

**Documentazione System Design**

**“GLITCH”**

**Progetto presentato da:**

Annunziata Elefante

Ferdinando Napolitano

Santolo Mutone

**INDICE**

1. **Introduzione . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .** 
   1. Scopo del sistema . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   2. Obiettivi di progettazione . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   3. Definizioni, acronimi e abbreviazioni . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   4. Panoramica . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
2. **Architettura software corrente . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .**
3. **Architettura software proposta . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .** 
   1. Panoramica . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   2. Decomposizione del sottosistema . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   3. Mappatura hardware/software . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   4. Gestione dei dati persistenti . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   5. Controllo e sicurezza degli accessi. . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   6. Controllo globale del software . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
   7. Condizioni limite . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
4. **Servizi del sottosistema** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
5. **Glossario** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
6. **Introduzione**
   1. **Scopo del sistema**

Il progetto Glitch è stato commissionato con le specifiche di permettere a tutti gli utenti registrati di poter acquistare, comodamente da casa, tutti i videogiochi e le console che desiderano.

Il sistema proposto andrà a velocizzare la vendita dei prodotti sopra citati, permettendo il passaggio da un negozio fisico, in cui il cliente deve recarsi senza avere la certezza di trovare ciò di cui necessita, a un e-commerce ricco di prodotti e di facile utilizzo. Ne segue che il software sarà rivolto a tutti gli appassionati di “gaming”, ai quali sarà consentito di inserire i prodotti scelti in un apposito carrello personale e acquistarli comodamente dal proprio divano.

Tenendo presente le specifiche, Glitch dovrà permettere di filtrare i prodotti mediante appositi filtri, consentirne l’acquisto o la rimozione dal carrello dei prodotti, permettere un servizio di assistenza, gestire il catalogo.

* 1. **Obiettivi di progettazione**

Glitch è stato sviluppato per semplificare le procedure di visualizzazione e vendita di console e videogiochi. Ciò grazie ad interfacce intuitive che consentono di interagire e gestire il sistema.

Il sistema si appoggia su un DBMS esterno, sfruttandone al massimo le potenzialità di ottimizzazione per il recupero dei dati. La documentazione gestita da Glitch è a carattere riservato e ad uso esclusivo dell’utente, pertanto si è scelto di adottare un sistema di autenticazione che controlla l’accesso al sistema tramite l’inserimento di un username e una password assegnati agli utenti al momento della registrazione.

Il linguaggio di programmazione scelto per il suo sviluppo è Java, con paradigma di programmazione ad oggetti.

* 1. **Definizioni, acronimi e abbreviazioni**
* **SDD**: System Design Document
* **DBMS**: DataBase Management System
* **DB**: DataBase
  1. **Panoramica**

Il seguente documento SDD è diviso in sezioni ed ha la seguente composizione:

* Sezione di INTRODUZIONE: vi è presente una descrizione dell’esigenza di progettare il sistema nel miglior modo per poter facilitare le operazioni degli utenti, seguita dal contesto di utilizzo del sistema per poi presentarne gli obiettivi principali. Importante è anche la presenza di un elenco di definizioni, acronimi e abbreviazioni usato per facilitare la comprensione dei concetti citati al lettore.
* Sezione sull’ ARCHITETTURA SOFTWARE CORRENTE: mostra le tecnologie e le idee di progettazione prese in considerazione per lo sviluppo di Glitch.
* Sezione sull’ ARCHITETTURA SOFTWARE PROPOSTA: in primo luogo si mostra una panoramica sull’idea di base di come il sistema dovrebbe essere progettato. Ne segue la presentazione dei vari sottosistemi in cui il sistema è stato decomposto. Si continua con la descrizione del mapping effettuato sia sulle componenti hardware sia sulle componenti software, seguito dal modo in cui sono stati gestiti i dati persistenti. Le altre due sessioni successive spiegano il controllo degli accessi e della sicurezza, nonché quello globale del software. Infine, vi è una presentazione delle condizioni di confine esterne al sistema.
* Sezione sui SERVIZI DEL SOTTOSISTEMA: descrive i servizi forniti da ciascun sistema in termini di operazioni.
* Sezione di GLOSSARIO: in conclusione del SDD per specificare i termini utilizzati nel documento così da evitare ambiguità.

1. **Architettura software corrente**

Nella progettazione del sistema i progettisti non si nono potuti avvalere di un sistema preesistente, in quanto il cliente non disponeva di alcun sistema di gestione e vendita prodotti on-line.

Quindi si è proceduto con un’implementazione di tipo “Greenfield Engineering”, ovvero un’implementazione da zero volta a soddisfare le esigenze del cliente, prendendo anche spunto da siti simili che aiuteranno a implementare e migliorare Glitch cercando di tenere sempre presenti i requisiti.

I siti considerati come suggerimenti si possono individuare ai seguenti link: [www.amazon.it](http://www.amazon.it), esempio di sito di e-commerce generico, e [www.gamestop.it](http://www.gamestop.it) , come sito di vendita console e videogiochi.

1. **Architettura software proposta**
   1. **.** **Panoramica**

Il sistema commissionato è un software distribuito costituito da un nucleo centrale, contenente tutti i dati relativi agli utenti registrati, ai prodotti di cui il sito dispone, alle e-mail di assistenza, alle offerte proposte e al carrello utente con i relativi ordini effettuati.

Gli utenti interagiscono con il sistema centrale da diverse postazioni client previste da Glitch.

I client non hanno alcuna possibilità di interagire tra di loro, pertanto, tutte le richieste dovranno sempre essere fatte al server centrale. Ne deriva che l’architettura più idonea al progetto è di tipo Client-Server.

Le varie tipologie di utenti previste sono:

* l’utente classico, che si registra al sito e può effettuare vari acquisti;
* il gestore account, che ha il compito di definire i ruoli di alcuni utenti rendendoli gestori delle varie funzionalità del sito (del catalogo, account o assistenza);
* il gestore catalogo, che è colui che si occupa di inserire/rimuovere i prodotti e le offerte in modo da tenere il catalogo sempre aggiornato;
* il gestore assistenza, che si occupa di rispondere alle e-mail di assistenza degli utenti in modo tale da poter aiutare e risolvere i vari problemi che gli utenti possono riscontrare.

* 1. **. Decomposizione del sottosistema**

Considerando i requisiti individuati per Glitch, il sistema è stato scomposto in una serie di sottolivelli raggruppabili in due livelli principali: il Client e il Server.

Per quanto riguarda il lato Client sono stati individuati i seguenti sottolivelli:

* “InterfacciaUtente” (che comprende l’interfaccia “Catalogo”) che interagisce con i sottosistemi “Carrello”, “AssistenzaClienti” e “PaginaPersonale”. A loro volta le interfacce “Carrello” e “PaginaPersonale” si interfacciano rispettivamente con “Pagamento” e “ModificaProfilo”;
* “InterfacciaGestore” che in base al ruolo ricoperto dal gestore si specifica in:
  + “InterfacciaGestoreAccount”, che interagisce con il sottosistema “GestioneAccount”;
  + “InterfacciaGestoreCatalogo”, che interagisce con i sottosistemi “GestioneProdotti” e “GestioneOfferte”;
  + “InterfacciaGestoreAssistenza”, che interagisce con il sottosistema “GestioneAssistenza”.

Per il lato Server vi sono i seguenti sottolivelli:

* “ControlAccount”, che si occupa di validare i dati in caso di registrazione, di gestirne le operazioni e autenticare l’utente informando il Client su che tipo di interfaccia principale deve mostrare in base ai dati inseriti;
* “ControlCarrello”, che gestisce tutte le operazioni collegate al carrello da parte del Client, nonché la gestione del sottosistema pagamento;
* “ControlCatalogo”, che gestisce tutte le validazioni e le operazioni collegate alla gestione della “vetrina virtuale” del sistema;
* “ControlEmail”, che gestisce l’invio di e-mail di assistenza e le relative risposte;
* “ControlDBMS”, che si occupa di interfacciare le richieste effettuate dai vari sottosistemi al sottosistema Database.

**3.3. Mappatura hardware/software**

La struttura hardware proposta e costituita da un server centrale e da client utenti che possono essere costituiti da computer portatili o fissi nei quali possono girare browser diversi fra loro. Al server si collegano i client ed il database, il tipo di funzionalità del client è determinata in fase di autenticazione controllando nel database il tipo di utente che corrisponde allo username inserito in quel client. Gli utenti dovranno effettuare richieste al server per eseguire le operazioni legate alle loro rispettive funzionalità. Il database conterrà i dati relativi agli account, ai prodotti, alle offerte e alle e-mail. Il client e il server saranno connessi tramite una rete che utilizzerà il protocollo TCP/IP.

**3.4. Gestione dei dati persistenti**

Il sistema Glitch si avvale dell’uso di un database di tipo relazionale, al quale si interfaccia tramite un DBMS. Questa scelta è stata ponderata data la sicurezza offerta da un DBMS di ultima generazione, assieme ad una maggiore affidabilità e garanzia di coerenza e facilità di gestione, nonché dalla velocità di accesso e trasmissione dei dati.

**3.5. Controllo e sicurezza degli accessi**

­La sicurezza ed il controllo dei dati del database è garantita tramite l’utilizzo di username e password per ogni singolo utente, che verranno richieste per ogni singolo accesso. Ogni tipo di utente avrà a disposizione diverse interfacce grafiche (nel caso il ruolo lo richieda), in modo che ogni utente possa accedere solo alle rispettive funzionalità che rientrano nella sua categoria di utenza.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ATTORE\OGGETTO** | Utente | Prodotto | Offerta | E-mail |
| Utente | MP, RP | A |  | CA |
| Gestore account | MP, RP, AR, RR |  |  |  |
| Gestore catalogo | MP, RP | I, R | I, R |  |
| Gestore assistenza | MP, RP |  |  | RA |

**Legenda:**

* **MP =** Modifica profilo
* **RP =** Rimuovi profilo
* **AR =** Aggiungi Ruolo
* **RR =** Rimuovi ruolo
* **A =** Acquistare
* **I =** Inserire
* **R =** Rimuovere
* **CA =** Chiedere assistenza
* **RA =** Rispondere assistenza

**3.6. Controllo globale del software**

Il sistema è caratterizzato da un portale accessibile da browser e da un WebServer, attivo 24h, che deve provvedere a gestire gli accessi concorrenti da parte dei vari utenti. Quando un utente si logga e sottomette i propri dati, vi è un accesso al database (query di interrogazione) che permette di controllare l’esistenza del soggetto.

Dopo la conferma, l’utente può accedere a diverse operazioni messe a disposizione dal sistema in base al suo ruolo. L’utente base può accedere al sistema, visualizzare il catalogo, acquistare prodotti e modificare/rimuovere il proprio profilo.

I gestori, in aggiunta alle operazioni precedenti, gestiscono delle funzionalità di Glitch in base ai propri ruoli (chi gestisce gli account, chi il catalogo e chi l’assistenza).

Nella maggior parte delle operazioni effettuate dagli utenti si accede al DB in scrittura (si pensi alla registrazione o all’inserimento prodotto), mentre per le semplici visualizzazioni di dati ciò non è necessario.

L’intero sistema presente una struttura centralizzata, in quanto tutti i dati vengono gestiti all’ interno di una singola componente: il server.

**3.7. Condizioni limite**