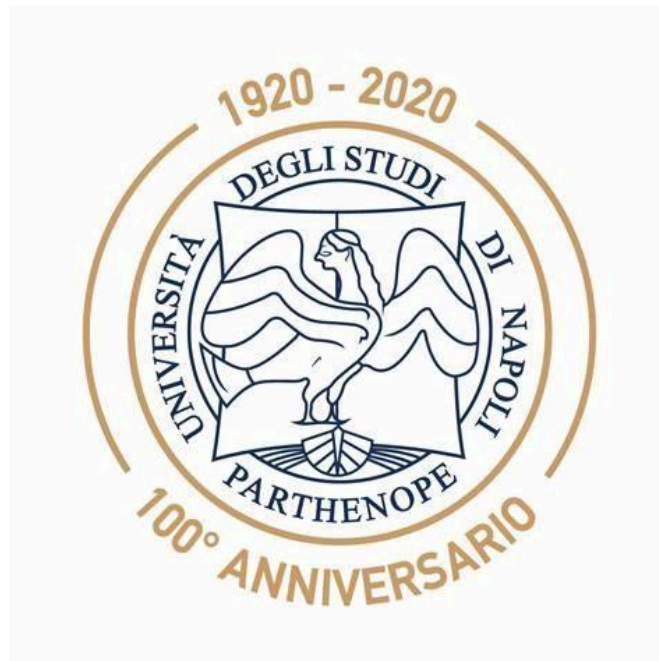


Università degli studi di Napoli Parthenope



Progetto Ingegneria del Software e Interazione Uomo Macchina

Games & Manga shop-SDD

Capasso Antonio - 0124002624

Cristiano Giuseppe - 0124002534

Lettieri Raffaele - 0124002631

INDICE

1 Introduzione.....	3
1.1 Scopo del sistema.....	3
1.2 Obiettivi di progettazione.....	3
1.3 Definizioni, acronimi e abbreviazioni.....	4
1.4 Riferimenti.....	5
1.4.1 Privilegi di Accesso.....	5
1.4.2 Adesione ai principi di una Progressive Web App.....	5
1.4.3 Utilizzo e creazione di API.....	6
2 SISTEMA CORRENTE.....	7
3 SISTEMA PROPOSTO.....	9
4 Panoramica.....	10
4.1 Decomposizione del sistema.....	10
4.2 Hardware/Software mapping.....	11
4.3 Gestione dei dati persistenti.....	12
4.4 Controllo accessi e sicurezza.....	12
4.5 Decisioni sul flusso di controllo globale.....	13
4.6 Condizioni Limite.....	16
5 SERVIZI DEL SOTTOSISTEMA.....	17
Interfaccia Login/Signup:.....	17
Interfaccia Utente:.....	17
Interfaccia Admin:.....	17
Interfaccia Fumettista Indipendente:.....	17
6 GLOSSARIO.....	18

1 Introduzione

1.1 Scopo del sistema

Nel mondo digitale in cui viviamo, la passione per i giochi e i manga si fonde con la comodità dello shopping online. Benvenuti in Games and Manga Shop, un punto di riferimento virtuale per gli appassionati dei fumetti manga e dei collezionisti di Funko Pop! Con una selezione accurata e diversificata di prodotti, il nostro obiettivo è offrire un'esperienza di shopping online senza pari per gli amanti della cultura pop giapponese e della cultura nerd occidentale.

Siamo consapevoli dell'importanza di offrire un servizio clienti impeccabile e un'esperienza di acquisto senza problemi. Pertanto, ci impegniamo a fornire una navigazione intuitiva del sito e un processo di acquisto sicuro e protetto.

In definitiva, Games and Manga Shop mira a essere molto più di una semplice piattaforma di vendita online; è un rifugio virtuale per gli appassionati dove la cultura pop giapponese e la cultura nerd occidentale si incontrano, creando un'esperienza di shopping immersiva e gratificante per tutti i fan.

1.2 Obiettivi di progettazione

- **Usabilità:**

Creare un'esperienza utente fluida e intuitiva, permettendo agli acquirenti di navigare agevolmente attraverso le varie categorie di prodotti, visualizzare dettagli, aggiungere articoli al carrello e completare gli acquisti in modo efficace.

- **Multiutenza:**

Implementare sessioni di accesso distinte: Offrire diverse sessioni di accesso per i vari ruoli all'interno del sistema:

Cliente: Consentire agli acquirenti di creare account personali, cercare determinati prodotti, inserirli nel carrello e procedere all'acquisto .

Amministratori: Fornire agli amministratori un accesso privilegiato per gestire il catalogo di prodotti, inserire sconti e di inserire un fumettista indipendente.

Fumettisti: Permettere ai fumettisti di contribuire all'inserimento dei loro fumetti nel negozio, gestendo i dettagli dei prodotti che desiderano mettere in vendita.

- **Privacy:**

Garantire la sicurezza dei dati e delle transazioni: Implementare misure di sicurezza robuste per proteggere i dati personali degli utenti, le transazioni finanziarie e le informazioni sensibili, prevenendo accessi non autorizzati e assicurando la conformità alle normative sulla privacy.

- **Versatilità:**

Il sistema si propone di essere flessibile e scalabile, in grado di adattarsi alle esigenze future, consentendo l'aggiunta di nuove funzionalità e l'espansione del catalogo di prodotti in modo efficiente e sicuro.

1.3 Definizioni, acronimi e abbreviazioni

1. **PWA: Progressive Web App** - Un'applicazione web che utilizza tecnologie moderne per fornire un'esperienza utente simile a quella di un'applicazione nativa su dispositivi mobili.
2. **API: Application Programming Interface** - Un set di definizioni e protocolli che consentono a software e applicazioni di comunicare tra loro. Un'API definisce i metodi e le strutture dati che possono essere utilizzati per interagire con un'applicazione.
3. **Python:** Python è un linguaggio di programmazione ad alto livello, interpretato, orientato agli oggetti e di facile apprendimento. È ampiamente utilizzato per lo sviluppo di software in vari settori, grazie alla sua semplicità, chiarezza e alla vasta gamma di librerie e framework disponibili.
4. **Flask:** Flask è un framework web leggero per Python che consente di creare applicazioni web in modo semplice e veloce. È progettato per essere flessibile e facile da usare, consentendo agli sviluppatori di creare rapidamente applicazioni web di diversi livelli di complessità.

5. **Route:** Una route in Flask è un modo per mappare una URL specifica ad una funzione Python all'interno dell'applicazione web. Le route definiscono gli endpoint HTTP che possono essere chiamati da un browser web o da altri servizi per eseguire determinate azioni o ottenere determinati dati.

1.4 Riferimenti

1. Privilegi di Accesso.
2. Conformità ai principi di una Progressive Web App.
3. Utilizzo e creazione di API.

1.4.1 Privilegi di Accesso

L'applicazione mette a disposizione 3 livelli di accesso:

- **Cliente:** Il livello con meno privilegi, può visualizzare i prodotti, inserirli nel carrello e procedere all'acquisto.
- **Fumettista indipendente:** Livello intermedio, possiede tutti i privilegi del livello inferiore, oltre ai privilegi per **aggiungere/rimuovere** i propri fumetti.
- **Admin:** Il livello massimo, possiede tutti i privilegi messi a disposizione dall'applicazione, permette di **inserire, modificare o rimuovere** prodotti, oltre all'inserimento di un nuovo fumettista indipendente.

1.4.2 Adesione ai principi di una Progressive Web App

Si ha conformità ai principi di una **Progressive Web App** quando un'applicazione web rispetta e implementa le caratteristiche fondamentali che definiscono una PWA. Queste caratteristiche includono:

- **Affidabilità:** L'applicazione deve essere affidabile e funzionare anche in condizioni di rete instabili o assenti. Ciò può essere ottenuto utilizzando la memorizzazione nella cache, in modo che l'applicazione possa essere caricata anche offline.

- **Velocità:** L'applicazione deve essere veloce e reattiva, garantendo tempi di caricamento rapidi e prestazioni fluide. Questo può essere raggiunto tramite tecniche di ottimizzazione del caricamento delle risorse e del rendering.
- **Esperienza utente coinvolgente:** L'applicazione deve fornire un'esperienza utente coinvolgente simile a quella di un'applicazione nativa, inclusa la navigazione senza soluzione di continuità, le animazioni fluide e una sensazione di reattività.
- **Installabilità:** L'applicazione deve essere installabile sul dispositivo dell'utente, consentendo di essere accessibile dalla schermata iniziale o dal menu delle app, senza la necessità di visitare un negozio di app.
- **Aggiornabilità:** L'applicazione deve essere facilmente aggiornabile in background, garantendo che gli utenti ricevano sempre le ultime correzioni di bug e miglioramenti delle funzionalità senza dover effettuare manualmente un aggiornamento.

1.4.3 Utilizzo e creazione di API.

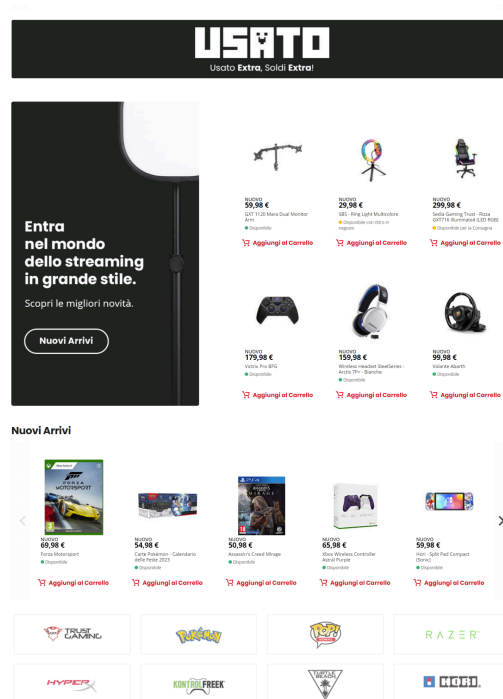
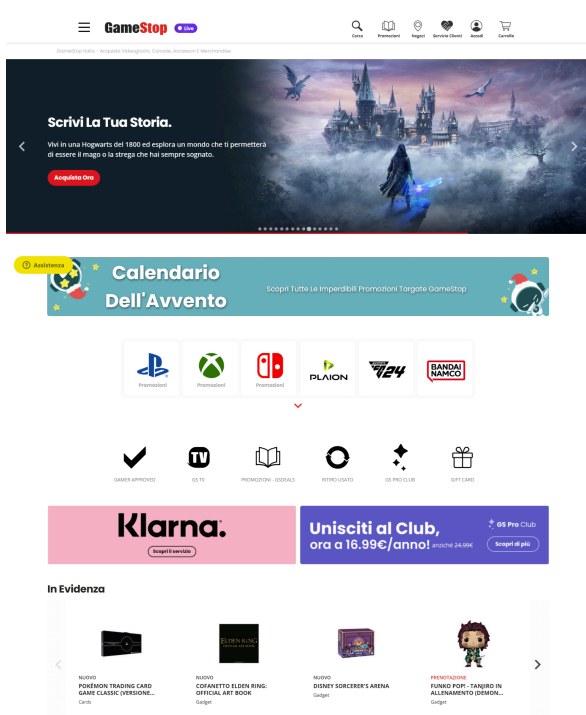
Le **API** (Application Programming Interface) sono insiemi di definizioni e protocolli che consentono a software e applicazioni di comunicare tra loro. In termini semplici, le API fungono da ponte che consente a due sistemi di interagire e scambiare informazioni.

- In **Flask**, un framework web per Python, è possibile creare API definendo delle "route" utilizzando il decoratore `@app.route()`. Queste route definiscono degli endpoint HTTP che possono essere chiamati da altri servizi o applicazioni per accedere ai dati o eseguire determinate azioni.

2 SISTEMA CORRENTE

Il sito che abbiamo preso in considerazione è quello di gamestop, uno dei più noti rivenditori di videogiochi, accessori e hardware. Alcuni dei punti a favore di questo sito sono:

- l'ampia selezione di prodotti;
- i vari programmi di membership e ricompense;
- le offerte speciali che propone;
- i servizi di ricondizionamento e scambio che consentono ai clienti di scambiare giochi e dispositivi usati, offrendo loro la possibilità di acquistare prodotti a prezzi più convenienti.



Nonostante il sito sia molto conosciuto, dal nostro punto di vista risulta essere molto confusionario e disorganizzato. Come si può notare dalle foto c'è un uso elevato di banner e pubblicità che rendono l'esperienza dell'utente più complicata di quello che dovrebbe essere. A questo si unisce il fatto che i prodotti vengono esposti in maniera piuttosto caotica, con item appartenenti a categorie totalmente diverse esposti uno di fianco all'altro.

Detto ciò riconosciamo l'importanza di GameStop nel mondo dei rivenditori di videogiochi, ma crediamo che ci sia spazio per miglioramenti

3 SISTEMA PROPOSTO

Link dell'applicativo: [gamesandmangashop](https://gamesandmangashop.com).

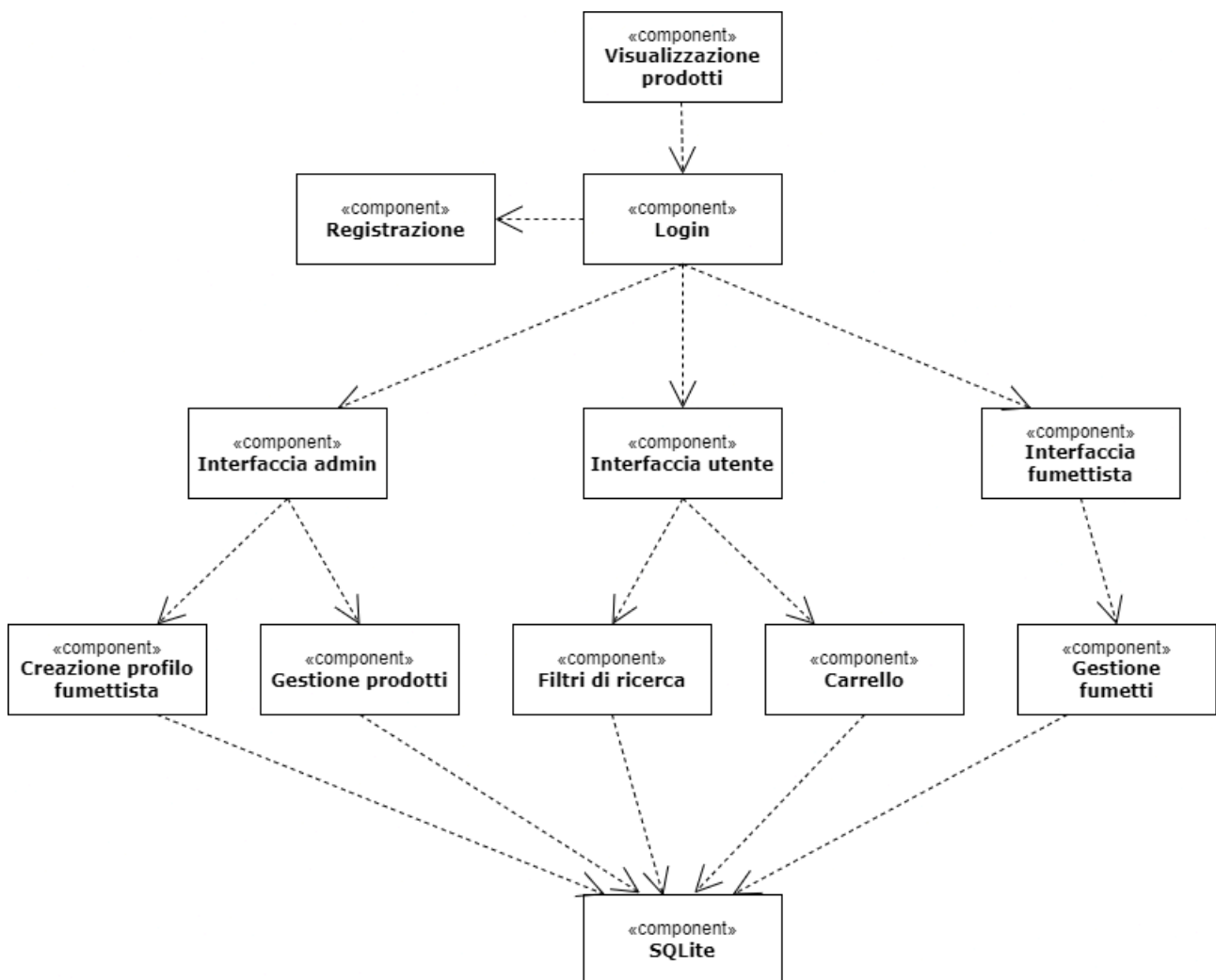
Paragonando le due home-page si può notare come gli articoli siano posizionati in maniera più schematica e ordinata fin da subito, mettendo in evidenza un pannello scorrevole con i prodotti principali, come ad esempio: “ultimi arrivi”, “giochi sotto ai 40€” e “giochi per tutti”.

È ben chiara la distinzione tra le varie sezioni del sito tramite il menù a tendina posizionato in cima alla home-page che permette di spostarsi rapidamente tra le varie pagine.

4 Panoramica

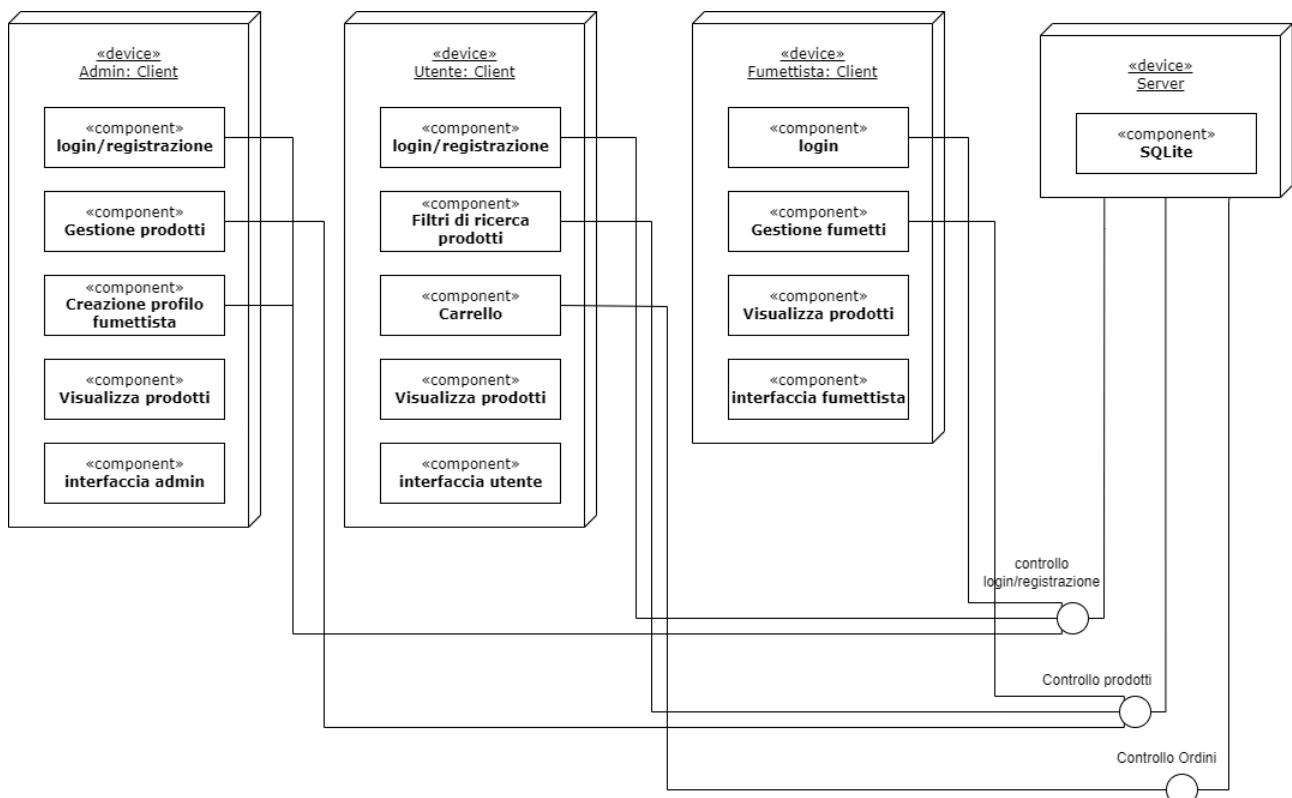
"Games and Manga Shop" è un'applicazione web che offre una vasta selezione di videogiochi, manga e Funko Pop per gli appassionati di cultura pop. Questa PWA combina le caratteristiche di un sito web e di un'applicazione mobile, offrendo un'esperienza utente fluida e responsive su diverse piattaforme e dispositivi. Gli utenti possono navigare facilmente tra le varie categorie di prodotti, effettuare acquisti in modo sicuro. Grazie alla natura progressiva dell'app, i clienti possono accedere rapidamente al negozio senza dover scaricare o installare nulla, rendendo l'esperienza di shopping comoda e immediata.

4.1 Decomposizione del sistema



- **Visualizzazione prodotti:** Permette di visionare i prodotti.
- **Login:** Permette di effettuare l'accesso ai vari utenti, all'admin o ai fumettisti.
- **Registrazione:** Permette di effettuare la registrazione agli utenti.
- **Interfaccia admin:** Fornisce i servizi di interfaccia grafica per l'admin.
 - **Creazione profilo fumettista:** Permette all'admin di creare un profilo specifico per i fumettisti.
 - **Gestione prodotti:** Permette di aggiungere o rimuovere prodotti dal catalogo e aggiungere sconti.
- **Interfaccia fumettista:** fornisce i servizi di interfaccia grafica per un fumettista
 - **Gestione fumetti:** Permette ai fumettisti di inserire o rimuovere i propri fumetti dal catalogo.
- **Interfaccia utente:** Fornisce i servizi di interfaccia grafica per un utente
 - **Carrello:** Permette la gestione dei prodotti che si vogliono acquistare.
 - **Filtri di ricerca:** Permettono di filtrare agevolmente i prodotti che ci interessano.
- **SQLite:** fornisce servizi per l'interfacciamento con il database interno.

4.2 Hardware/Software mapping



4.3 Gestione dei dati persistenti

Per la gestione di dati persistenti abbiamo usato **SQLite** perché dispone dei seguenti vantaggi :

Leggerezza e semplicità: SQLite è un database SQL leggero e senza server che può essere facilmente integrato nelle PWAs senza richiedere una configurazione complessa del server. Questo lo rende ideale per applicazioni web che richiedono uno storage locale dei dati.

Supporto dei browser: SQLite è supportato dalla maggior parte dei browser moderni, rendendolo una scelta affidabile per garantire la compatibilità e la portabilità delle PWAs su diverse piattaforme e dispositivi.

Performance: SQLite è progettato per essere leggero e veloce, offrendo prestazioni efficienti per le operazioni di lettura e scrittura dei dati. Questo è particolarmente vantaggioso per le PWAs che devono gestire grandi quantità di dati localmente senza dipendere da una connessione di rete.

4.4 Controllo accessi e sicurezza

- **Autenticazione Multi-Livello:**

Implementare un sistema di autenticazione robusto con livelli differenziati per Cliente, Fumettista indipendente e Amministratore.

- **Gestione dei Privilegi:**

Definire chiaramente i privilegi e le autorizzazioni per ciascun tipo di utente. Consentire ai clienti di poter visualizzare i prodotti ed eventualmente

acquistarli, garantire all'admin di gestire i prodotti del catalogo e di inserire dei fumettisti indipendenti, i quali dovranno poter inserire o rimuovere solo i loro fumetti pubblicati nel sito.

- **Crittografia dei Dati:**

Garantire la riservatezza e l'integrità dei dati durante la trasmissione e lo storage, impedendo accessi non autorizzati e proteggendo le informazioni da letture o modifiche non autorizzate.

Amministratore	Fumettista indipendente	Cliente	Operazioni
✓	✓	✓	Login
✗	✗	✓	Registrazione
✗	✗	✓	Visualizzazione Prodotti
✗	✗	✓	Acquisto Prodotti
✓	✗	✗	Gestione Prodotti
✓	✗	✗	Inserimento Fumettista Indipendente
✓	✗	✗	Inserimento Sconti
✓	✓	✗	Gestione fumetti

4.5 Decisioni sul flusso di controllo globale

Le decisioni sul flusso di controllo globale servono per fornire una panoramica dell'architettura del software e delle strategie adottate per gestire il flusso di esecuzione del programma.

Architettura Client-Server

Il nostro applicativo è strutturato seguendo un'architettura client-server, dove il lato client è composto da HTML, CSS e JavaScript, mentre il lato server è implementato in Python utilizzando Flask come framework. Questa architettura permette una separazione chiara tra la logica di presentazione dei dati e la gestione delle operazioni di backend.

Risorse Client

Le risorse client, come le pagine HTML, i fogli di stile CSS e gli script JavaScript, sono organizzate in modo da presentare contenuti interattivi agli utenti attraverso il browser web. Le pagine HTML definiscono la struttura del contenuto, i fogli di stile CSS gestiscono l'aspetto visuale e gli script JavaScript aggiungono funzionalità interattive all'interfaccia utente.

Interazione Utente

L'interazione utente è gestita principalmente attraverso eventi JavaScript, che vengono utilizzati per catturare input dagli utenti e fornire risposte dinamiche. Ad esempio, gestiamo la validazione dei moduli utilizzando JavaScript per garantire che i dati inseriti dagli utenti siano corretti e gestiamo i clic sui pulsanti per attivare azioni specifiche nell'applicativo.

Comunicazione Client-Server

La comunicazione tra il client e il server avviene tramite richieste HTTP e AJAX. Utilizziamo gli endpoint Flask per gestire queste richieste sul lato server, fornendo risposte dinamiche agli utenti senza dover ricaricare completamente la pagina web.

Flusso di Controllo lato Server

Il flusso di controllo lato server è gestito utilizzando Python e Flask. Definiamo i modelli di routing nelle route Flask per indirizzare le richieste degli utenti ai corrispondenti gestori di funzioni, che si occupano di elaborare le richieste e fornire le risposte appropriate.

Persistenza dei Dati

I dati dell'applicazione, come le informazioni sui prodotti in vendita, sono gestiti e memorizzati utilizzando un database. Flask interagisce con il database per recuperare e memorizzare le informazioni necessarie, garantendo una gestione efficiente e affidabile dei dati dell'applicazione.

Gestione degli Errori

Gli errori sono gestiti sia lato client che lato server. Utilizziamo meccanismi di gestione delle eccezioni in Python e Flask per gestire errori imprevisti lato server, mentre presentiamo messaggi di errore agli utenti attraverso l'interfaccia utente per segnalare errori lato client.

4.6 Condizioni Limite

Di seguito sono elencati potenziali problemi o scenari che potrebbero incorrere in bug.

- **Bug sull'aggiunta di articoli al catalogo:** si potrebbe verificare un bug qualora l'admin o un fumettista aggiungano un prodotto al catalogo che contiene un'immagine associata di dimensioni particolarmente diverse dallo standard.
- **Problemi di Connessione al Database:** potrebbero verificarsi interruzioni della connessione al database, interrompendo le funzionalità del sistema.
- **Problemi di performance:** potrebbero esserci dei rallentamenti dell'applicativo dovuto ad un numero di utenti non previsto.

5 SERVIZI DEL SOTTOSISTEMA

Di seguito sono riportate le funzionalità o i compiti specifici forniti da un sottosistema all'interno del software.

Interfaccia Login/Signup:

- **Servizio di registrazione:** Consente agli utenti di creare nuovi account fornendo informazioni come nome utente, email e password.
- **Servizio di login:** Permette agli utenti di accedere all'applicazione utilizzando le loro credenziali di accesso.

Interfaccia Utente:

- **Servizio di ricerca:** Consente agli utenti di cercare videogiochi, fumetti e Funko utilizzando filtri come categoria e prezzo.
- **Servizio di gestione carrello:** Permette agli utenti di aggiungere, rimuovere e modificare gli articoli nel loro carrello di acquisti.
- **Servizio di checkout:** Gestisce il processo di pagamento e conferma degli ordini.

Interfaccia Admin:

- **Servizio di gestione prodotti:** Permette agli amministratori di aggiungere, modificare e rimuovere prodotti dal catalogo.
- **Servizio di gestione sconti:** Consente agli amministratori di creare e gestire sconti sui prodotti.
- **Servizio di gestione utenti:** Permette agli amministratori di gestire gli account degli utenti, inclusi privilegi speciali e credenziali.

Interfaccia Fumettista Indipendente:

- **Servizio di caricamento fumetti:** Consente ai fumettisti indipendenti di caricare o rimuovere i propri fumetti sul sito.

6 GLOSSARIO

Termine	Definizione	Sinonimi
Admin	Utente che gestisce i prodotti del catalogo, autorizza i fumettisti indipendenti di aggiungere i loro fumetti e che è responsabile del sito e della sua manutenzione.	Amministratore, proprietario.
Fumettista indipendente	Utenti autorizzati dall'admin per la gestione indipendente dei fumetti.	Fumettista.
Cliente	Utente con meno privilegi che ha solo la possibilità di visualizzare e magari comprare degli oggetti presenti nel catalogo.	Utente base.
Prodotto	Articoli presenti nel catalogo del negozio.	Videogioco, manga, fumetto, funko-pop.