**Immagine che contiene interni

Descrizione generata automaticamente**

**Documentazione System Design**

**“GLITCH”**

**Progetto presentato da:**

Annunziata Elefante

Ferdinando Napolitano

Santolo Mutone

**INDICE**

1. **Introduzione . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3**
   1. Scopo del sistema . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
   2. Obiettivi di progettazione . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
   3. Definizioni, acronimi e abbreviazioni . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
   4. Panoramica . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
2. **Architettura software corrente . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5**
3. **Architettura software proposta . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5**
   1. Panoramica . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .5
   2. Decomposizione del sottosistema . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
   3. Mappatura hardware/software . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
   4. Gestione dei dati persistenti . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .8
   5. Controllo e sicurezza degli accessi. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
   6. Controllo globale del software . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .12
   7. Condizioni limite . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
4. **Servizi del sottosistema . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12**
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, garantendo all’utente una maggior *usabilità*.

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, ciò per garantirne l’*affidabilità*.

Il linguaggio di programmazione scelto per il suo sviluppo è Java, con paradigma di programmazione ad oggetti, che consente l’inserimento di commenti che ne semplificheranno la *sostenibilità* in caso di modifiche future.

* 1. **Definizioni, acronimi e abbreviazioni**
* **SDD**: System Design Document
* **DBMS**: DataBase Management System
* **DB**: DataBase
* **GREENFIELD ENGINEERING**: Tipologia di sviluppo che comincia da zero, non esiste nessun sistema a priori e i requisiti sono ottenuti dall’utente finale e dal cliente. Nasce, perciò, a partire dai bisogni dell’utente.
  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 proposto 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 *livello 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”;
* per la gestione del sito vi sono tre diverse interfacce, una per ogni specifico ruolo:
  + “InterfacciaGestoreAccount”, che interagisce con il sottosistema “GestioneRuoli”;
  + “InterfacciaGestoreCatalogo”, che interagisce con i sottosistemi “GestioneProdotti” e “GestioneOfferte”;
  + “InterfacciaGestoreAssistenza”, che interagisce con il sottosistema “GestioneAssistenza”.

Per il *livello Server* vi sono i seguenti sottolivelli:

* “ControlAccount”, che si occupa di validare i dati in caso di registrazione, di gestirne le operazioni connesse e autenticare l’utente informando il Client su che tipo di interfaccia principale deve mostrare in base ai dati inseriti;
* “ControlRuoli”, che gestisce l’assegnazione o l’eliminazione di un ruolo ad un utente;
* “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.

**3.2.1. Decomposizione in sottosistemi**

Immagine che contiene testo, mappa

Descrizione generata automaticamente

**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.

**3.3.1. Diagramma di Deployment**

Immagine che contiene screenshot

Descrizione generata automaticamente

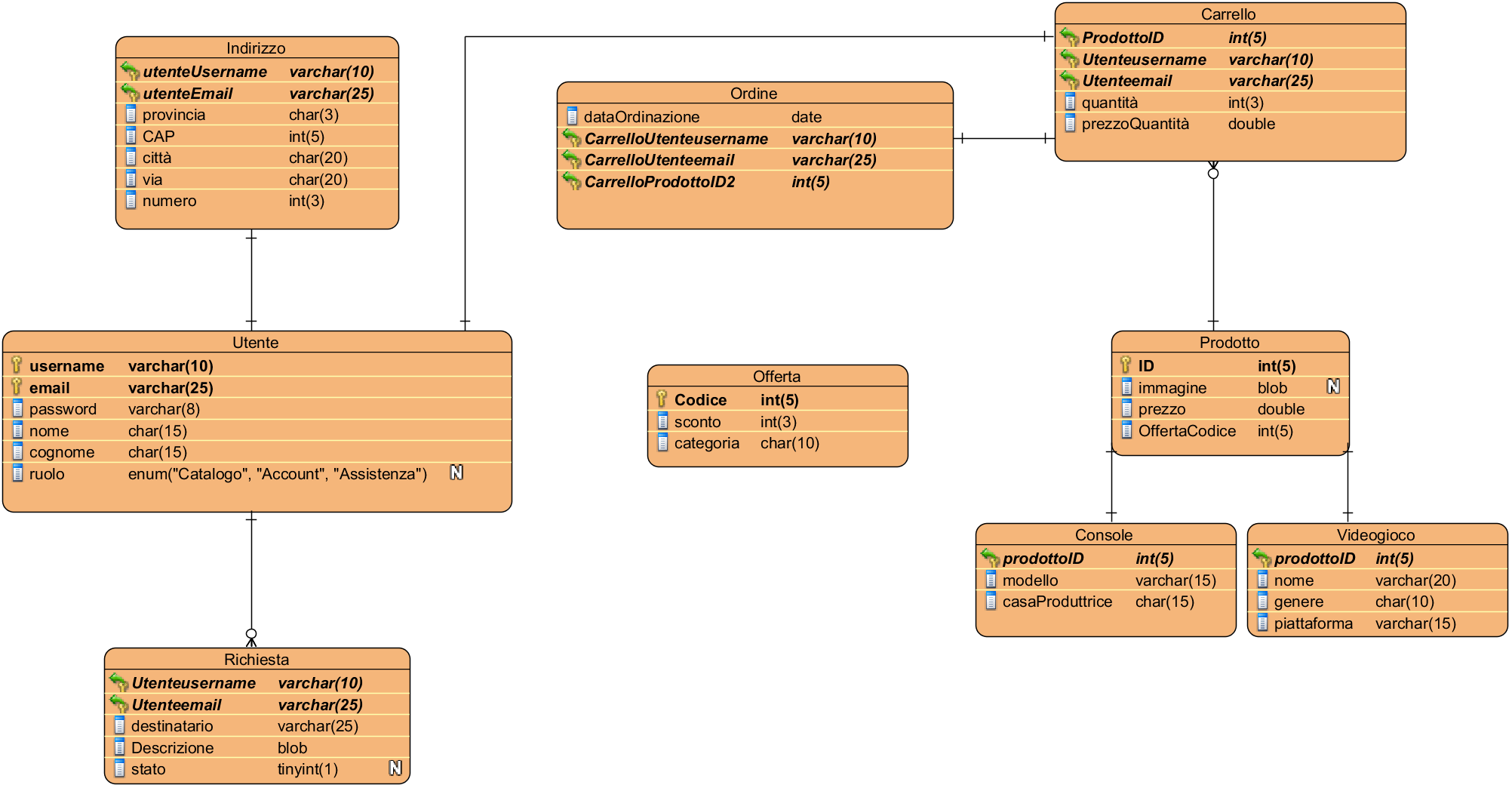
**3.4. Gestione dei dati persistenti**

Il sistema Glitch si avvale dell’uso di un database di tipo relazionale, al quale si interfaccia tramite un DBMS (in questo caso MySQL). 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.

**3.4.1. Schema priorità dei dati**

|  |  |  |
| --- | --- | --- |
| **Nome indicatore** | **Dati** | **Priorità** |
| **Credenziali** | * username * password | Urgente |
| **Dati personali** | * nome * cognome * e-mail * indirizzo * ruolo | Urgente |
| **Dati prodotti e ordini** | * nome * genere * piattaforma * modello * casa produttrice * immagine * prezzo | Urgente |
| **Dati offerta** | * nome * percentuale * categoria | Urgente |
| **Dati pagamento** | * intestatario * numero identificativo * scadenza * CVV | Consigliata |
| **Dati assistenza** | * destinatario * oggetto * descrizione * stato | Urgente |

**3.4.2. Schema E-R**

****

**3.4.2.1.** Utente

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipo** | **Null** | **Key** |
| username | varchar(10) | not null | primary key |
| e-mail | varchar(25) | not null | primary key |
| password | varchar(8) | not null |  |
| nome | char(15) | not null |  |
| cognome | char(15) | not null |  |
| ruolo | enum(“Catalogo”, “Account”, “Assistenza”) | nullable |  |

**3.4.2.2.** Indirizzo

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipo** | **Null** | **Key** |
| utenteUsername | varchar(10) | not null | external key |
| utenteEmail | varchar(25) | not null | external key |
| provincia | char(3) | not null |  |
| CAP | int(5) | not null |  |
| città | char(20) | not null |  |
| via | char(20) | not null |  |
| numero | int(3) | not null |  |

**3.4.2.3.** Richiesta

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipo** | **Null** | **Key** |
| utenteUsername | varchar(10) | not null | external key |
| utenteEmail | varchar(25) | not null | external key |
| destinatario | varchar(25) | not null |  |
| descrizione | blob | not null |  |
| stato | tinyint(1) | not null |  |

**3.4.2.4.** Ordine

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipo** | **Null** | **Key** |
| carrelloUtenteUsername | varchar(10) | not null | external key |
| carrelloUtenteEmail | varchar(25) | not null | external key |
| carrelloProdottoID | int(5) | not null | external key |
| dataOrdinazione | date | not null |  |

**3.4.2.5.** Offerta

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipo** | **Null** | **Key** |
| codice | int(5) | not null | primary key |
| sconto | int(3) | not null |  |
| categoria | char(10) | not null |  |

**3.4.2.6.** Carrello

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipo** | **Null** | **Key** |
| prodottoID | int(5) | not null | external key |
| utenteUsername | varchar(10) | not null | external key |
| utenteEmail | varchar(25) | not null | external key |
| quantità | int(3) | not null |  |
| prezzoQuantità | double | not null |  |

**3.4.2.7.** Prodotto

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipo** | **Null** | **Key** |
| ID | int(5) | not null | primary key |
| immagine | blob | nullable |  |
| prezzo | double | not null |  |
| offertaCodice | int(5) | not null |  |

**3.4.2.8.** Console

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipo** | **Null** | **Key** |
| prodottoID | int(5) | not null | external key |
| modello | varchar(15) | not null |  |
| casaProduttrice | int(5) | not null |  |

**3.4.2.9.** Videogioco

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Tipo** | **Null** | **Key** |
| prodottoID | int(5) | not null | external key |
| nome | varchar(20) | not null |  |
| genere | char(10) | not null |  |
| piattaforma | varchar(15) | not null |  |

**3.5. Controllo e sicurezza degli accessi**

­La sicurezza ed il controllo dei dati del database è garantita mediante 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 funzionalità che rientrano nella sua categoria di utenza.

**3.5.1. Tabella accessi**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sottosistema**  **Attore** | **Gestione** | | | | | |
| **Accesso** | **Account** | **Acquisto** | **Prodotti** | **Ruoli** | **Assistenza** |
| **Utente** | R, Li, Lo | MP, RP | A |  |  |  |
| **Gestore account** | R, Li, Lo | MP, RP | A |  | AR, RR |  |
| **Gestore catalogo** | R, Li, Lo | MP, RP | A | IP, RP, IO, RO |  |  |
| **Gestore assistenza** | R, Li, Lo | MP, RP | A |  |  | CA, RA |

**Legenda:**

* **R =** Registrazione
* **Li – Lo =** Login - Logout
* **MP =** Modificare profilo
* **RP =** Rimuovere profilo
* **AR =** Aggiungere Ruolo
* **RR =** Rimuovere ruolo
* **A =** Acquistare (comprende tutte le operazioni annesse al carrello)
* **IP =** Inserire prodotto
* **RP =** Rimuovere prodotto
* **CA =** Chiedere assistenza
* **RA =** Rispondere assistenza
* **IO =** Inserire offerta
* **RO =** Rimuovere offerta

**3.6. Controllo globale del software**

Il sistema è caratterizzato da un portale accessibile da browser e da un WebServer attivo 24/7, 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**

**3.7.1. Inizializzazione**

Per l’inizializzazione del sistema Glitch è necessario l'avvio di un web server che fornisca il servizio di un Database MySQL per la gestione dei dati persistenti e l’interpretazione ed esecuzione del codice lato server. In seguito, tramite l'interfaccia di Login, sarà possibile autenticarsi tramite opportune credenziali (username e password) come utente che, in base al suo ruolo, ha pieno accesso a tutte le funzionalità del sistema.

Una volta effettuato l'accesso, Glitch presenterà all'utente la sua homepage, dalla quale si possono effettuare tutte le operazioni che il sistema fornisce.

**3.7.2. Terminazione**

Al momento della chiusura del sito si ha la terminazione del sistema con un regolare Logout. Viene assicurata la consistenza dei dati, annullando eventuali operazioni che erano in esecuzione.

**3.7.3. Fallimento**

Il sistema lato server può fallire solo a causa di condizioni eccezionali quali mancanza di elettricità o guasti all’hardware (hard disk danneggiato ecc., che danneggiano permanentemente il sistema) o in caso di crash di sistema (attacchi al server dall’esterno). Il sistema lato client può fallire a causa di guasti temporanei (hardware o software), o per la caduta della linea telefonica, ma ciò non influisce in alcun modo sul lato server (che resterà comunque stabile), neanche se ci fosse stata una connessione attiva al DB, visto che il DBMS gestisce le transazioni. Per recuperare da un fallimento basta riaccendere il web server, ma ovviamente, nel frattempo, i client non possono usufruire nel sistema visto che il server è down.

**3.7.4. Eccezioni**

Il sistema lato client può lanciare delle eccezioni mirate in caso di operazioni errate dell’utente (login errato) o non consentitegli (accedere al carrello se non si è registrati), in questo modo non vi è un crash della pagina.

1. **Servizi del sottosistema**
2. **Glossario**

**Browser:** programma per scorrere velocemente una banca dati, in particolare per la navigazione su Internet.

**Web server:** un’applicazione software che, in esecuzione su un server, è in grado di gestire le richieste di trasferimento di pagine web di un client, ovverosia di un browser.

**Database:** i database sono sistemi di gestione dei dati elettronici logicamente strutturati, che con l’aiuto di un sistema di gestione del database regolano appartenenze e diritti di accesso e archiviano le informazioni riguardo alla base di dati contenuta. La maggior parte dei database si possono aprire, modificare e leggere soltanto con particolari applicazioni per database. Tra i software più diffusi per la gestione di database è annoverato **MySQL**.