Sistem Design Document (SDD)

“Gioco Mania”

# 1- Introduzione 1.1- Scopo del sistema

**Lo scopo del sistema è quello di progettare un sito Ecommerce operante nel settore della vendita di videogame, intraprendendo un’azione di marketing altamente aggressiva, permettendo al cliente di accedere al sistema da un qualsiasi browser, come semplice navigatore per ottenere informazioni sui tempi e costi di consegna, sulle modalità di pagamento e sui punti vendita più vicini al loro domicilio.**

**Il cliente può effettuare richieste su tutto il territorio italiano.**

**Si vuole invece che gli amministratori possano avere diretto contatto con il database del negozio per poter consultare il catalogo dei prodotti disponibili presso il magazzino principale ed acquistare prodotti per conto dei propri clienti che si recano presso i loro punti di accedere al sistema e di localizzare il punto di vendita più vicino al suo domicilio, e di recarsi ad esso per l’eventuale acquisto di un prodotto.**

# 1.2- Obiettivi del design

**Possiamo selezionare gli obiettivi di design da una lista di qualità desiderabili, mediante criteri suddivisi in cinque gruppi:**

**Criteri Performance, Affidabilità, Costi, Mantenimento, Criteri End User.**

**Criteri di Performance**

***Tempo di risposta:***

**Il sistema risponderà alle richieste impartite dai componenti in tempo reale.**

**Se la pagina dei prodotti ricercati da un componente è di grandi dimensioni, il sistema la visualizzerà con leggero ritardo (impiegherà qualche attimo in più) rispetto ad una pagina contenente pochi prodotti.**

***Throughput:***

**Il cliente effettuerà l’acquisto in poco tempo, semplicemente selezionando il prodotto richiesto da un componente.**

***Memoria:***

**La dimensione della memoria è *dinamica*, in quanto dipende dalla grandezza del DataBase.**

## Criteri di Affidabilità

***Robustezza:***

**Eventuali richieste di prodotti e/o iscrizione a Gioco Mania esterne al territorio nazionale, verranno rifiutate per motivi logistici e relativi ai costi gestionali.**

**Inoltre prima della reale iscrizione come utente registrato vi sarà una attenta verifica di attendibilità dei dati immessi dall’utente (controllo del codice fiscale, esistenza reale della persona fisica, ecc…).**

***Attendibilità:***

**I risultati prodotti dalle pagine dinamiche (le servlet) riguardo la disponibilità o meno di prodotti in magazzino sono attendibili nel senso che rispecchiano istante per istante la reale situazione del magazzino stesso; non si deve mai verificare che venga visualizzata la disponibilità di un determinato prodotto se questo non c’è effettivamente.**

***Disponibilità:***

**Una volta che il sistema è stato realizzato sarà disponibile ogni qualvolta che il cliente e/o l’admin ne richiederà l’utilizzo (a meno di guasti temporanei).**

***Sicurezza:***

**La sicurezza è garantita nei limiti da una *login* e di *password*  non criptata.**

## Criteri di Mantenimento

***Estendibilità:***

***E'* consentito, in quanto è possibile aggiungere in futuro, nuove funzionalità al sistema (ex:come estendere il campo d’azione al di fuori del territorio nazionale) oppure creare nuove classi, con l’estensione di quelle già esistenti.**

***Modificabilità:***

**Il sistema in futuro potrà estendere il proprio campo di azione in tutta Europa oppure sbilanciarsi persino in tutto il mondo.**

**Quindi l’internazionalizzazione, ossia la capacità di cambiare il sistema per aggiungere convenzioni internazionali (come linguaggi, unità di misura e formati numerici) al sito, potrà essere presa in considerazione in un successivo momento. Il sistema è gestito da un server.**

***Adattabilità:***

**Il sistema può essere facilmente portato su diversi domini di applicazione (multipiattaforma).**

***Portabilità:***

**Tutto il software relativamente associato al sistema sarà scritto usando il linguaggio di programmazione Java, per conformarsi alla politica corrente della compagnia.**

**Nessun vincolo è imposto dalla piattaforma hardware.**

**Quindi, in virtù della portabilità del linguaggio, l’intero sistema è portabile su differenti piattaforme.**

***Leggibilità:***

**Mediante la lettura del codice il sistema è comprensibile.**

**Infatti semplici istruzioni consentono di comprendere le funzionalità del sistema e con l’ausilio di commenti e documentazione il lavoro di comprensione è ulteriormente semplificato.**

***Tracciabilità dei requisiti:***

**In questa fase non siamo in grado di definire la facilità della mappatura del codice nei requisiti specifici, in virtù del fatto che non esiste ancora la stesura del codice.**

**Criteri End User:**

***Utilità:***

**Il lavoro dell’utente verrà supportato nel miglior modo possibile dal sistema.**

**Infatti l’utente compierà le operazioni ad egli consentite, senza aver il minimo problema. Il sistema lo seguirà passo passo.**

***Usabilità:***

**Il cliente può accedere al sistema facilmente.**

**Altrettanto facile sarà la visualizzazione dei punti vendita più vicini.**

**Quando questo ha l’intenzione di visualizzare i punti vendita, deve richiedere la pagina relativa alla loro visualizzazione, mediante un link.**

**Quando vuole acquistare uno o più giochi, gli viene richiesto di inserire i suoi dati personali (relativi alla sua ubicazione, o alla sua carta (se decide di pagare on-line)) all’interno di un form.**

**Quando l’utente è registrato, dopo aver inserito la sua login e la sua password, visualizza la pagina relativa al catalogo, avendo poi la possibilità di interrogare la pagina medesima, eventualmente acquistando dei prodotti.**

**Infine quest’ultimo ha la possibilità di dimettersi da tale ruolo (quello di utente registrato) premendo uno specifico pulsante.**

# 1.3- Definizioni, acronimi e abbreviazioni

**Acronimi :**

**RAD: Requirements Analysis Document**

**SDD: System Design Document**

**ODD: Object Design Document**

**DB: DataBase**

**DBMS: DataBase Management System**

***Definizioni***

***Amministratore*: Gestore ed organizzatore del sistema.**

***Utente*: Colui che svolge una o più delle attività offerte dal sistema.**

***Database*: Base di dati (DB), indica un archivio di dati o un insieme di archivi, in cui le informazioni in esso contenute sono strutturate e collegate tra loro secondo un particolare modello logico e in modo tale da consentire la gestione e/o organizzazione efficiente degli stessi dati e l’interfacciamento con le richieste dell’utente attraverso i cosiddetti “querylanguage”.**

# 1.4. References

**BerndBruegge& Allen H. Dutoit, Object-Oriented Software Engineering : Using UML,Patterns and Java, (2nd edition), Prentice-Hall, 2003.**

**Sommerville, Software Enginneering (6th edithion, 2001), Addison Wesely.**

**Materiale didattico integrativo**

# 1.5. Overview

**Tale documento lo si può considerare diviso in tre parti:**

1. **Introduzione: fornisce una rapida panoramica degli obiettivi progettuali.**
2. **Current System: fornisce una rappresentazione del sistema corrente.**
3. **Proposed Software architetture: fornisce una dettagliate descrizione dell’architettura del sistema software proposto e la sua decomposizione in sottosistemi.  
   Verranno inoltre analizzati alcuni aspetti essenziali come: gestione di dati persistenti, controllo degli accessi, condizioni di sicurezza.**

# 2-*Current System Architecture*

**Come esposto e documentato nei precedenti documenti, il sistema in questione, non andrà ad aggiornare, integrare o modificare altri progetti già sviluppati. L’architettura Software viene realizzata ed ideata a partire dall’analisi e lo studio dei requisiti identificati accuratamente nella documentazione precedente.  
Rifacendoci ai testi di riferimento ed al materiale didattico integrativo fornitoci dai docenti si è scelto di realizzare il sistema secondo il pattern “MVC”.**

3- Architettura proposta

3.1 Overview

**Attraverso l’analisi dei casi d’uso identificati nel documento precedente (R.A.D) si è scelto di utilizzare un’architettura software che permettesse di suddividere il sistema in tre sottosistemi.**

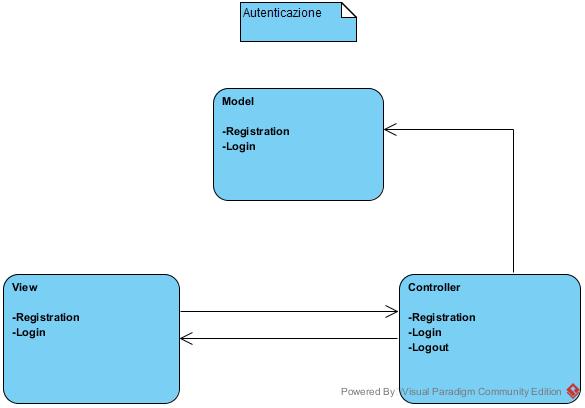
**L’architettura adoperata è identificata attraverso l’acronimo “MVC” ovvero “Model-View-Control**

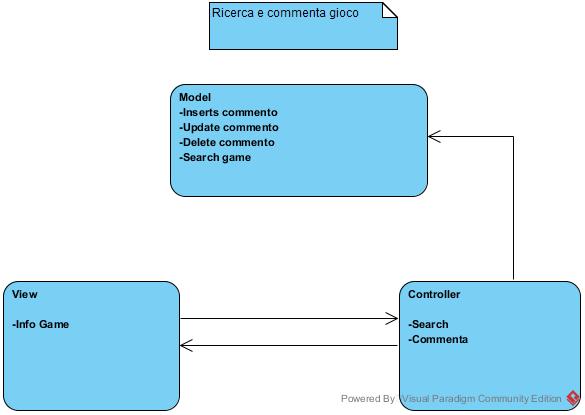
**La scelta di questa architettura software è stata effettuata considerando i vantaggi che essa offre in termini di “riusabilità” del codice;**

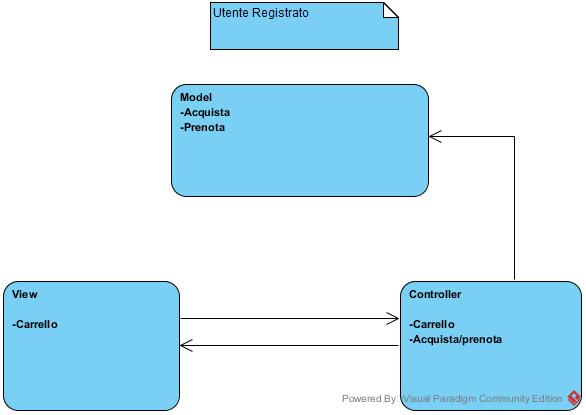
* **Il “Model” fornisce i metodi per accedere ai dati che verranno memorizzati nel Sistema.**
* **Il “View” visualizza i dati contenuti nel “Model” e si occupa dell’interazione tra utenti e sistema rappresenta quindi l’interfaccia grafica con la quale gli utenti accedono al sistema. Tale strato racchiude tutta la parte grafica che permette la rappresentazione dell’applicativo. Risulterà estremamente semplice un’eventuale modifica del “look and fell” del sistema agendo solo su questo sotto strato del sistema senza alterare i restanti due.**
* **Il “Controller” riceve i comandi dall’utente attraverso il “View” e gestisce il flusso dell’applicazione, fornendo una sorta di connessionetra “View” e “Model”. Tale strato conterrà esclusivamente la parte logica del sistema.**

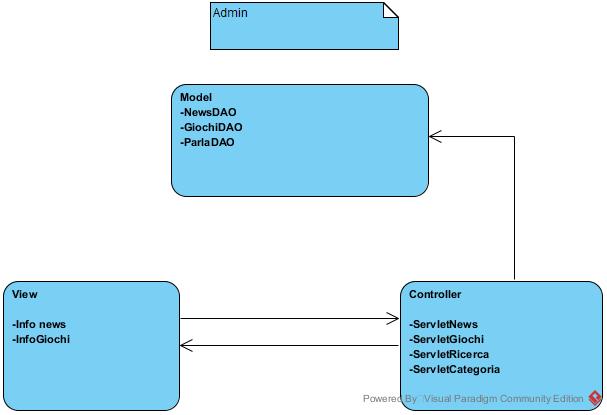
# 3.2- Decomposizione in sottosistemi

Logical view sottosi

stemi web







* **L’amministratore e l’utente registrato sono specifiche di utente (utente è la generalizzazione di admin e utente registrato). Di conseguenza le operazioni svolte dall’utente possono essere svolte da tutti gli attori**
* **L’amministratore inoltre interagisce con il model “parla” che serve per unire le News con i Giochi**

*Model*

**Il sottosistema “Model” è una rappresentazione dei dati persistenti presenti all’interno del Sistema. Dunque tale sottosistema si suddivide in:**

**-Giudizio**

**-Acquista**

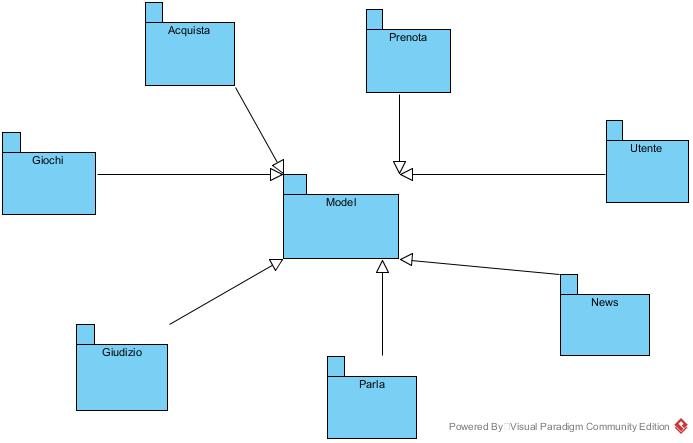
**-Prenota**

**-Utente**

**-News**

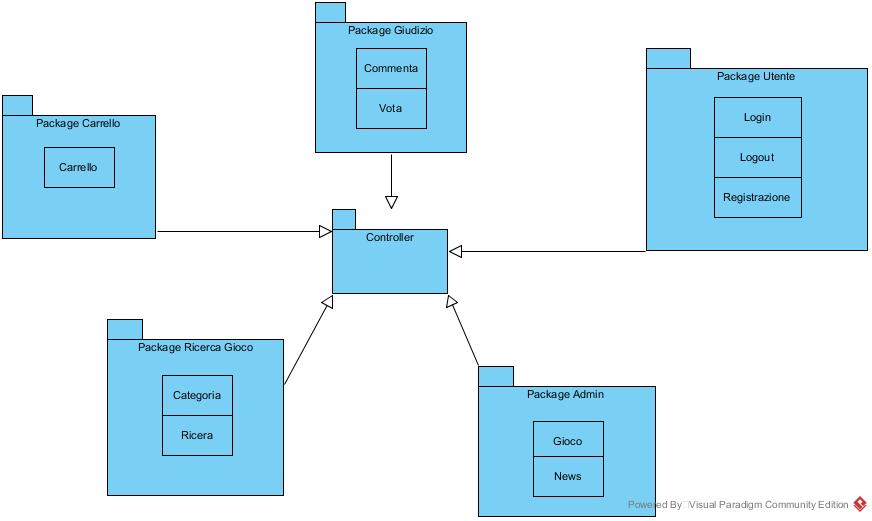
**-Parla**

**-Giochi**



*Controller e sotto-sistemi*

**La logica implementata permette di dividere il sistema in ulteriori sotto-sistemi basandosi sull’aggregazione delle funzionalità “simili” che il sistema dovrà svolgere. I compiti del software sono svolti dai controller e quindi la divisione del sistema è fata tramite package di controller**

****

# 3.3- Mapping Hardware/Software

**L’hardware necessario per il funzionamento del sistema in questione è costituito da due nodi fondamentali:**

* ***Client*: L’utente accederà al sistema attraverso l’uso di un computer il quale utilizzerà un web-browser.**
* ***Server*: Conterrà il web server su cui sarà configurato il sistema, ed un DBMS per la gestione dei dati persistenti.**

**Il collegamento client-server avverrà attraverso una richiesta http da parte del client, a tale richiesta seguirà una pagina di autenticazione nella quale verrà identificato il tipo di utente (Utente registrato o Amministratore).**

**Per eseguire operazioni sul server, i client dovranno effettuare determinate richieste.**

**I dati persistenti risiederanno su di un database, il quale comunicherà con il server attraverso l’uso di un DBMS.**

**Il protocollo di comunicazione tra client e server sarà TCP/IP.**

**Per usufruire dei servizi offerti dal server, il client dovrà avere un web-browser che supporti le tecnologie web di base (html, java, javascript, css).**

**Non sono richiesti particolari requisiti hardware da parte del client; la velocità di caricamento delle pagine è strettamente legata alla banda a disposizione.**

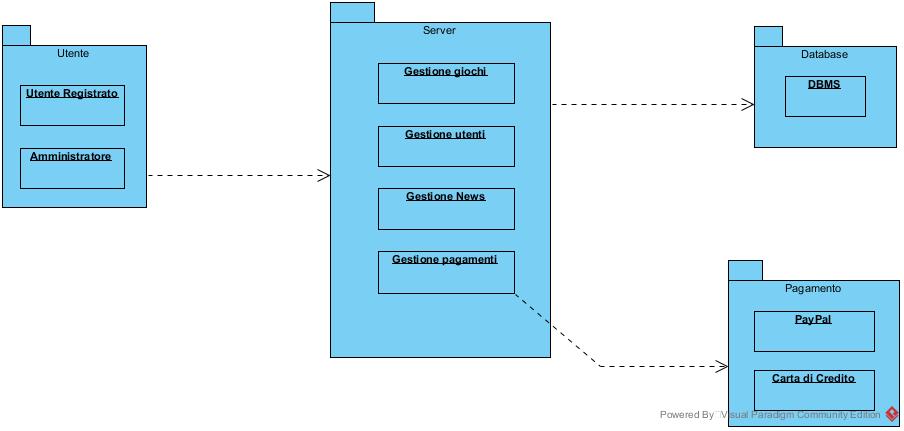
**Sul server verrà installato il sistema; esso conterrà fisicamente il database ed i codici sorgenti.**

**Per il sistema, basato su un’architettura distribuita client/server, sono state scelte questi tipi di configurazioni:**

**PIATTAFORMA: Qualsiasi che supporta un browser java compatibile (client)**

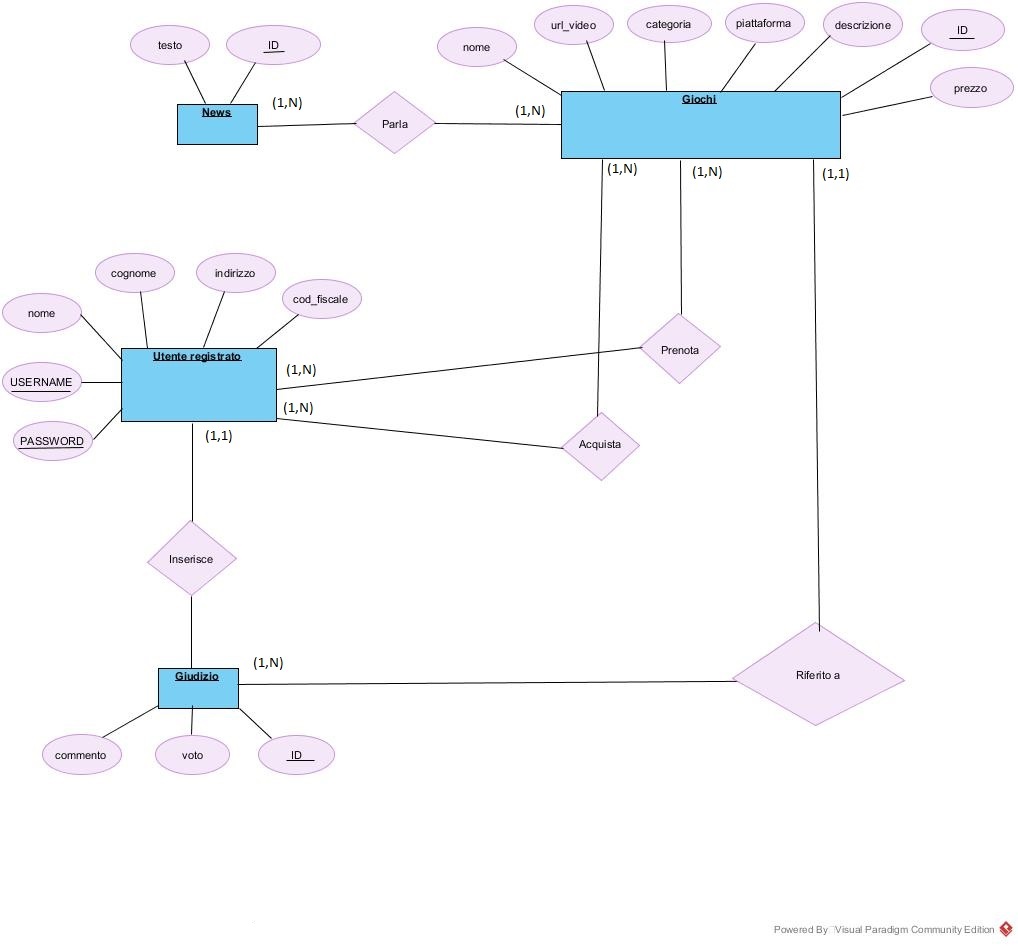
**COMPUTAZIONE: Monoprocessore (server) almeno un i586(pentium) intel**

**CONNETTIVITA: Tasso di trasmissione standard (minimo 56 kbps) protocollo: HTTP,TCP/IP**

****

# 3.4- *Persistente data management*

# 3.5- Access control and security

*Matrice degli accessi*

**Gli attori possono accedere alle funzionalità del sistema attraverso dei diritti di accesso regolati secondo le politiche del sistema e alla tipologia di utenza.**

**Di seguito sono documentate i diritti degli attori del sistema e per ogni tipo di oggetto sono documentate le operazioni che gli attori possono svolgere su di essi.**

|  |  |  |
| --- | --- | --- |
|  | | Utenti |
| Amministratore | Elimina utente  Visualizza lista completa degli utenti  Visualizza informazioni singolo utente Contatta utenti | |
| Utente | Visualizza le informazioni personali  Contatta amministratore  Modifica dati personali | |

|  |  |  |
| --- | --- | --- |
|  | | Giochi |
| Amministratore | Inserisci gioco  Modifica gioco  Elimina gioco  Visualizza elenco giochi | |
| Utente | Visualizza informazioni del gioco  Prenotazione di un gioco  Acquisto di un gioco  Inserisce giudizio al gioco | |

|  |  |  |
| --- | --- | --- |
|  | | News |
| Amministratore | Inserisci una news  Modifica una news  Visualizza l’elenco delle news | |
| Utente | Visualizza l’elenco delle news | |

|  |  |  |
| --- | --- | --- |
|  | | Pagamenti |
| Amministratore | Visualizza l’elenco dei pagamenti | |
| Utente | Effettua Pagamento  Visualizza pagamenti  Visualizza storico pagamento | |

*3.5.2 Sicurezza*

**Il sistema sarà protetto da accessi non autorizzati attraverso un meccanismo di autenticazione che farà uso di un username e una password, per l’utente, impostati al momento della registrazione al sistema e per gli amministratori impostati a priori.**

**Sottomessa la form con username e password il sistema processerà i dati passati verificando che esista un “record” all’interno del database relativo alle credenziali inserite, nel caso non esista tale record, verrà segnalata una condizione d’errore e riproposto l’inserimento delle credenziali.**

**Il sistema distinguerà automaticamente il tipo di utente e le relative operazione permesse.**

# 3.6- Flusso di controllo globale

**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 di utenti e amministratori. Quando un amministratore 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’admin può accedere a diverse operazioni messe a disposizione dal sistema. Ogni operazione (ad eccezione dell’acquisto) è indipendente dalle altre ed è attivabile dalla pressione di un bottone (submit).**

**VisualizzazioneProdotto consiste di una ricerca di un singolo prodotto o di una classe di prodotti simili nel tipo o nel prezzo; poiché tale accesso è in sola lettura non c’è bisogno della gestione della concorrenza (infatti non vi è modifica di dati).**

**AcquistaProdotto richiede un flusso di operazioni molto più complesso.**

**Per prima cosa vi è la chiamata allo UseCase ViualizzazioneProdotto, in modo da ricercare ciò a cui si è davvero interessati.**

**Prima di confermare l’acquisto si entra in una sezione critica ove il DBMS lato server gestisce la concorrenza di più utenti, per evitare che vi siano accavallamenti di decremento di disponibilità (se per esempio due utenti acquistano contemporaneamente lo stesso prodotto).**

**Per quanto riguarda lo UseCase ModificaDatiUtente e ModificaDatiGioco non c’è gestione di concorrenza, in quanto ogni utente si trova nella propria sessione ed il sistema accede ad DB modificando esclusivamente i dati della persona in questione.**

**L’utente non registrato invece può accedere al sistema solo per visualizzare i punti vendita o per iscriversi.**

**In entrambi i casi non si accede al DB in scrittura.**

# 3.7- Condizioni limite

* + 1. ***Inizializzazione***

**Al momento dell’avvio del sistema, esso leggerà un file di configurazione contenente l’indirizzo IP della macchina dove risiede il database e dove è installato il DBMS.**

* + 1. ***Terminazione***

**Il sistema sarà sempre accessibile in quanto il server che lo ospita non sarà mai off-line.**

* + 1. ***Fallimento***

**Se il server dove risiede il sistema dovesse arrestarsi o ci fossero problemi legati alla rete, il sistema verrebbe bloccato dopo aver salvato lo stato al momento del malfunzionamento. Un ipotetico malfunzionamento ad una macchina client non comprometterebbe minimamente il funzionamento del sistema o di altri client.**