Università degli Studi di Salerno
Corso di Ingegneria del Software

BirdComics
System Design Document
Versione 2.0

BirdComics

Data: 24/11/2024

Coordinatore del progetto:

Nome	Matricola

Partecipanti:

Nome	Matricola
De Rosa Giuseppe	0512110683
Buccino Adriano	0512120484

Scritto da:	

Revision History

Data	Versione	Descrizione	Autore
07/01/2024	2.0	Riorganizzazione del documento con ridefinizione dei sottosistemi e del relativo hw/sw mapping	De Rosa Giuseppe Buccino Adriano
24/11/2024	1.0	Prima Versione del System Design Document	De Rosa Giuseppe Buccino Adriano

1. INTRODUZIONE

1.1. Scopo del sistema

Il sistema ha lo scopo di sviluppare e gestire una piattaforma di e-commerce dedicata alla vendita di fumetti, denominata *BirdComics*. La creazione di un e-commerce focalizzato sulla vendita di fumetti, si rivela una risposta appropriata alle necessità di un mercato in continua crescita. Differenziandosi nettamente da un negozio fisico. Un aspetto fondamentale di questa iniziativa è l'accessibilità: un e-commerce è aperto 24 ore su 24, 7 giorni su 7, permettendo ai consumatori di effettuare acquisti in qualsiasi momento, senza le restrizioni di orario imposte dai negozi tradizionali. Questa piattaforma permette agli utenti di acquistare fumetti in formato cartaceo, fornendo maggiore flessibilità e comodità esplorando una vasta selezione di titoli (dai grandi classici ai fumetti in edizione limitata o rari da trovare) che un negozio online può offrire rispetto a uno spazio fisico, il quale è limitato dalla capacità espositiva. L'obiettivo principale è offrire un'esperienza di acquisto comoda, accessibile e personalizzata, raggiungendo un pubblico globale e soddisfacendo le crescenti esigenze dei lettori di fumetti. Inoltre, il sistema dovrà essere altamente scalabile, permettendo di gestire un ampio catalogo di prodotti, utenti e transazioni senza compromettere le performance.

1.2. Definizioni, acronimi e abbreviazioni

RNF: Requisito Non Funzionale
RAD: Requirements Analysis Document

1.3. Riferimenti

Nel corso del documento facciamo riferimento ai Requisiti Non Funzionali, agli Use Cases e al Class Diagram presenti nel documento RAD.

1.4. Panoramica

Nei paragrafi successivi presentiamo l'analisi architetturale della piattaforma BirdComics. La sua decomposizione in sottosistemi, strategie di deployment su

hardware, strategie per la gestione dei dati persistenti ed eventuali Boundary Condition del sistema.

2. Architettura attuale

Il sistema che andremo a sviluppare rappresenta un progetto completamente nuovo e innovativo, poiché al momento non esistono soluzioni simili o analoghe nel mercato. Attualmente, non esiste una piattaforma dedicata alla vendita online di fumetti con le caratteristiche e le funzionalità che intendiamo implementare. Pertanto, il nostro progetto avrà una dimensione pionieristica, occupando uno spazio di mercato vuoto e rispondendo a esigenze non ancora adeguatamente soddisfatte da altri attori del settore.

3. Architettura software proposta

3.1. Obiettivi di progettazione

3.1.1. Prestazioni

Tempo di risposta	Il sistema deve rispondere a ogni richiesta dell'utente (come la navigazione o l'acquisto) entro 2 secondi (**RNF7**).
Throughput	Il sistema deve supportare almeno 1000 utenti simultanei senza degradare le prestazioni (**RNF8**).
Memoria	L'uso della memoria deve essere ottimizzato per mantenere prestazioni elevate garantendo il tempo di risposta ed il throughput sopra citati
Velocità ricerca	I risultati della ricerca nel catalogo prodotti devono essere aggiornati in tempo reale senza ricaricare la pagina (**RNF9**).

3.1.2. Garanzia di funzionamento

Robustezza	Il sistema deve resistere a input non validi e a comportamenti imprevisti degli utenti senza andare in crash.
Affidabilità	Il sistema deve garantire un funzionamento stabile, con una disponibilità del 99.9% (**RFN5**).
Disponibilità	Il sito deve essere operativo quasi costantemente, con downtime ridotto al minimo per manutenzione
Tolleranza ai guasti	Deve continuare a funzionare anche in caso di errori, ad esempio grazie a server di backup
Sicurezza	Protezione contro accessi non autorizzati (**RNF1**), password crittografate (**RNF2**), prevenzione di attacchi SQL Injection (**RNF3**).
Sicurezza utenti	Protezione dei dati personali degli utenti e delle transazioni di pagamento.
Sicurezza catalogo	Gli aggiornamenti al catalogo e alle scorte devono essere immediati (**RFN6**)

3.1.3. Manutenzione

Estendibilità	Facilità nell'aggiungere nuove funzionalità (come l'integrazione con nuovi gateway di pagamento).
Modificabilità	Possibilità di modificare funzionalità esistenti senza influire negativamente sul sistema.
Adattabilità	Capacità di adattarsi a diversi contesti o domini (ad es. un altro paese con diverse normative).
Leggibilità	Codice ben documentato e strutturato per

	facilitare la comprensione e il debug (**RNF10**).
Tracciabilità dei requisiti	Possibilità di mappare facilmente il codice alle specifiche originali del progetto.

3.1.4. Utente finale

Utilità	Il sistema deve supportare gli utenti nel completare i loro acquisti, gestire i carrelli, e seguire i processi di pagamento in modo semplice.
Usabilità	Il sistema deve essere intuitivo, con interfaccia user-friendly
Feedback	Il sistema deve fornire messaggi di conferma e notifiche degli errori in tempo reale (**RNF4**).
Accessibilità	Deve essere accessibile su vari dispositivi, come desktop, tablet e smartphone.

3.1.5. Compromessi

Spazio vs. Velocità	Potrebbe essere necessario utilizzare più memoria per ottenere tempi di risposta più rapidi
Tempo di consegna vs. Funzionalità	Ridurre le funzionalità per rispettare le scadenze di lancio del sito
Sicurezza vs. Usabilità	Aumentare la sicurezza (ad esempio, richiedendo password complesse e autenticazione a due fattori) può rendere il sistema meno intuitivo e creare frustrazione per gli utenti che cercano un'esperienza rapida e fluida.

3.2. Gestione dati persistenti

La gestione dei dati persistenti è fondamentale in un sistema distribuito. Rendere i dati persistenti, ovvero archiviati su memoria di massa non volatile, permette ai sottosistemi in esecuzione su VM o su Host fisici sparsi per la rete, di accedere ad un archivio, che sia esso centrale o distribuito, garantisce che le informazioni siano conservate e recuperabili in qualsiasi momento.
L'archivio che andremo a realizzare per BirdComics sarà separato dal sistema distribuito TomCat e gestito da nodi specializzati alla sola archiviazione dei dati. L'archiviazione è resa possibile tramite i servizi di database e Network Attached Storage installati su diversi nodi fisici, distribuiti geograficamente.
Per la gestione dei database utilizziamo il server MySql, che tratterà i dati di tipo carattere, stringhe, interi, ecc… mentre per l'archiviazione dei file multimediali, come immagini, pdf, ecc… si utilizzerà un sistema di Network Attached Storage.
Implementando queste tecnologie esterne al sistema TomCat, garantiamo come caratteristiche minori, che al riavvio del sistema, i dati non debbano essere ricaricati dall'attore in quanto "sopravvissuti" al riavvio.
Le caratteristiche più importanti includono la distribuzione geografica dei dati, che consente ai clienti di accedervi più rapidamente grazie alla riduzione della distanza fisica (numero di hop), la gestione dell'accesso concorrente tramite i database, la duplicazione dei dati su più hop, la scalabilità attraverso l'implementazione di ulteriori host e la sicurezza.
Il sistema sarà implementato in questo modo, nelle prime fasi iniziali, per ridurre il costo di realizzazione iniziale e dato che i clienti non saranno molti da far rallentare il sistema, ed i fumetti non saranno molti si è deciso di integrare le immagini all'interno del server TomCat. Una volta che il sito amplierà il catalogo ed i clienti aumenteranno, si passerà ad un sistema di gestione immagini esterno NAS.

3.3. Modello strutturale

Il sistema BirdComics è sviluppato e diviso secondo questi modelli:

- **Server TomCat**: l'applicativo utilizzerà il modello MVC (Model View Control) implementato tramite il modello Observer tra Model e View. La View non accede al model in maniera diretta, ma si iscrive a dei

Topic, dove preleva le informazioni che precedentemente il model aveva pubblicato. Nella modifica delle informazioni, la view passa le informazioni al Control e quest'ultimo per salvare le informazioni di modifica, passa le informazioni al model in maniera diretta. Il model sarà diviso in sottosistemi, dove ogni sottosistema avrà un ruolo specifico con basso accoppiamento tra loro, inoltre saranno implementati tramite stile "facade". Per ogni sottosistema implementato nel Model, esso è implementato tramite la stratificazione chiusa quindi un modulo del sottosistema non potrà essere invocato o utilizzare i servizi forniti da un altro sottosistema.

- **Client**: Il client si interfaccerà unicamente con la View del sistema Tomcat, la comunicazione sarà di tipo client-server utilizzando il protocollo https
- **Database**: Per salvare le informazioni il Model, interagisce con i database non in maniera diretta in quanto creeremo alto accoppiamento, ma tramite Strategy Pattern, quindi nel Model creeremo un'interfaccia invocata tra tutti i sottosistemi e lato database, ci sarà un sistema che implementa l'interfaccia. Utilizzando questa strategia possiamo garantire basso accoppiamento garantendo il cambio di database senza modificare tutti i sottosistemi esistenti, ma semplicemente creando una nuova implementazione dell'interfaccia.

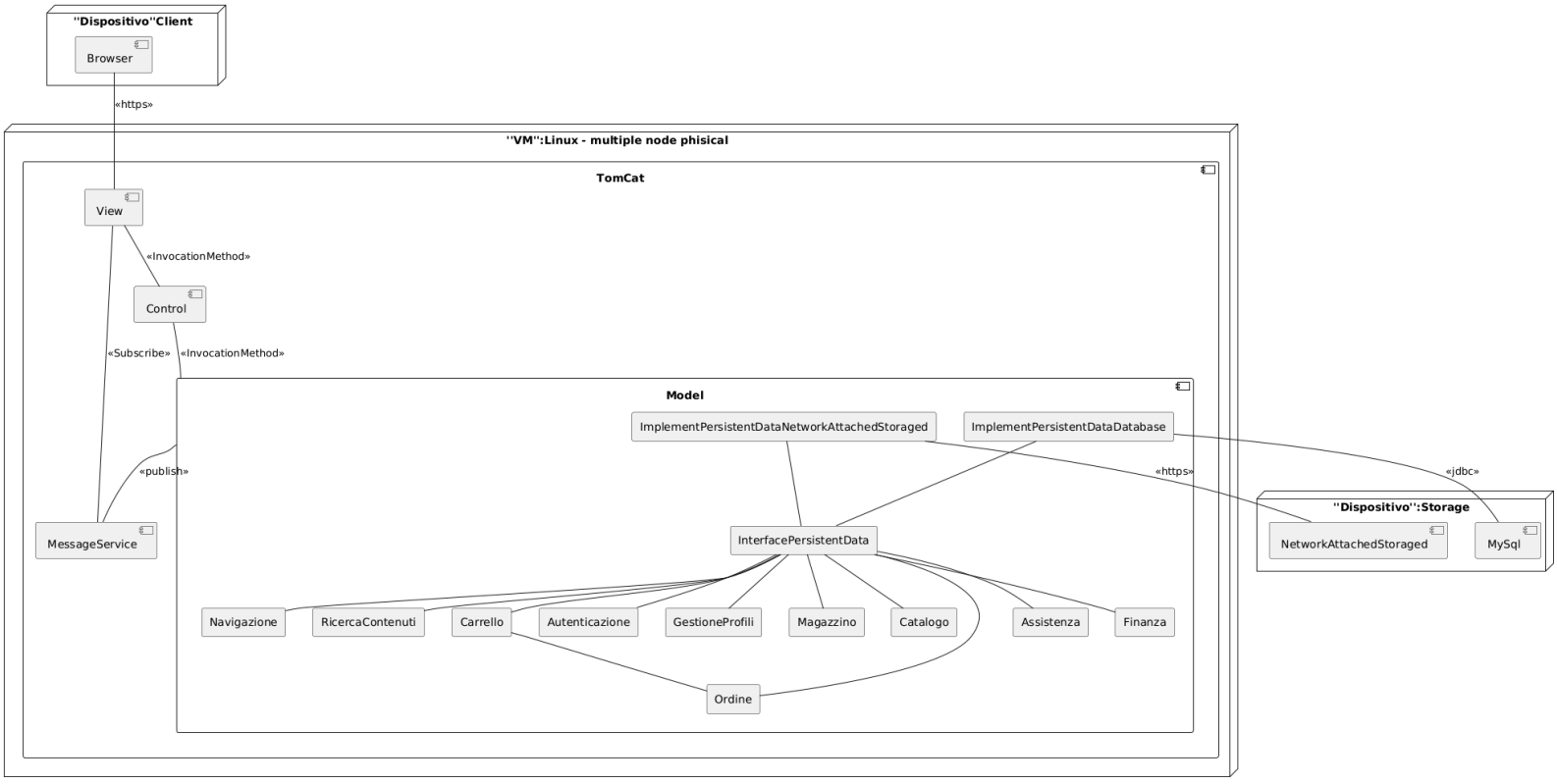
3.4. Controllo Software

Il sistema BirdComics adotta un'architettura di tipo event-driven, in cui le richieste degli utenti provenienti dal client browser, vengono inizializzate tramite interazioni con l'interfaccia grafica. Quando un utente effettua un'azione, come cliccare su un fumetto per acquistarlo o aggiornare il proprio profilo, viene generato un evento che interagisce con il layer VIEW presente sui server BirdComics. Per ogni evento generato dall'utente, esiste un rispettivo listener sul sistema che innesca un processo di elaborazione. L'elaborazione delle richieste da parte della View si traduce, in un cambio di pagina o in un cambio di stato dei dati. Nel caso di cambio pagina, la view interagisce con il layer control il quale provvede a reindirizzare l'attore sulla pagina richiesta. Nel caso di cambio consistente dei dati (aggiungi al carrello, modifica profilo, effettua acquisto, etc…) la View riceve i dati da parte del browser, inseriti da parte dell'attore, li inoltra

al Control il quale effettua delle operazioni su questi dati se ne ha bisogno, e poi questi dati vengono inoltrati al Model il quale aggiorna i dati persistenti ricevuti. Come è possibile intuire sia il Control che il Model ricevono eventi dagli strati a loro collegati. Per il caricamento e la visualizzazione delle informazioni persistenti, Il Model carica i dati e li inoltra ad un sistema di messaggistica basato su publish/subscribe. La View si registra come lettore su un “listener specifico”, il Model pubblica le informazioni sopra questo listener. Se il Control invoca un cambio di stato all’interno del Model, esso provvede prima a cambiare le informazioni all’interno del set dati persistenti ed una volta aggiornate manda un nuovo messaggio sul listener, la View riceve il nuovo messaggio ed aggiorna le informazioni sul browser del client mostrando le informazioni aggiornate.

3.5. Mappatura Hardware e Software

Il sistema distribuito di BirdComics non è stato progettato per girare su una configurazione hardware specifica, ma bensì su una configurazione software specifica. Gli host possono essere costruiti con un ampio parco hardware, purché sia architettura x86 a 64 bit con la possibilità di differenziare l’hardware di ogni singolo host. La configurazione software di ogni host è un sistema operativo che permette di distribuire la potenza di calcolo, proxmox, con installato al suo interno una macchina virtuale linux contenente tomcat versione 9.



3.6. Boundary Condition

Le condizioni boundary identificate per il nostro e-commerce includono situazioni legate all'avvio del sistema, il suo spegnimento, il riavvio in seguito a un fallimento.

3.6.1. Primo avvio

Prima del primo avvio, lo sviluppatore che sta configurando il sistema dovrà inserire manualmente almeno un profilo Direttore Generale all'interno del database MySQL, dopo aver avviato il relativo servizio. Questo è necessario perché, nella nostra piattaforma, solo un amministratore può registrare nuovi amministratori o gestori.
Completata questa operazione, si potrà avviare il web server che ospita l'e-commerce.

Per gli avvii successivi, sarà sufficiente avviare il DBMS e il web server.

3.6.2. Avvio in caso di fallimento

In caso di un'interruzione improvvisa del sistema, il database utilizza transazioni per garantire la consistenza delle informazioni. Eventuali modifiche parziali in corso non verranno salvate, evitando inconsistenze nei dati.
Non sono previste funzionalità per il ripristino dello stato precedente all'arresto improvviso.

3.6.3. Spegnimento

Per spegnere correttamente il sistema, sarà sufficiente terminare l'esecuzione del web server. Poiché il web server gestisce anche la connessione al DBMS, non sarà necessario spegnere manualmente il database.

Eventuali transazioni non completate verranno annullate, garantendo la consistenza dei dati persistenti.

3.6.4. Errato accesso ai dati persistenti

Se si verificano errori nella configurazione del database o nell'utilizzo del driver per connettersi ad esso, il sistema mostrerà una pagina di errore personalizzata. Questa pagina fornirà dettagli sulla causa e sul punto preciso del codice sorgente in cui si è verificato il problema.
Lo sviluppatore dovrà correggere il problema prima di riavviare il sistema. Una volta risolto, il sistema tornerà operativo come di norma.

3.7. Decomposizione in sottosistemi

3.7.1. Sottosistema Navigazione

3.7.1.1. Servizi offerti e specifica ad alto livello del comportamento

Questo sottosistema nella fase iniziale del progetto non verrà implementato e quindi utilizzato. Servirà a salvare le personalizzazioni della gui, permettendo all'attore di salvare le informazioni di personalizzazione su uno storage persistente. Possiamo ricaricare le stesse impostazioni su device diversi o sullo stesso nel momento in cui esegue una disconnessione e poi un'autenticazione successiva.

3.7.2. Sottosistema Ricerca contenuti

3.7.2.1. Servizi offerti e specifica ad alto livello del comportamento

Il sottosistema consente agli utenti di effettuare ricerche mirate sui fumetti presenti nel sistema. Saranno supportate diverse modalità di ricerca, come per genere, per nome o tramite identificativo univoco. Questo componente garantirà un'esperienza di navigazione semplice ed efficace, permettendo un rapido accesso alle informazioni richieste.

3.7.2.2. Gestione dati persistenti

Questo sottosistema non prevede dati persistenti.

3.7.2.3. Operazioni con parametri e controllo degli accessi

Nome Metodo	Parametri	Controllo Accesso
FumettiRandom	null	All
findGenere	genere	All
findTitolo	titolo	All
findId	id	GestoreCatalogo

3.7.3. Sottosistema Carrello

3.7.3.1. Servizi offerti e specifica ad alto livello del comportamento

Il sottosistema permette agli utenti di gestire in modo completo gli articoli selezionati per l'acquisto. Gli utenti possono aggiungere fumetti al carrello, visualizzarne il contenuto, modificare la quantità di ciascun fumetto, rimuovere singoli articoli o svuotare completamente il carrello. Questo sottosistema è progettato per garantire un’esperienza di gestione degli acquisti semplice e intuitiva, agevolando il processo di selezione e preparazione all’acquisto.

3.7.3.2. Gestione dati persistenti

Il carrello prevede di rendere i dati persistenti. L’attore ospite oppure cliente, una volta inserito almeno un fumetto al carrello (nel caso dell’ospite se chiude la pagina il carrello viene perso) se esso effettua la disconnessione dalla piattaforma, con un carrello che si trova in uno stato di esistenza popolato da qualche fumetto, esso viene salvato nell’archiviazione persistente di BirdComics. La volta successiva che il Cliente accede al proprio account, il carrello viene ricaricato dalla banca dati di BirdComics in modo che non abbia perso i fumetti di interesse e possa procedere direttamente con l’acquisto.

3.7.3.3. Operazioni con parametri e controllo degli accessi

Nome Metodo	Parametri	Controllo Accesso
totaleCarrello()	null	Cliente, Ospite
salvaCarrello()	null	Cliente
aggiungiACarrello()	fumetto, quantita	Cliente, Ospite
fumettiACarrello()	null	Cliente, Ospite
eliminaFumettoDalCarrello()	fumetto	Cliente, Ospite
svuotaCarrello()	null	Cliente, Ospite
modificaQuantitaCarrello()	fumetto, quantita	Cliente, Ospite
caricaCarrello()	null	Cliente

3.7.4. Sottosistema Autenticazione

3.7.4.1. Servizi offerti e specificare ad alto livello del comportamento

Il sottosistema gestirà l'accesso al sistema tramite funzionalità di registrazione, autenticazione e disconnessione. Gli utenti potranno creare un account fornendo i dati necessari, accedere al sistema attraverso la verifica delle proprie credenziali e terminare la sessione in modo sicuro. Il comportamento del sottosistema sarà orientato a garantire la protezione dei dati personali e delle sessioni, mantenendo un'esperienza fluida e intuitiva.

3.7.4.2. Gestione dati persistenti

3.7.4.3. Operazioni con parametri e controllo degli accessi

Nome Metodo	Controllo Accesso
registra()	Ospite
disconnessione()	Cliente, Gestori

autenticazione()	All

3.7.5. Sottosistema Gestione profili

3.7.5.1. Servizi offerti e specifica ad alto livello del comportamento

Il sottosistema consente ai clienti di visualizzare i propri ordini e le relative fatture, oltre a gestire le informazioni personali del proprio profilo, con la possibilità di modificarle o eliminarle in modo definitivo. Per quanto riguarda la gestione del personale, il sistema supporta operazioni quali l'assunzione, il licenziamento e la modifica dei dati relativi ai direttori di magazzino, agli addetti e al personale risorse umane. Questo componente è progettato per offrire una gestione centralizzata ed efficiente, garantendo la sicurezza delle informazioni e un'esperienza intuitiva per tutti gli utenti del sistema.

3.7.5.2. Gestione dati persistenti

3.7.5.3. Operazioni con parametri e controllo degli accessi

Nome Metodo	**Parametri**	**Controllo Accesso**
insertDipendente()	email, password, nome, cognome, data di nascita, indirizzo, telefono, codice postale, ruolo	Risorse Umane
insertHr()	email, password, nome, cognome, data di nascita, indirizzo, telefono, codice postale	Direttore Magazzino
insertDirettoreMagazzino()	email, password, nome, cognome, data di nascita, indirizzo, telefono, codice postale, ruolo	Direttore Generale
deleteDirettoreMagazzino	direttoreMagazzino	Direttore Generale
DeleteHr	hr	Direttore Magazzino

DeleteDipendente	dipendente	Risorse Umane
destroy	null	Cliente
info()	null	All
modificaPassword()	password	All
modificaIndirizzo()	indirizzo	Cliente
modificaInfo()	indirizzo, telefono, codice postale	Cliente
modificaInfoDipendente()	indirizzo, telefono, codice postale, dipendente	Risorse Umane
modificaInfoRisorseUmane()	indirizzo, telefono, codice postale, hr	Direttore Magazzino
modificaInfoDirettoreMagazzino()	indirizzo, telefono, codice postale, direttoreMagazzino	Direttore Generale
direttoriMagazzino()	null	Direttore Generale
risorseUmane()	null	Direttore Magazzino
dipendenti()	null	Risorse Umane

3.7.6. Sottosistema Magazzino

3.7.6.1. Servizi offerti e specifica ad alto livello del comportamento

Il sottosistema gestisce le operazioni legate all'organizzazione e al funzionamento del magazzino. Consente di aprire e chiudere il magazzino, aggiungere e rimuovere scaffali, e organizzare i fumetti sugli scaffali. Questo sottosistema è progettato per garantire un controllo preciso e flessibile della disposizione dei fumetti, facilitando la gestione e l'accessibilità dei contenuti nel magazzino.

3.7.6.2. Gestione dati persistenti

3.7.6.3. Operazioni con parametri e controllo degli accessi

Nome Metodo	Parametri	Controllo Accesso
sedi()	null	Direttore Generale
insertSede()	indirizzo, nomeSede	Direttore Generale
deleteSede()	sede	Direttore Generale
addScaffale()	nomeScaffale, nrFumetti	Direttore Generale
deleteScaffale()	nomeScaffale	Direttore Generale
scaffali()	null	Direttore Generale

3.7.7. Sottosistema Catalogo

3.7.7.1. Servizi offerti e specifica ad alto livello del comportamento

Il sottosistema gestisce l'organizzazione dei fumetti disponibili all'interno del sistema. Consente di aggiungere nuovi fumetti al catalogo, rimuovere quelli esistenti e modificarne i dettagli. Questo sottosistema è progettato per garantire un aggiornamento rapido e accurato delle informazioni, assicurando che il catalogo rimanga sempre coerente e aggiornato.

3.7.7.2. gestione dati persistenti

3.7.7.3. operazioni con parametri e controllo degli accessi

Nome Metodo	Parametri	Controllo Accesso
caricaContenuti	null	All
insertFumetto()	titolo, descrizione, prezzo, quantita	Gestore Catalogo
deleteFumetto()	fumetto	Gestore Catalogo
updateFumetto()	Fumetto, nome,	Gestore Catalogo

	descrizione, prezzo, quantita	
fumettiRandom()	null	All
quantitaDisponibile()	fumetto, quantita	All
info()	idFumetto	All

3.7.8. Sottosistema Ordine

3.7.8.1. Servizi offerti e specifica ad alto livello del comportamento

Il sottosistema Ordine gestisce le operazioni legate all'acquisto e al pagamento dei fumetti. Consente di finalizzare un acquisto, completare il pagamento e aggiungere un codice di tracciamento per monitorare la spedizione. Questo sottosistema garantisce una gestione fluida e sicura degli ordini, permettendo agli utenti di seguire l'intero processo dall'acquisto alla consegna.

3.7.8.2. Gestione dati persistenti

3.7.8.3. Operazioni con parametri e controllo degli accessi

Nome Metodo	**Parametri**	**Controllo Accesso**
saveOrder()	fumetti, paypalTransazione	Cliente
ordini()	cliente	Cliente, Assistenza
codiceTracciamento()	ordine	Cliente, Operatore logistico
insertTracciamento()	ordine	Operatore logistico
ordiniNoTracking()	null	Operatore logistico
modificaTracking	ordine	Operatore logistico

3.7.9. Sottosistema Assistenza

3.7.9.1. Servizi offerti e specifica ad alto livello del comportamento

Questo sottosistema nella fase iniziale del progetto non verrà implementato in quanto il supporto che l'azienda BirdComics prevede di fornire sarà prettamente Telefonico. Nelle fasi successive, in cui i clienti affluiranno e i call center saranno intasati, si prevederà qualche altra strategia per fornire assistenza. Nel momento si prevede di implementare altre strategie di assistenza, il sistema sarà predisposto con il sottosistema dedicato a questo.

3.7.10. Sottosistema Finanza

3.7.10.1. Servizi offerti e specifica ad alto livello del comportamento

Questo sottosistema nella fase iniziale del progetto non verrà implementato in quanto l'azienda BirdComics non ha ricevuto l'accesso alle API dell'agenzia delle entrate. La previsione di questo sottosistema, serve per l'interfacciamento con il sistema di interscambio dell'iva e fatturazione, non solo delle fatture che BirdComics emette ma anche quelle che riceve. Il reparto di gestione finanza utilizzerà questo modulo per vedere l'andamento dell'azienda finanziariamente e decidere quali scelte di marketing adottare.