



Laurea Triennale in informatica - Università di Salerno  
Corso di Ingegneria del Software - Prof. C. Gravino

# System Design Document



Riferimento	NC24_SDD_ver1.0
Versione	1.0
Data	24/11/2025
Destinatario	Prof. Carmine Gravino
Presentato da	Gruppo NC24
Approvato da	



## Revision History

Data	Versione	Descrizione	Autori
24/11/2025	0.1	Prima stesura	Mario Branca Simone Sammartano Paolo Visconti
25/11/2025	0.2	Stesura del secondo capitolo	Tutto il team
27/11/2025	0.3	Stesura del terzo capitolo	Tutto il team
30/11/2025	0.4	Stesura del quarto capitolo	Paolo Visconti Simone Sammartano
09/12/2025	0.5	Revisione e accorgimenti	Mario Branca Gabriele De Luca
18/12/2025	0.6	Correzione di errori e sistemazione documento	Gabriele De Luca Paolo Visconti
09/01/2026	0.7	Correzione di alcune inconsistenze	Mario Branca Simone Sammartano
14/01/2026	0.8	Rimozione di ridondanze	Gabriele De Luca
19/01/2026	0.9	Correzione di inconsistenze	Tutto il team
07/02/2026	1.0	Sistemazione per la consegna finale	Tutto il team

## Team Members

Nome	Ruolo	Acronimo	Informazioni di contatto
Mario Branca	Team Member	M.B.	m.branca2@studenti.unisa.it
Gabriele De Luca	Team Member	G.D.L.	g.deluca65@studenti.unisa.it
Simone Sammartano	Team Member	S.S.	s.sammartano1@studenti.unisa.it
Paolo Visconti	Team Member	P.V.	p.visconti4@studenti.unisa.it



# Sommario

<b>Revision History</b> .....	<b>1</b>
<b>Team Members</b> .....	<b>1</b>
<b>Sommario</b> .....	<b>2</b>
<b>1. Introduzione</b> .....	<b>3</b>
1.1 Scopo del Sistema.....	3
1.2 Obiettivi di design.....	4
Design Goals.....	5
Trade-Offs.....	8
1.3 Definizioni, acronimi, e abbreviazioni.....	8
1.4 Riferimenti.....	9
1.5 Organizzazione del documento.....	9
<b>2. Architettura del sistema corrente</b> .....	<b>9</b>
<b>3. Architettura del sistema proposto</b> .....	<b>10</b>
3.1 Panoramica sulla sezione.....	10
3.2 Scomposizione in sottosistemi.....	11
3.3 Mapping hardware/software.....	14
3.4 Gestione dei dati persistenti.....	15
3.5 Controllo degli accessi e Sicurezza.....	17
3.6 Controllo globale del software.....	19
3.7 Condizioni limite.....	19
Avvio del Sistema.....	19
Spegnimento del Sistema.....	20
Fallimento del Sistema.....	21
Errore di accesso ai dati persistenti.....	21
<b>4. Servizi dei sottosistemi</b> .....	<b>22</b>
Sottosistema Gestione Account.....	22
Sottosistema Catalogo.....	23
Sottosistema Carrello.....	23
Sottosistema Acquisto.....	24
Sottosistema Recensioni.....	24
Sottosistema Gestione Amministrativa.....	25
<b>5. Glossario</b> .....	<b>26</b>



# 1. Introduzione

---

## 1.1 Scopo del Sistema

Il sistema Readify ha come obiettivo principale quello di modernizzare l'esperienza di acquisto di libri tramite un portale e-commerce dedicato. L'applicazione mira a offrire agli utenti un ambiente semplice e intuitivo attraverso cui consultare un catalogo digitale di opere editoriali, effettuare ricerche tramite filtri avanzati, gestire un carrello personale e completare un processo di acquisto guidato.

Il sistema svolge un duplice compito: da un lato semplifica l'incontro tra domanda e offerta nel mercato dei libri online, dall'altro fornisce agli amministratori strumenti strutturati per la gestione del catalogo, degli ordini e degli utenti, garantendo controllo e tracciabilità.

In particolare, il sistema permette a un utente registrato di:

- Navigare il catalogo dei libri tramite categorie, filtri e barra di ricerca.
- Aggiungere, modificare o rimuovere prodotti dal carrello personale.
- Effettuare l'acquisto tramite un processo guidato che include inserimento indirizzo di spedizione, metodi di pagamento e riepilogo ordine.
- Accedere alla propria area personale per visualizzare ordini, dettagli e stato delle spedizioni.

Per quanto riguarda la parte amministrativa, il sistema consente a un operatore con privilegi di amministratore di:

- Gestire l'inserimento, la modifica e la rimozione dei libri nel catalogo.
- Monitorare gli ordini ricevuti e aggiornare lo stato.
- Gestire i profili degli utenti registrati.

L'obiettivo complessivo è quello di fornire una piattaforma affidabile e scalabile che permetta ai clienti di vivere un'esperienza di acquisto fluida e ai gestori del sistema di mantenere un controllo efficiente e centralizzato sulle attività di vendita.

Il progetto Readify, tramite le funzionalità sopra indicate, si propone di offrire un servizio moderno e accessibile, aumentando la soddisfazione dell'utente finale e garantendo al tempo stesso un modello di gestione chiaro e professionale.



## 1.2 Obiettivi di design

In questa sezione vengono definiti i Design Goals del sistema, ovvero le qualità architetturali e di servizio che esso deve perseguire in modo prioritario. Tali obiettivi rappresentano un ponte tra i requisiti non funzionali e le decisioni progettuali: esplicarli permette di giustificare le scelte di architettura, di gestire i trade-off e di mantenere coerenza durante l'evoluzione del sistema.

I Design Goals sono stati individuati a partire dai requisiti non funzionali raccolti per il progetto e seguono le linee guida del libro "Object-Oriented Software Engineering" di Bernd Bruegge. Per facilitare l'analisi, essi sono stati raggruppati nelle seguenti categorie:

- **Performance:** includono i requisiti di spazio e velocità imposti sul sistema.
- **Dependability:** determinano quanto sforzo deve essere speso per minimizzare i fallimenti del sistema (crash, fallo di sicurezza) e le loro conseguenze.
- **Maintenance:** determinano quanto è difficile modificare il sistema dopo il suo rilascio.
- **End User:** includono requisiti desiderabili dal punto di vista dell'utente e coperti dai criteri di Performance e Dependability.

Per garantire tracciabilità e confrontabilità, ogni Design Goal viene descritto tramite i seguenti campi:

- **Rank**, specifica un valore di priorità compreso tra 1 e 16 (1 massima e 16 minima).
- **ID Design Goal**, un identificatore univoco e un nome esplicativo.
- **Descrizione**, una descrizione del design goal.
- **Categoria**, ovvero la categoria di appartenenza del design goal.
- **RNF di origine**, ovvero il requisito non funzionale che lo ha generato.



## Design Goals

Rank	ID Design Goal	Descrizione	Categoria	RNF di origine
2	DG_1 Tempi di risposta	Il sistema deve garantire un tempo di risposta non superiore a 5 secondi per le principali funzionalità (navigazione del catalogo, gestione del carrello e checkout).	Performance	RNF_PERF_3
3	DG_2 Spazio di archiviazione	Il sistema deve poter gestire tutti i dati persistenti da memorizzare, occupando uno spazio di archiviazione non superiore ai 10 gigabyte.	Performance	RNF_PERF_2
4	DG_3 Affidabilità delle operazioni	Il sistema deve garantire che le operazioni principali (registrazione, login, aggiunta/rimozione al carrello, acquisto) vengano completate con successo oppure correttamente annullate, evitando stati parziali o incoerenti.	Dependability	RNF_A1
12	DG_4 Fallimento del sistema	Il sistema deve sapersi comportare in situazioni di fallimento (es. server o database irraggiungibili), notificando l'utente tramite messaggi chiari, così da favorire il ripristino del servizio in tempi contenuti.	Dependability	RNF_A_3



6	DG_5 Gestione permessi	Il sistema deve garantire una netta divisione tra le varie categorie di utenti (visitatore, utente registrato, amministratore), limitando le funzionalità accessibili a ciascun ruolo in modo da preservare l'integrità del catalogo, degli ordini e dei dati utente.	Dependability	RNF_A_5
13	DG_6 Disponibilità del sistema	Il sistema deve garantire un'elevata disponibilità del servizio di e-commerce, con un limite indicativo di 8 ore all'anno di downtime complessivo (pianificato e non pianificato).	Dependability	RNF_P_3
1	DG_7 Sicurezza dei dati	Il sistema deve garantire la massima sicurezza dei dati conservati, utilizzando protocolli di comunicazione sicuri, tecniche adeguate per la conservazione dei dati persistenti e assicurando che i dati siano visualizzati solo dagli utenti autorizzati.	Dependability	RNF_A_2, RNF_LE_1
7	DG_8 Manutenibilità	Il sistema deve essere facilmente manutenibile al fine di ridurre l'impatto delle modifiche e favorire il riuso del codice.	Maintenance	RNF_S_2
5	DG_9 Estensibilità	Il sistema deve prestarsi bene all'aggiunta di nuove funzionalità tipiche di un e-commerce di libri, senza richiedere interventi	Maintenance	RNF_S_2



		invasivi sull'architettura esistente.		
8	DG_10 Facilità d'uso	Il sistema deve risultare facilmente comprensibile ed utilizzabile anche da utenti poco esperti: la ricerca dei libri, la gestione del carrello e la procedura di acquisto devono essere lineari e guidare l'utente evitando passaggi superflui.	End user	RNF_U_1, RNF_U_4
9	DG_11 Interfaccia intuitiva	L'interfaccia utente della piattaforma deve permettere di eseguire le azioni principali (ricerca, filtraggio per categoria/autore, aggiunta al carrello, checkout, accesso all'area personale) in modo chiaro e semplice, rendendo evidente la funzione di ogni elemento visuale (pulsanti, icone, link).	End user	RNF_U_2
10	DG_12 Feedback esplicito	Ogni azione rilevante della piattaforma deve fornire un chiaro feedback all'utente tramite messaggi di conferma o di errore, evitando situazioni in cui lo stato del sistema non è evidente.	End user	RNF_U_3
11	DG_13 Deployment semplificato	Il sistema deve essere facilmente implementabile anche su una singola macchina, esponendo l'applicazione web sulla rete e consentendo l'accesso a tutte le risorse	End User	RNF_PA_1



		necessarie con una configurazione documentata e ripetibile.		
14	DG_14  Interfaccia con le utility di sistema per l'amministratore	Il sistema deve interagire correttamente con le utility del sistema operativo sul quale viene eseguito il server, così da permettere all'amministratore di controllare lo stato dell'applicazione e intervenire in caso di problemi.	End User	RNF_OP_1

## Trade-Offs

Trade-off	Descrizione
<b>Tempi di rilascio vs Funzionalità</b>	Per garantire maggiori funzionalità e completezza dell'operatività del sistema si potrebbe ricorrere a un tempo di rilascio maggiore.
<b>Affidabilità delle operazioni vs Tempi di risposta</b>	Per garantire maggiore affidabilità delle operazioni si potrebbe ricorrere a tempi di risposta più lenti.

## 1.3 Definizioni, acronimi, e abbreviazioni

Vengono riportati di seguito alcune definizioni presenti nel documento corrente:

- **Sottosistema:** un sottoinsieme dei servizi del dominio applicativo, formato da servizi legati da una relazione funzionale.
- **Design Goal:** le qualità sulle quali il sistema deve essere focalizzato.
- **Dati Persistenti:** dati che sopravvivono all'esecuzione del programma che li ha creati e che dunque vengono salvati.



- **Mapping Hardware/Software:** studio della connessione tra parti fisiche e logiche di cui si compongono il sistema.
- **SDD:** System Design Document
- **RAD:** Requirements Analysis Document

## 1.4 Riferimenti

Di seguito sono riportati alcuni riferimenti a documenti utili durante la lettura:

- [Readify - SOW](#)
- [Readify - RAD](#)
- [Readify - Object Design](#)
- [Readify\\_TP.docx](#)
- [Readify\\_TCS.docx](#)
- [Readify\\_TIR.docx](#)

## 1.5 Organizzazione del documento

Il presente documento, riguardante la fase di System Design, è stato suddiviso nelle seguenti sezioni:

**Introduzione:** Viene descritto in forma generica lo scopo del sistema e gli obiettivi di design da raggiungere nella sua realizzazione.

**Architettura software corrente:** Viene descritto lo stato attuale dell'architettura del software (se già presente).

**Architettura software proposta:** Viene descritto come il sistema sarà definito e partizionato in sottosistemi, il loro mapping Hardware/Software, la gestione dei dati persistenti. Viene poi presentata la struttura dei singoli sottosistemi e cosa accade nei casi di condizioni limite del sistema.

**Glossario:** Lista della terminologia specifica usata nel documento con relativa spiegazione.

## 2. Architettura del sistema corrente

---

Il sistema di e-commerce di libri è già stato sviluppato in un altro corso di programmazione adottando un'architettura basata sul pattern MVC (Model–View–Controller).



In tale soluzione:

- il Model comprende le entità di dominio (Libro, Utente, Carrello, Ordine, IndirizzoSpedizione, ecc.) mappate sul database;
- i Controller sono realizzati tramite Servlet, che gestiscono le richieste HTTP e coordinano l'esecuzione dei casi d'uso;
- le View sono costituite da pagine JSP, responsabili della presentazione dei dati all'utente.

L'architettura è di fatto monolitica: presentazione, logica applicativa e accesso ai dati risiedono nello stesso contesto applicativo. Parte della logica di business risulta distribuita tra Controller e JSP e l'accesso ai dati non è completamente encapsulato in un livello dedicato.

Questa impostazione può risultare adatta a un progetto di programmazione, ma meno coerente con gli obiettivi di manutenibilità, estensibilità e chiara separazione delle responsabilità proposti in questo progetto, motivo per cui si propone una riorganizzazione in architettura a tre livelli.

## 3. Architettura del sistema proposto

---

### 3.1 Panoramica sulla sezione

L'architettura proposta riorganizza il sistema esistente in una architettura a tre livelli (Three-Tier), mantenendo il modello Client–Server ma rendendo esplicita la separazione tra:

- Livello di Presentazione
- Livello di Logica Applicativa
- Livello di Gestione dei Dati

L'obiettivo è quello di superare i limiti dell'attuale implementazione MVC monolitica, in cui presentazione e logica sono più strettamente intrecciate, introducendo una struttura che:

- migliora la manutenibilità (modifiche circoscritte a un livello);
- facilita la gestione dei ruoli e dei controlli di accesso (incapsulati nella logica applicativa);
- permette una scalabilità più flessibile (eventuale distribuzione futura dei livelli su nodi distinti);



- supporta meglio l'evoluzione del sistema (es. introduzione di nuove interfacce utente o servizi esterni).

In questa nuova architettura:

- ❖ il **Livello di Presentazione** rimane basato su pagine JSP e accessibili solo tramite Servlet, che si occupano di mostrare il catalogo, il carrello, il dettaglio ordine, ecc.;
- ❖ il **Livello di Logica Applicativa** è costituito da componenti di tipo Control che implementano i casi d'uso e centralizzano i controlli di accesso e le regole di business;
- ❖ il **Livello di Gestione dei Dati** comprende le entità persistenti e i componenti responsabili dell'accesso al database (DAO/Repository per Libro, Utente, Carrello, Ordine, ecc.), in modo da isolare le operazioni di persistenza dal resto del sistema.

Per l'implementazione, il front-end non utilizza framework, ma solo JavaScript, HTML5 e CSS3 per creare un'interfaccia utente moderna e responsiva.

Il back-end è sviluppato con Java e Javascript, con l'aggiunta di Ajax per il caricamento di dati in maniera dinamica riguardo funzionalità del sistema.

La gestione dei dati è affidata a JDBC Driver, che si interfaccia con un database SQL ospitato su MySQL Workbench ed hostato in rete tramite Microsoft Azure per garantire scalabilità, sicurezza, ma soprattutto semplicità di connessione e installazione.

## 3.2 Scomposizione in sottosistemi

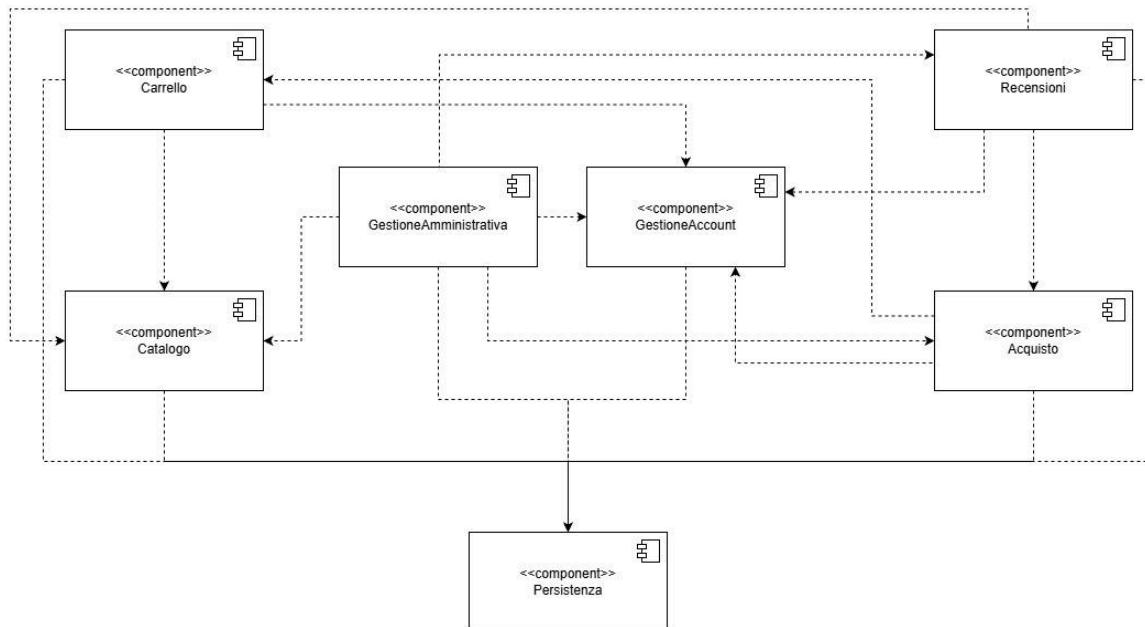
I sottosistemi individuati sono:

- **Gestione Account:** si occupa della registrazione degli utenti, dell'autenticazione (Login/Logout), della gestione dei ruoli (utente registrato / amministratore), della visualizzazione dell'Area Utente e della modifica dei dati dell'account, inclusi gli indirizzi di spedizione associati all'utente.
- **Catalogo:** si occupa della consultazione dei libri disponibili dal punto di vista dell'utente. È responsabile della visualizzazione dei prodotti con le relative informazioni (titolo, autore, ISBN, descrizione, prezzo, disponibilità), nonché delle funzionalità di ricerca, filtraggio e ordinamento.
- **Carrello:** si occupa delle operazioni relative alla gestione del carrello, consentendo l'aggiunta, la modifica e la rimozione dei prodotti selezionati dall'utente e il calcolo

del totale corrente dell'acquisto. Questo sottosistema in particolare gestisce due modalità distinte: un carrello temporaneo, mantenuto a livello di sessione per i visitatori non autenticati, e un carrello persistente, associato agli utenti registrati e memorizzato tramite il livello di persistenza.

- **Acquisto:** si occupa della creazione e gestione degli ordini, a partire dal contenuto del carrello. Gestisce la selezione dell'indirizzo di spedizione, il riepilogo dell'ordine, la conferma e il successivo pagamento, nonché la consultazione dello storico ordini da parte dell'utente.
- **Gestione Amministrativa:** si occupa delle operazioni di amministrazione della piattaforma da parte degli utenti con ruolo di amministratore, come la gestione del catalogo (inserimento, modifica e rimozione dei libri), la gestione degli utenti e la gestione degli ordini e delle recensioni (consultazione e moderazione).
- **Recensioni:** si occupa delle operazioni relative al rilascio, all'eventuale modifica e alla visualizzazione dei feedback degli utenti sui libri acquistati, nonché del calcolo e della presentazione delle valutazioni aggregate per ciascun libro.
- **Persistenza:** si occupa della memorizzazione fisica e della consistenza dei dati, rappresentando il database relazionale che conserva in modo permanente tutte le informazioni del sistema.

Sono mostrate in seguito le dipendenze tramite un Component Diagram UML:



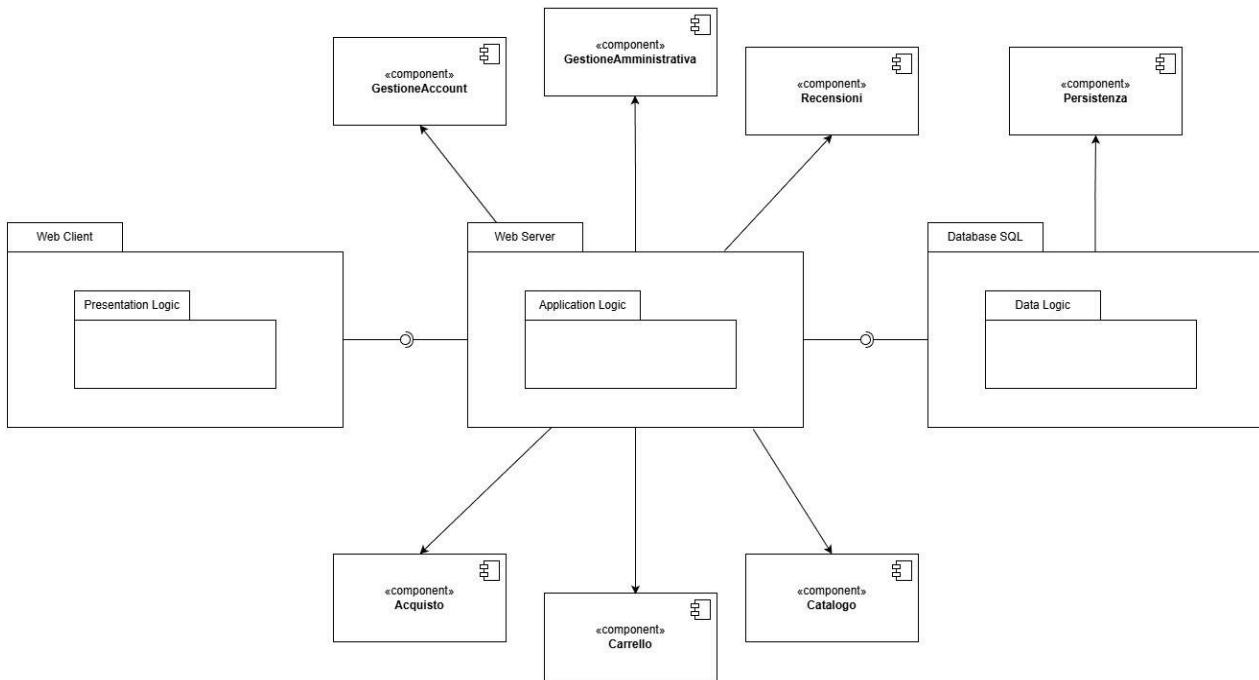


Alcuni sottosistemi di Readify sono supportati da componenti **COTS** (Commercial Off-The-Shelf). In particolare:

- La **persistenza dei dati** è affidata a un **DBMS relazionale MySQL**, messo a disposizione dall'ateneo, e hostato in rete tramite **Microsoft Azure**, messo a disposizione gratuitamente da Microsoft, utilizzati come repository dei dati del sistema.
- L'esecuzione dell'applicazione web è affidata a un **container Servlet/JSP** (Apache Tomcat), che fornisce il supporto COTS per la gestione delle richieste HTTP e il rendering delle pagine JSP.

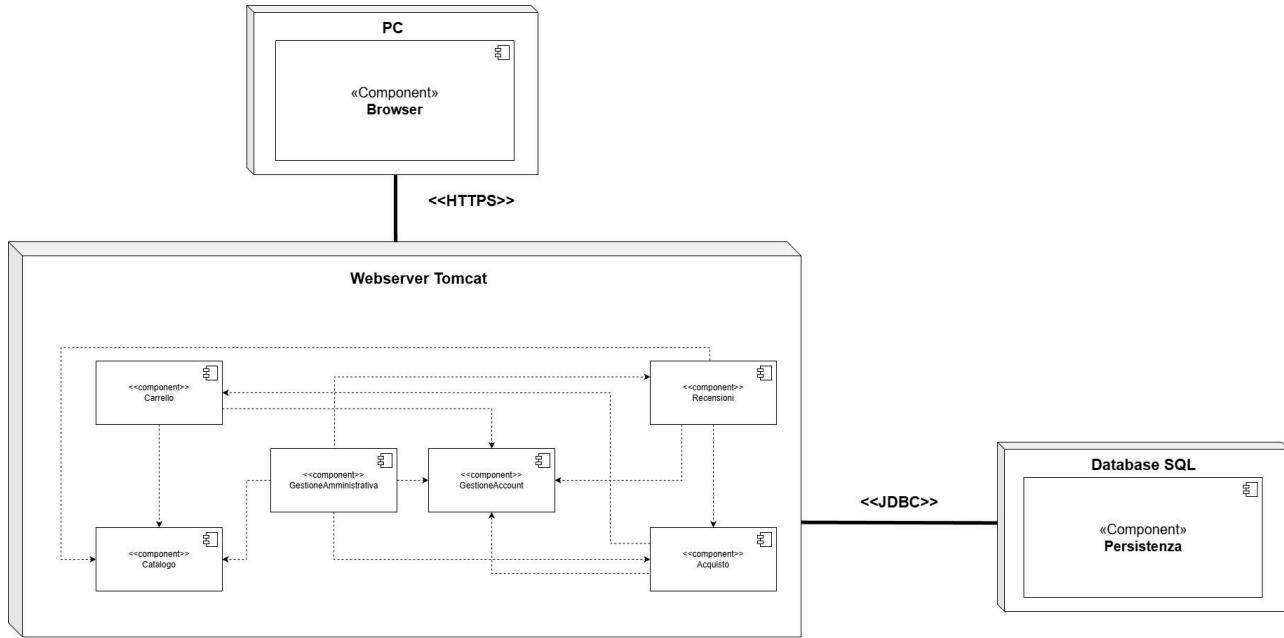
Ciascun sottosistema applicativo è realizzato utilizzando le seguenti componenti principali:

- **GUI (Graphic User Interface):**  
Comprende le pagine **JSP** e le risorse HTML/CSS/JavaScript utilizzate per costruire l'interfaccia web del sistema. Le JSP sono collocate all'interno della cartella **WEB-INF** e non sono direttamente accessibili dall'esterno, ma vengono raggiunte tramite i Controller. Questo livello corrisponde al **livello di presentazione** dell'architettura a tre livelli.
- **Controller:**  
È costituito dalle **Servlet** che ricevono le richieste HTTP, effettuano i controlli di accesso (distinzione tra utente non autenticato, utente registrato e amministratore), orchestrano l'esecuzione dei casi d'uso e selezionano le viste JSP appropriate da mostrare all'utente. I Controller fanno parte del **livello di logica applicativa**.
- **Service:**  
Raccoglie le classi di **logica di business** responsabili dell'implementazione delle operazioni applicative (gestione catalogo, gestione carrello, creazione ordini, gestione account, gestione recensioni, funzionalità amministrative). I Controller delegano a questi servizi l'esecuzione delle operazioni di dominio. Anche i Service appartengono al **livello di logica applicativa**.
- **DAO (Data Access Object):**  
Comprende le classi responsabili dell'accesso ai dati persistenti mediante l'utilizzo diretto di tecnologie di persistenza a basso livello (ad esempio JDBC), incapsulando le operazioni di lettura e scrittura sul database. In tal modo, il livello di logica applicativa risulta disaccoppiato dai dettagli implementativi della persistenza. Queste componenti costituiscono il livello di gestione dei dati del sistema.



### 3.3 Mapping hardware/software

Il sistema Readify si basa su una piattaforma hardware costituita da un server che risponde alle richieste effettuate dagli utenti da qualsiasi macchina dotata di browser e connessione a Internet. Il sistema adotta un'architettura three-tier, ma, poiché si tratta di una web application non distribuita, tutti e tre i livelli risiedono sullo stesso nodo server, che ospita il container Servlet/JSP, i componenti applicativi e il database relazionale hostato su Microsoft Azure. Di seguito è riportato un UML deployment diagram che descrive il mapping hardware/software.



### 3.4 Gestione dei dati persistenti

Per la gestione dei dati persistenti del sistema Readify è stato scelto l'utilizzo di un database relazionale, ritenuto maggiormente adatto alle esigenze di un e-commerce di libri.

Questa scelta permette di gestire in modo efficace l'accesso concorrente ai dati (catalogo libri, utenti registrati, carrelli, ordini, indirizzi di spedizione, recensioni) e di garantire la consistenza tramite le funzionalità offerte dal DBMS.

La decisione è coerente con i Design Goals definiti in precedenza, in particolare quelli relativi a sicurezza dei dati, affidabilità e manutenibilità, e offre i seguenti vantaggi:

- **Vincoli di integrità sui dati:** Il DBMS consente di definire vincoli di chiave primaria, chiave esterna, unicità e altri vincoli di integrità referenziale, assicurando che la base di dati rimanga coerente con il dominio applicativo. Ogni modifica (inserimento, aggiornamento, cancellazione) viene automaticamente verificata rispetto a tali vincoli.
- **Protezione dei dati:** L'uso di un DBMS relazionale permette di controllare l'accesso alla base dati tramite utenti e privilegi, limitando le operazioni consentite a ciascun account. In questo modo è possibile separare i permessi di amministrazione del

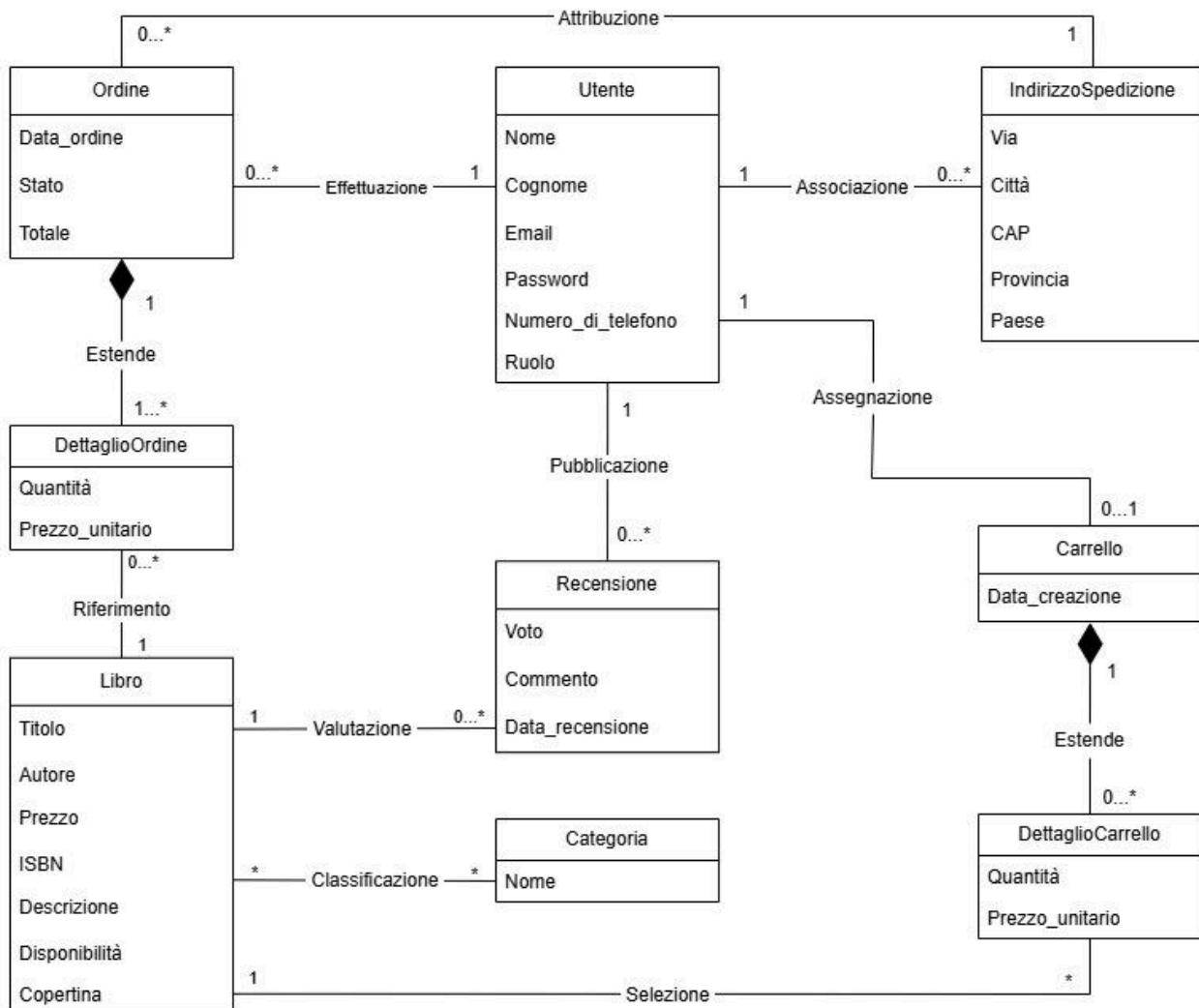


database da quelli usati dall'applicazione, contribuendo al rispetto dei design goals di sicurezza.

- **Affidabilità e recupero da guasti:** Il DBMS mette a disposizione meccanismi di backup e ripristino che consentono di recuperare lo stato della base dati in caso di malfunzionamenti software o hardware, riducendo il rischio di perdita di informazioni critiche (ordini, pagamenti registrati, dati utente).
- **Atomicità delle transazioni:** Le operazioni critiche, come la conferma di un ordine o l'aggiornamento del carrello, vengono eseguite all'interno di transazioni atomiche: o tutte le operazioni coinvolte vanno a buon fine, oppure nessuna modifica viene applicata. Questo è fondamentale per mantenere uno stato coerente tra il carrello dell'utente, gli ordini registrati e le informazioni di pagamento.

Dal punto di vista implementativo, l'accesso ai dati persistenti è realizzato tramite il pattern DAO (Data Access Object), che utilizza JDBC per interagire con il database relazionale, incapsulando le operazioni di lettura e scrittura e isolando la logica applicativa dai dettagli della persistenza.

Per agevolare lo sviluppo, il testing e l'installazione della piattaforma, è stato scelto di hostare il database in rete tramite Microsoft Azure, che mette a disposizione un'istanza condivisa del database, accessibile a tutti i membri del progetto.



### 3.5 Controllo degli accessi e Sicurezza

Attori / Oggetti	Amministratore	Utente
Gestione Account	<ul style="list-style-type: none"> <li>○ Registrazione</li> <li>○ Login</li> <li>○ Visualizzazione dati account</li> <li>○ Modifica dati account</li> </ul>	<ul style="list-style-type: none"> <li>○ Registrazione</li> <li>○ Login</li> <li>○ Visualizzazione dati account</li> <li>○ Modifica dati account</li> </ul>



	<ul style="list-style-type: none"><li>○ Visualizzazione storico ordini</li><li>○ Logout</li></ul>	<ul style="list-style-type: none"><li>○ Visualizzazione storico ordini</li><li>○ Logout</li></ul>
Catalogo	<ul style="list-style-type: none"><li>○ Visualizzazione elenco libri</li><li>○ Visualizzazione scheda prodotto</li><li>○ Ricerca / filtro libri</li></ul>	<ul style="list-style-type: none"><li>○ Visualizzazione elenco libri</li><li>○ Visualizzazione scheda prodotto</li><li>○ Ricerca / filtro libri</li></ul>
Carrello	<ul style="list-style-type: none"><li>○ Visualizzazione carrello</li><li>○ Aggiungi prodotto</li><li>○ Rimuovi prodotto</li><li>○ Modifica quantità</li></ul>	<ul style="list-style-type: none"><li>○ Visualizzazione carrello</li><li>○ Aggiungi prodotto</li><li>○ Rimuovi prodotto</li><li>○ Modifica quantità</li></ul>
Acquisto	<ul style="list-style-type: none"><li>○ Avvio procedura di acquisto</li><li>○ Selezione / aggiunta indirizzo di spedizione</li><li>○ Inserimento dati di pagamento</li><li>○ Riepilogo ordine</li></ul>	<ul style="list-style-type: none"><li>○ Avvio procedura di acquisto</li><li>○ Selezione / aggiunta indirizzo di spedizione</li><li>○ Inserimento dati di pagamento</li><li>○ Riepilogo ordine</li></ul>
Recensioni	<ul style="list-style-type: none"><li>○ Visualizzazione recensioni</li><li>○ Inserimento recensione</li><li>○ Modifica / cancellazione di una recensione</li></ul>	<ul style="list-style-type: none"><li>○ Visualizzazione recensioni</li><li>○ Inserimento recensione</li><li>○ Modifica / cancellazione di una recensione</li></ul>
Gestione Amministrativa	<ul style="list-style-type: none"><li>○ Visualizzazione lista utenti</li><li>○ Modifica / rimozione utenti</li><li>○ Aggiungi libro</li><li>○ Rimuovi libro</li><li>○ Modifica dettagli libro</li><li>○ Visualizzazione storico ordini piattaforma</li><li>○ Visualizzazione e moderazione recensioni</li></ul>	



### 3.6 Controllo globale del software

Il sistema Readify è una web application interattiva: ogni funzionalità viene attivata a seguito di un'azione dell'utente sull'interfaccia (click, invio di form, selezione di opzioni).

Ogni azione genera una richiesta HTTP che viene gestita dal Controller (Servlet) corrispondente, il quale instrada il flusso verso il sottosistema applicativo responsabile e, quando necessario, utilizza i componenti Service, che a loro volta richiamano i DAO, per l'accesso ai dati persistenti tramite JDBC.

Per questi motivi, il controllo globale del software in Readify segue un **meccanismo** di tipo **event-driven**, tipico delle applicazioni web: non esiste un ciclo di controllo centralizzato, ma il flusso di esecuzione è determinato dalle richieste generate dinamicamente dalle interazioni dell'utente.

### 3.7 Condizioni limite

Nel presente paragrafo verranno presentate le boundary conditions inerenti all'avvio del sistema, spegnimento del sistema, fallimento del sistema ed errore di accesso ai dati persistenti.

#### Avvio del Sistema

Identificativo		Data	02/12/2025
UCBC_1 – Avvio del Sistema		Versione	0.1
		Autore	Simone Sammartano
Descrizione		Lo Use Case descrive cosa succede all'avvio del sistema	
Amministratore		Admin	
Attori secondari		NA	
Entry Condition		L'Admin accede al Server	
Exit condition On success		Il sistema si avvia correttamente	
Exit condition On failure		Il sistema non viene avviato	
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Admin:	Esegue il comando che avvia il sistema.	
2	Sistema:	Verifica l'integrità dei dati persistenti e, nel caso, rende disponibili i suoi servizi e le sue funzionalità agli utenti.	
I Flusso di Eventi Alternativo: I Dati Persistenti sono danneggiati			
2.a1	Sistema:	Notifica l'Admin di problemi ai dati persistenti e non effettua	



		I'avvio.
2.a2	Admin:	Corregge i dati persistenti.
2.a3	Admin:	Esegue il passaggio 1.

## Spegnimento del Sistema

Identificativo	UCBC_2 – Spegnimento del Sistema	Data Versione Autore	02/12/2025 0.1 Mario Branca
	<b>Descrizione</b> Lo Use Case permette lo spegnimento del sistema		
	<b>Amministratore</b> Admin		
<b>Attori secondari</b>		<b>NA</b>	
<b>Entry Condition</b>		L'Admin accede al Server AND Il Sistema è stato precedentemente avviato AND Il Sistema non è stato ancora spento	
<b>Exit condition</b> On success		Il sistema viene spento correttamente	
<b>Exit condition</b> On failure		Il sistema non viene spento	

### FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO

1	Admin:	Invia un segnale di spegnimento al Sistema.
2	Sistema:	Controlla che non ci siano connessioni ancora aperte da o verso l'esterno e, se non ci sono, termina l'esecuzione del sistema.

### I Flusso di Eventi Alternativo: Ci sono connessioni ancora aperte

2.a1	Sistema:	Notifica all'Admin che ci sono ancora connessioni aperte verso l'esterno.
2.a2	Sistema:	Attende una quantità di tempo per rispondere a eventuali richieste dall'esterno, non generando nuove connessioni se non allo scopo di rispondere a richieste già in corso.
2.a3	Sistema:	Controlla che non ci siano connessioni ancora aperte da o verso l'esterno e, se non ci sono, termina l'esecuzione del sistema.
2.a4	Sistema:	Notifica l'Amministratore dell'avvenuto spegnimento del sistema.

### Il Flusso di eventi alternativo: I Dati Persistenti sono danneggiati

2.a3.a1	Sistema:	Notifica l'Admin di problemi ai dati persistenti e non effettua lo spegnimento.
2.a3.a2	Admin:	Corregge i dati persistenti.



2.a3.a3	Admin:	Esegue il passaggio 1.
---------	--------	------------------------

## Fallimento del Sistema

Identificativo	UCBC_3 – Fallimento del Sistema	Data	02/12/2025
		Versione	0.1
		Autore	Gabriele de Luca
Descrizione	Lo Use Case definisce il comportamento del Sistema in caso di fallimento.		
Amministratore	Admin		
Attori secondari	NA		
Entry Condition	Il Sistema viene terminato all'improvviso		
Exit condition On success	Il Sistema viene riavviato correttamente		
Exit condition On failure	Il Sistema non viene riavviato		
<b>FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO</b>			
1	Admin:	Include UCBC_1	

## Errore di accesso ai dati persistenti

Identificativo	UCBC_4 – Errore di accesso ai dati persistenti	Data	02/12/2025
		Versione	0.1
		Autore	Paolo Visconti
Descrizione	Lo Use Case descrive il comportamento del sistema qualora fosse impossibile accedere ai dati persistenti o questi risultassero corrotti.		
Amministratore	Admin		
Attori secondari	NA		
Entry Condition	Il Sistema non può accedere ai dati persistenti OR I dati persistenti risultano corrotti		
Exit condition On success	Il Sistema riprende il normale funzionamento		
Exit condition On failure	Il Sistema non riprende il normale funzionamento		
<b>FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO</b>			
1	Sistema:	Notifica l'Admin dell'impossibilità di accedere ai dati	



		persistenti.
2	<b>Sistema:</b>	Smette di processare eventuali richieste e risponde a tutte le richieste con un messaggio di errore.
3	<b>Admin:</b>	<i>Include UCBC_2</i>
4	<b>Admin:</b>	Ripristina l'accessibilità o l'integrità dei dati persistenti.
5	<b>Admin:</b>	<i>Include UCBC_1</i>

## 4. Servizi dei sottosistemi

---

In questa sezione vengono descritti tutti i servizi dei vari sottosistemi elencati finora.

### Sottosistema Gestione Account

Servizio	Descrizione	Interfaccia
Registrazione	Questa funzionalità permette di registrarsi sulla piattaforma	AccountService
Login	Questa funzionalità permette di effettuare l'accesso alla piattaforma attraverso le proprie credenziali	AccountService
Logout	Questa funzionalità permette di disconnettersi dalla piattaforma	AccountService
Visualizza dati account	Questa funzionalità permette di visionare i dati personali dell'utente/admin	AccountService
Modifica dati account	Questa funzionalità permette di modificare i dati personali dell'utente/admin	AccountService
Eliminazione account	Questa funzionalità permette di eliminare l'account personale dell'utente/admin	AccountService
Visualizzazione storico ordini	Questa funzionalità permette di visualizzare lo storico degli ordini effettuati	AccountService



## Sottosistema Catalogo

Servizio	Descrizione	Interfaccia
Visualizzazione elenco libri	Questa funzionalità permette di visionare l'elenco dei libri presenti sulla piattaforma	CatalogService
Visualizzazione scheda prodotto	Questa funzionalità permette di visionare i dettagli relativi a un libro, quali autore, prezzo, descrizione e disponibilità	CatalogService
Ricerca / Filtro libri	Questa funzionalità permette la ricerca di un libro tramite ricerca per parola o tramite filtri appositi che possono essere applicati sulla tipologia o sul prezzo del libro	CatalogService

## Sottosistema Carrello

Servizio	Descrizione	Interfaccia
Visualizzazione carrello	Questa funzionalità permette di visionare la pagina del carrello con eventuali prodotti all'interno	CartFacade
Aggiungi prodotto	Questa funzionalità permette di aggiungere un prodotto al carrello	CartFacade
Rimuovi prodotto	Questa funzionalità permette di rimuovere un prodotto dal carrello	CartFacade
Modifica quantità	Questa funzionalità permette di modificare la quantità di un prodotto presente nel carrello	CartFacade



## Sottosistema Acquisto

Servizio	Descrizione	Interfaccia
Acquista prodotto	Questa funzionalità permette di procedere all'acquisto dei prodotti presenti all'interno del carrello	OrderService
Inserimento dati di pagamento	Questa funzionalità permette la compilazione del form dei dati necessari alla conferma dell'ordine ( <u>pagamento simulato</u> ), con controlli di formato/coerenza sui campi inseriti.	OrderService

## Sottosistema Recensioni

Servizio	Descrizione	Interfaccia
Rilascia recensione	Questa funzionalità permette di rilasciare una recensione ad un libro	ReviewService
Visualizza recensione	Questa funzionalità permette di visualizzare le eventuali recensioni rilasciate ad un libro	ReviewService
Modifica / Cancellazione di una recensione	Questa funzionalità permette di modificare o cancellare una recensione precedentemente rilasciata	ReviewService



## Sottosistema Gestione Amministrativa

Servizio	Descrizione	Interfaccia
Visualizzazione lista utenti	Questa funzionalità permette all'admin di visualizzare l'intera lista di utenti registrati sulla piattaforma	AdminUserService
Modifica dati utenti	Questa funzionalità permette all'admin di modificare i dati di un utente	AdminUserService
Rimozione utenti	Questa funzionalità permette all'admin di eliminare un utente dalla piattaforma	AdminUserService
Aggiungi libro	Questa funzionalità permette all'admin di aggiungere un libro al catalogo	AdminCatalogService
Rimuovi libro	Questa funzionalità permette all'admin di rimuovere un libro dal catalogo	AdminCatalogService
Modifica dettagli libro	Questa funzionalità permette all'admin di modificare i dettagli di un libro	AdminCatalogService
Visualizzazione e moderazione recensioni	Questa funzionalità permette all'admin di visionare le recensioni presenti ed eventualmente eliminarle se ritenute inappropriate	AdminReviewService
Visualizzazione storico ordini piattaforma	Questa funzionalità permette all'admin di visionare lo storico degli ordini sulla piattaforma	AdminOrderService



## 5. Glossario

Termino	Definizione
Ajax	Tecnica di sviluppo web che consente lo scambio asincrono di dati tra client e server senza ricaricare l'intera pagina.
Architettura three-tier	Architettura software che suddivide il sistema in livello di presentazione, livello di logica applicativa e livello di gestione dei dati.
Atomicità delle Transizioni	Proprietà che garantisce l'esecuzione completa di un'operazione composta oppure il suo completo annullamento.
Boundary Condition	Situazione particolare che descrive il comportamento del sistema in casi eccezionali, come avvio, spegnimento o fallimento.
COTS	Componente software o hardware già disponibile sul mercato e integrato nel sistema senza sviluppo personalizzato.
Dependability	Categoria di obiettivi di design che riguarda affidabilità, disponibilità e tolleranza ai guasti del sistema.
Deployment	Processo di installazione e configurazione del sistema software sull'infrastruttura di esecuzione.
Design Goal	Qualità architettoniche o di servizio che il sistema deve perseguire in base ai requisiti non funzionali.
Event Driven	Modello di controllo del software in cui il flusso di esecuzione è determinato da eventi generati dalle interazioni dell'utente.
GUI	Interfaccia grafica che consente all'utente di interagire con il sistema tramite elementi visivi.



JDBC	API Java che consente la connessione e l'interazione con database relazionali.
Mapping Hardware-Software	Descrizione della distribuzione dei componenti software sull'infrastruttura hardware.
MVC	Pattern architettonico che separa la gestione dei dati, la logica applicativa e la presentazione.
Repository	Componente che fornisce un'interfaccia di accesso ai dati, astratta rispetto alla loro memorizzazione fisica.
Servlet	Componente server-side che gestisce le richieste HTTP e coordina la logica applicativa.
Sottosistema	Insieme coerente di componenti software che forniscono servizi legati a una specifica area funzionale.
System Failure	Condizione in cui il sistema non è in grado di fornire correttamente i propri servizi.