Università degli Studi di Salerno

Corso di Ingegneria del Software

GameVault

System Design Document

Team di Sviluppo:

Nome	Matricola
Corsini Ernesto	0512114376
Orsatti Alessandro	0512114982
Palma Claudio Giuseppe	0512114151

Scritto da:	Claudio Giuseppe Palma

QINDICE

1. INTRODUZIONE

1.1 SCOPO DEL SISTEMA

Lo scopo del sistema è fornire un e-commerce classico con tutte le principali funzionalità offerte dai maggiori competitor.

1.2 OBBIETTIVI DI PROGETTAZIONE

Definizione delle priorità:

1) **Priorità Alta:** funzionalità essenziale che deve essere inclusa nel sistema software fin dall'inizio della prima release;
2) **Priorità Media:** funzionalità che può essere implementata nel sistema software in una futura release;
3) **Priorità Bassa:** funzionalità opzionale che può anche non essere implementata.

1.2.1 CRITERI DI PRESTAZIONE

- **Throughput**

  Il sistema deve essere in grado di gestire almeno 100 utenti contemporaneamente, dunque deve avere un sistema completamente scalabile.

  Priorità: Bassa

- **Memoria**

  Tutti i dati relativi al catalogo, agli ordini effettuati e agli utenti registrati sono conservati in un database relazionale. Lo spazio occupato dal sistema dipende di conseguenza dalle dimensioni di quest'ultimo.

  Priorità: Alta

- **Tempo di Risposta**

  Il sistema deve garantire un tempo di risposta massimo agli input dell'utente di 3 secondi. Per fare ciò ci affideremo a meccanismi di compressione e

decompressione dei dati del server container utilizzato e quelli di default dei browser. Poi in un secondo momento e in seguito ad un'analisi prestazionale verrà preso in considerazione l'uso di tool e librerie esterne per i suddetti meccanismi.
Priorità: Bassa

1.2.2 CRITERI DI AFFIDABILITÀ

- **Robustezza del sistema**
  Il sistema deve, così da evitare utilizzi improri e pericolosi delle proprie funzionalità, effettuare controlli sui dati inseriti dall'utente.
  Priorità: Alta
- **Attendibilità**
  Il sistema deve garantire l'attendibilità dei servizi messi a disposizione(Un ordine potrà essere effettuato solo nel momento in cui è disponibile, altrimenti verra notificato direttamente sul sito che quel prodotto non è più disponibile, non permettendo quindi l'acquisto).
  Priorità: Alta
- **Sicurezza**
  Il sistema deve garantire l'accesso alle proprie componenti esclusivamente agli utenti autorizzati(Le sezioni di gestione del sito non devono assolutamente essere accedute da utenti sprovvisti di un account da Admin).
  Priorità: Alta

1.2.3 CRITERI DI SUPPORTABILITÀ

- **Modello Three-Tier**
  Il sistema deve avere un'architettura a tre livelli che favorisca la manutenibilità. L'idea è quella di usare un'architettura Three-Tier che organizza le applicazioni in tre tier di calcolo logici e fisici: il tier presentazione, o interfaccia utente, il tier applicazione, dove i dati vengono elaborati, e il tier dati, dove i dati associati all'applicazione vengono memorizzati e gestiti.
  Priorità: Alta
- **Estensibilità**
  Il sistema deve essere facilmente estensibile per permettere l'aggiunta di funzionalità in futuro ma anche per assicurare la possibilità di un lancio dell'applicazione in fase intermedia, priva dunque di tutte le funzionalità progettate ma già interamente utilizzabile per il suo scopo primario. Ciò viene assicurato dall'utilizzo dell'architettura Three-Tier e da una componente di paradigma object-oriented.
  Priorità: Alta

1.2.4 CRITERI DI USABILITÀ

- **User-Friendly**

  Il sistema deve essere user-friendly per garantire un'esperienza utente piacevole ed intuitiva. Ciò verrà garantito tramite la GUI grazie alla presenza di una barra di navigazione e di icone familiari ed intuitive per servizi come il carrello e l'area personale.

  Il sistema inoltre deve aiutare l'utente nella navigazione e nell'utilizzo dei servizi offerti dall'applicazione. Verrà continuamente assistito tramite messaggi di errore che lo guideranno verso l'uso corretto dell'applicazione e nella corretta compilazione dei vari form quando necesssario.
  Priorità: Alta

- **Responsive**

  Il sistema deve essere utilizzabile dagli utenti tramite qualsiasi browser e da qualsiasi dispositivo connesso ad una rete internet. Ciò viene assicurato dal deployment dell'applicazione su un server container e dalle caratteristiche responsive che verranno ottenute tramite l'uso di media query.
  Priorità: Media

1.2 RIFERIMENTI

- **R.A.D.** : si farà riferimento ai requisiti non funzionali presenti nel R.A.D.

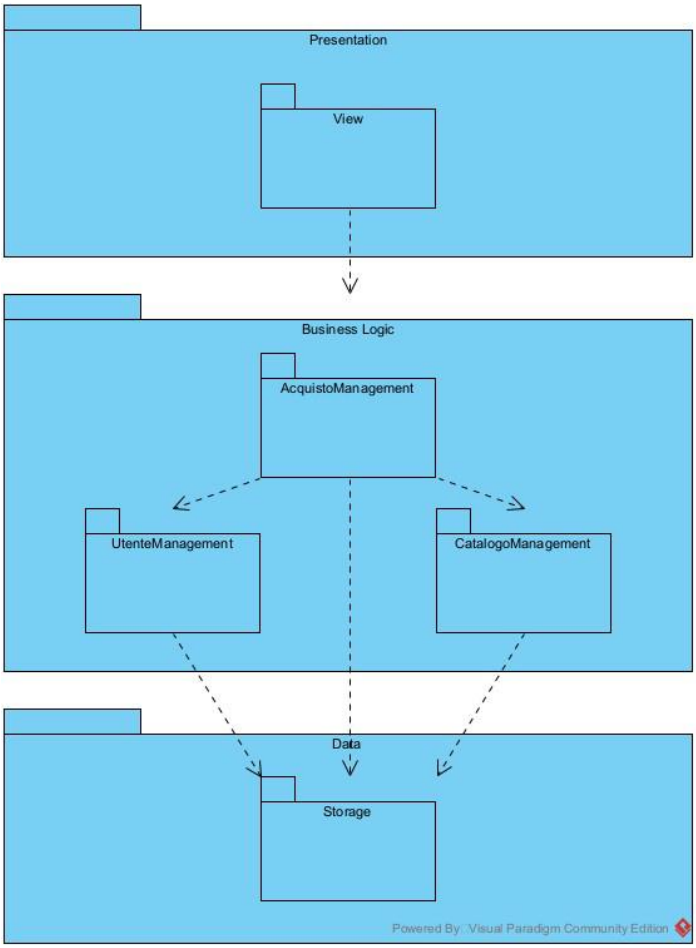
2. ARCHITETTURA DEL SISTEMA PROPOSTO

2.1 DECOMPOSIZIONE IN SOTTOSISTEMI

Il sistema risulta principalmente suddiviso in tre tier: Presentation, Business Logic e Data (suddivisione che viene ereditata dall'architettura Three-Tier).

Ognuno di questi layer è formato da uno o più sottosistemi generalmente caratterizzati da una forte coesione ma debolmente accoppiati tra loro.

Il seguente diagramma UML mostra un overview dei sottosistemi e delle loro relazioni:



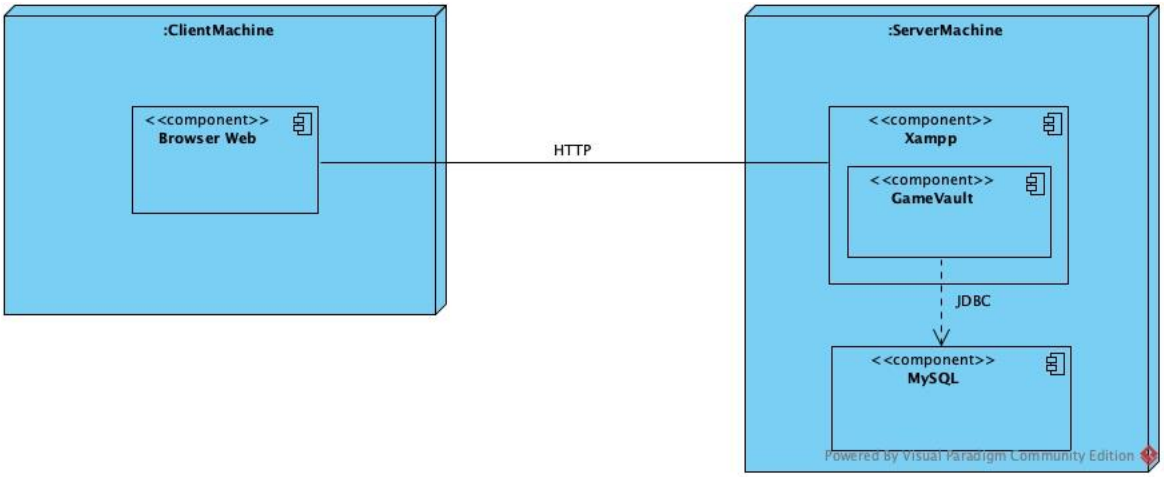
- **View:** è il sottosistema che si occupa di gestire ciò che l'utente visualizza su schermo.
- **AcquistoManagment:** è il sottosistema che si occupa di gestire la logica dietro l'acquisto di uno o più prodotti: pagamento, stato dell'ordine, carrello…
- **UtenteManagment:** è il sottosistema che si occupa di gestire la logica dietro l'autenticazione dell'utente e i suoi permessi.
- **CatalogoManagment:** è il sottosistema che si occupa di gestire il catalogo del sistema.
- **Storage:** è il sottosistema che si occupa dell'interazione con la base di dati. Per via della complessità richiesta, parte, o il totale, di queste operazioni saranno delegate ad un'istanza di un DBMS relazionale.

2.2 MAPPING HARDWARE/SOFTWARE

L'architettura del sistema è Client-Server con la gestione dei dati persistenti affidata ad un'istanza di un DBMS relazionale, quale MySQL, che si trova sulla stessa macchina del server.

La comunicazione tra il Browser, sulla ClientMachine, e il sistema, sulla HostMachine, avviene attraverso richieste e risposte HTTP.

Il deployment del sistema, impacchettato in un file war, viene fatto su di un server Xampp installato sulla HostMachine, che comunica con il DBMS attraverso un driver PDO(Php data Object).



2.3 GESTIONE DELLA PERSISTENZA

I dati del sistema che devono essere persistenti sono:

- Le informazioni degli utenti
- Le informazioni sui prodotti
- Le informazioni sul carrello
- Le informazioni sugli ordini

Tutto questo verrà delegato ad un'istanza di MySQL.

2.4 CONTROLLO DEGLI ACCESSI

Attori Oggetti	Utente non Registrato	Utente Registrato	Admin
Informazioni Utente	signUp()	login() logout() mostraDatiPersonali() modificaDatiPersonali()	login() logout()
Carrello		aggiungiAlCarrello() rimuoviDalCarrello() visualizzaCarrello() modificaQuantitaProdotto() paga()	
Ordine		visualizzaOrdini() creaOrdine()	visualizzaOrdini() filtraOrdini() aggiornaStatoOrdine()
Prodotto	visualizzaProdotto()	visualizzaProdotto()	visualizzaProdotto() modificaProdotto()
Catalogo	visualizzaProdotti() filtraProdotti() cercaProdotti()	visualizzaProdotti() filtraProdotti() cercaProdotti()	aggiungiProdotto() rimuoviProdotto()

2.5 CONTROLLO GLOBALE DEL SOFTWARE

Il controllo globale del software è di tipo event-based. Essendo un applicazione web, sarà il Web Server ad occuparsi dello smistamento delle varie richieste HTTP verso l' apposito server Xampp che si occuperanno di gestire la richiesta, interagire con le altre componenti del sistema ed elaborare una risposta.

2.6 CONDIZIONI LIMITE

Le boundary condition del sistema sono le seguenti:

- **Installazione del sistema:** l'installazione del sistema verrà eseguita da un sistemista ed un programmatore che avranno anche il compito di riempire la base di dati con gli account dei vari gestori ed il catalogo. In particolare, il server container Xampp, sul quale verrà fatto il deployment del sistema, verrà installato su di una macchina remota, per questo ci si affiderà ad un'azienda di web hosting.
- **Avvio del sistema:** il sistema verrà avviato da un sistemista in seguito all'installazione. Dopo l'avvio viene attivato il DBMS MySQL, la cui connessione col sistema verrà stabilita mediante driver PDO(Php data Object).

- **Manutenzione del sistema:** per l'applicazione di un aggiornamento al sistema si dovrà scegliere una finestra oraria(in genere 1-2 ore) nella quale il sistema sarà offline, saranno applicate le dovute modifiche e poi verrà eseguito nuovamente l'avvio. Queste operazioni verranno eseguite da un sistemista e un programmatore. La finestra oraria di riferimento sarà dalle ore 04:00 alle 05:00 UTC+1, fascia oraria dove si prevede la minor attività per il sistema(l'orario potrebbe cambiare una volta studiati i dati di attività del sistema, in genere dopo un anno di attività).
- **Fallimento del sistema:** nel caso di fallimento del sistema a causa di un problema relativo al servizio di hosting, sarà responsabilià del nostro team cercare di risolvere il problema quanto prima.
  Nel caso in cui il fallimento sia dovuta ad un'eccezione non gestita dal nostro sistema un sistemista lo riavvierà quanto prima per poi segnalare il programma al team di programmatori in modo da risolverlo il prima possibile.