.

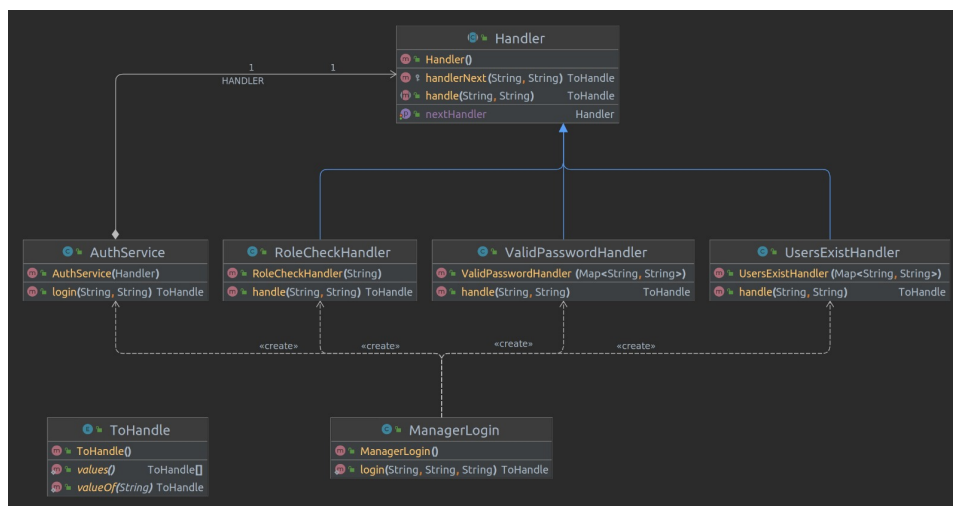


Figura 3.4: Chain of Responsibility

3.6 Decisioni sul flusso di controllo globale

Le decisioni sul flusso di controllo globale riguardano il modo in cui l'applicazione gestisce e indirizza il flusso delle operazioni e delle funzionalità in

modo coerente e organizzato. Queste decisioni sono fondamentali per garantire che il sistema funzioni correttamente e in modo efficiente. Di seguito, alcune decisioni comuni sul flusso di controllo globale per l'applicazione ThetaSport:

1. **Gestione degli Errori:** Assicurarsi di fornire messaggi di errore significativi per guidare l'utente in caso di errori.
2. **Autenticazione e Autorizzazione:** sistema di autenticazione per verificare le credenziali degli utenti e garantire che solo gli utenti autorizzati possano accedere alle funzionalità protette dell'applicazione.
3. **Gestione delle Sessioni:** sistema di gestione delle sessioni per tenere traccia dell'autenticazione degli utenti e delle loro attività all'interno dell'applicazione.

3.7 Condizioni limite

La gestione degli errori e delle eccezioni è una parte molto importante per garantire un'esperienza utente senza intoppi e per proteggere l'integrità e la sicurezza dei dati. Mediante la gestione adeguata degli errori e delle eccezioni, il sistema può gestire diversi casi limite e fornire un feedback significativo agli utenti riguardo agli errori commessi. Di seguito, sono forniti alcuni esempi di come il sistema ThetaSport affronta tali situazioni:

1. **Registrazione di un nuovo utente:** Se un utente tenta di registrarsi con un'email già esistente nel sistema, il sistema cattura l'errore e mostra un messaggio di errore appropriato all'utente, informandolo che l'email è già in uso e chiedendo di utilizzare un'email diversa.
2. **Controllo accessi:** Se un utente tenta di accedere a una funzionalità per la quale non ha le autorizzazioni, il sistema cattura l'errore e reindirizza l'utente a una pagina di errore o a una pagina di accesso negato, fornendo un messaggio chiaro che indica la mancanza di autorizzazioni.
3. **Procedura di check-out:** Se un utente tenta di effettuare un acquisto senza prodotti nel carrello, il sistema cattura l'errore e mostra un messaggio di errore all'utente, avvisandolo che il carrello è vuoto e che deve aggiungere prodotti prima di procedere con il check-out.

4. **Gestione dell'inventario:** Se un amministratore tenta di inserire una quantità negativa di prodotti nel sistema, il sistema cattura l'errore e rifiuta l'operazione, mostrando un messaggio di errore che indica l'impossibilità di inserire una quantità negativa.

Inoltre, il sistema può generare eccezioni e gestirle in modo appropriato per situazioni impreviste o critiche, come errori di connessione al database o altri problemi tecnici. La gestione corretta delle eccezioni consente di fornire informazioni dettagliate per il debugging e per risolvere i problemi in modo tempestivo.

Capitolo 4

Servizi del sottosistema

Di seguito sono elencati alcuni dei servizi del sottosistema di ThetaSport:

1. **Registrazione e Autenticazione Utenti:** Fornisce la possibilità agli utenti di registrarsi al sistema, creando un account con email e password, e di autenticarsi per accedere alle funzionalità riservate.
2. **Catalogo dei Prodotti:** Mostra il catalogo completo dei prodotti disponibili nell' e-commerce, con dettagli, immagini e prezzi.
3. **Ricerca dei Prodotti:** Permette agli utenti di effettuare ricerche per categoria, nome del prodotto o altre caratteristiche per trovare i prodotti desiderati.
4. **Gestione del Carrello:** Gli utenti possono visualizzare il contenuto del carrello, aggiornare le quantità dei prodotti o svuotarlo completamente.
5. **Procedura di Check-out:** Consente agli utenti di procedere con il pagamento e la conferma dell'ordine.
6. **Gestione degli Ordini:** Permette agli utenti di visualizzare lo stato dei loro ordini e le informazioni relative alle spedizioni.
7. **Gestione dell'Inventario:** Permette agli amministratori di aggiungere, modificare o rimuovere prodotti dal catalogo.
8. **Generazione di Offerte Personalizzate:** Gli amministratori possono creare offerte speciali e promozioni personalizzate per gli utenti sulla base dei loro storico acquisti.

9. **Gestione del Pagamento:** Fornisce opzioni di pagamento sicuro e facilita il processo di pagamento dell'ordine.
10. **Gestione delle Categorie:** Consente agli amministratori di gestire le categorie dei prodotti e organizzare il catalogo.
11. **Statistiche e Analisi:** Fornisce report e statistiche sull'andamento delle vendite, preferenze degli utenti, prodotti più venduti, ecc.

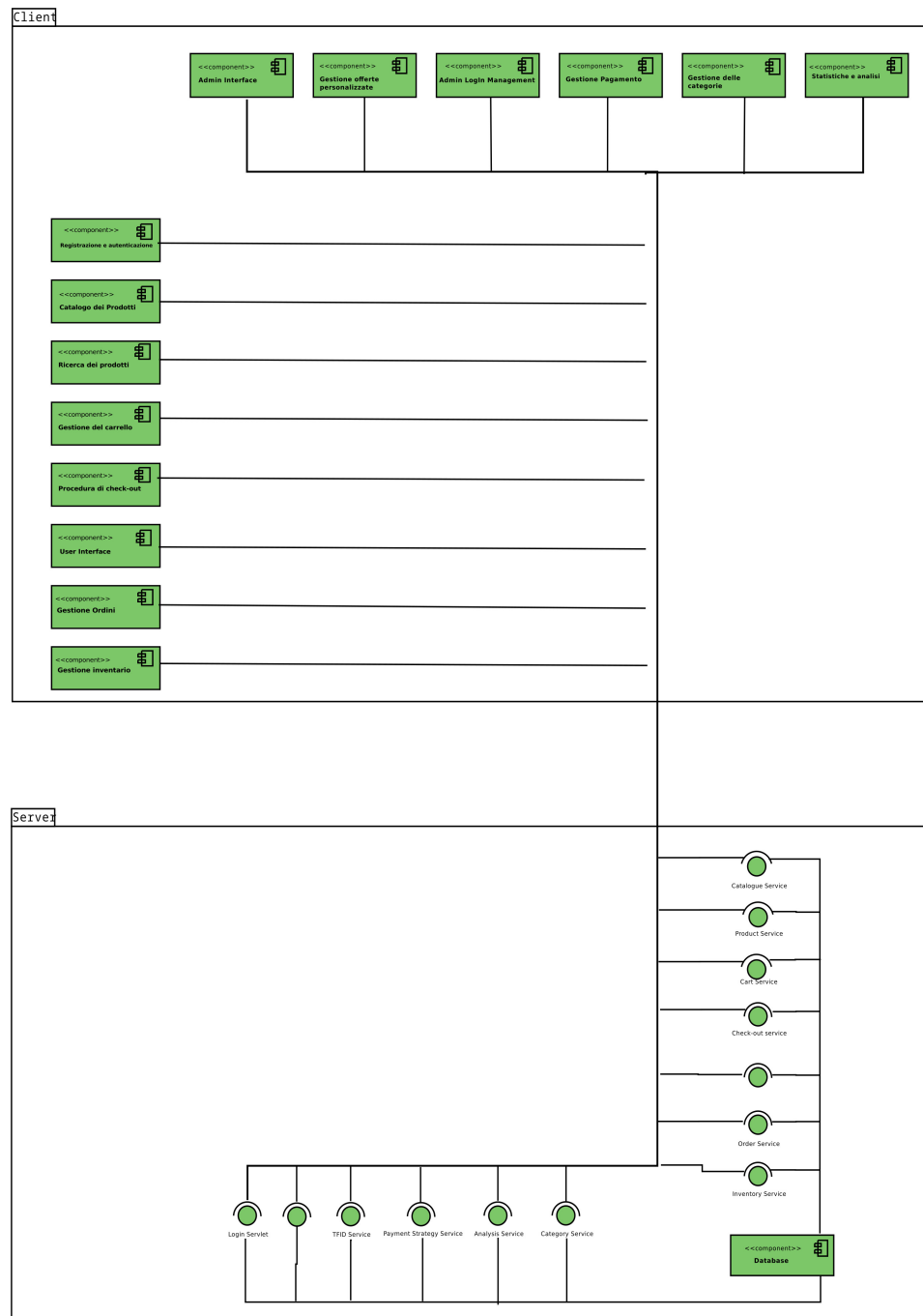


Figura 4.1: Sottosistema