INFINITY GAMES

System Design Document

Membri del Team

Falciano Ciro	0512103596
Schirosa Alessandro	0512104514
Ruggiero Simone	0512106074
De Lucia Antonino	0512105366

REVISION HISTORY

DATA	VERSIONE	DESCRIZIONE	AUTORE
30/11/20	1.0	Prima versione del documento.	C. Falciano, S. Ruggiero, A. Schirosa, A. De Lucia

SOMMARIO

1. INTRODUZIONE

1.1. Obiettivi del sistema

L'obiettivo del sistema è quello di consentire agli utenti di poter acquistare videogiochi e visualizzarli in una libreria personale, oltre che a ricercare prodotti nello store e aggiungerli alla lista dei desideri. L'interfaccia del sistema dovrà essere estremamente intuitiva, in modo da invogliare gli utenti ad effettuare gli acquisti.

1.2. Design Goals

ID	DESCRIZIONE	CATEGORIA	ORIGINE
DG_1 Tempo di risposta	Il sistema deve garantire un tempo di risposta quanto più breve possibile.	Performance	NFR_10
DG_2 Portabilità	Il sistema sarà responsive, in quanto il sito sarà sviluppato in modo da permettere la visualizzazione anche da tablet e smartphone.	Usabilità Interfaccia	NFR_1 NFR_2 NFR_21
DG_3 Usabilità	L'usabilità del sistema sarà data dall'interfaccia intuitiva, contenente pulsanti, titoli e messaggi di errori semplici da comprendere.	Usabilità	NFR_4
DG_4 Throughput	il sistema dovrà essere quanto più reattivo possibile, anche con grandi carichi di lavoro.	Performance	NFR_10

ID	DESCRIZIONE	CATEGORIA	ORIGINE
DG_5 Sicurezza	Il sistema consente l'accesso tramite autenticazione ed ogni utente può usufruire esclusivamente delle funzionalità associate alla propria categoria. Inoltre, il sistema deve tenere al sicuro i dati da chi non è autorizzato ad accedervi.	Affidabilità	NFR_5 NFR_8 NFR_9
DG_6 Memoria	Si utilizzerà il database relazionale MySQL per gestire la grande quantità di dati, pertanto il sistema non dovrà subire rallentamenti.	Performance	NFR_11 NFR_13
DG_7 Disponibilità	Il sistema dovrà essere perennemente usufruibile dagli utenti, fatta eccezione per le ore di manutenzione.	Affidabilità Implementazione	NFR_6 NFR_7 NFR_18
DG_8 Robustezza	Eventuali inserimenti non validi o mancanti da parte dell'utente devono essere segnalati tramite messaggi di errore.	Affidabilità	NFR_8
DG_9 Affidabilità	Il sistema garantirà una corretta gestione delle funzionalità.	Affidabilità	NFR_8 NFR_5 NFR_6

ID	DESCRIZIONE	CATEGORIA	ORIGINE
DG_10 Adattabilità	Il sistema dovrà adattarsi facilmente alle nuove necessità del cliente.	Interfaccia	NFR_21

1.3. Definizione, Acronimi e Abbreviazioni

RAD: Requirements Analysis Document.

SDD: System Design Document.
ODD: Object Design Document.
NFR: Requisiti non funzionali.

UC: Use case.
FR: Requisiti funzionali.
DG: Design Goal
MVC: Model View Controller
GUI: Graphic User Interface
DBMS: Database Management System
JSP: Java Server Page
HTTP: HyperText Transfer Protocol
Amministratore: Utente che si occupa di gestire la piattaforma e gli utenti.
Utente: Un qualsiasi utilizzatore della piattaforma.

1.4. Riferimenti

-Object-Oriented Software Engineering Using UML, Patterns, and Java, Bernd Bruegge & Allen H. Dutoit.

-Steam

-Infinity Games - RAD

1.5. Panoramica

Il documento si compone di una prima parte in cui vengono introdotti gli obiettivi di design.
Al secondo punto del documento viene presentata l'architettura del sistema corrente.

Al terzo punto viene presentata l'architettura del sistema proposto e nel dettaglio:

- La decomposizione del sistema in sottosistemi;
- Il mapping Hardware/Software;
- La gestione dei dati persistenti;
- Il controllo degli accessi e della sicurezza;
- Il controllo del flusso globale del sistema;
- Le condizioni limite.

Al quarto punto vengono presentati i servizi di ogni sottosistema.

Al quinto punto viene fornito il glossario dei termini utilizzati nel documento con le relative definizioni.

2. ARCHITETTURA DEL SISTEMA CORRENTE

L'architettura del sistema corrente si rifà a Steam, una delle piattaforme più importanti ed utilizzate nell'ambito della vendita di prodotti di stampo videoludico.

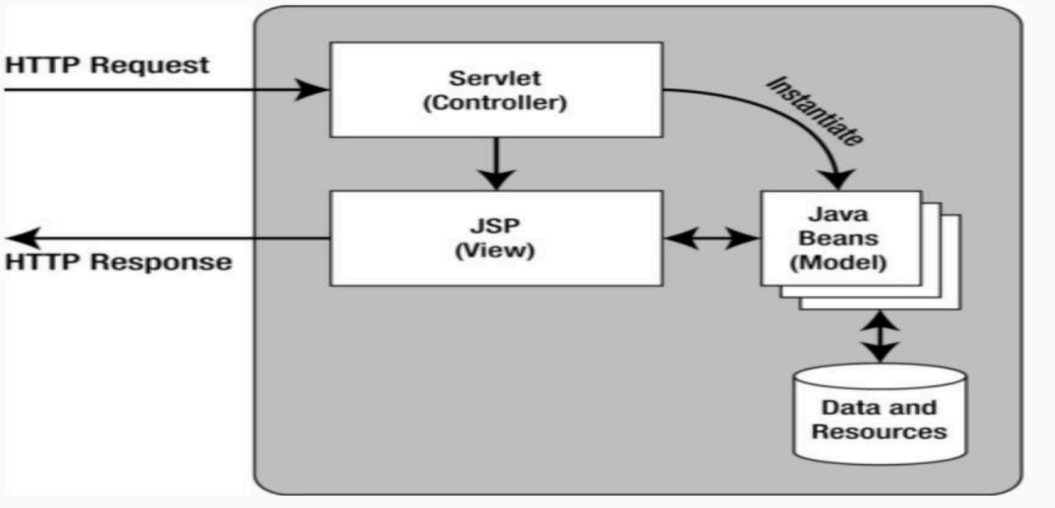
3. ARCHITETTURA DEL SISTEMA PROPOSTO

3.1. Panoramica

L'architettura del sistema proposto si rifà a quella del sistema corrente, offrendo delle funzioni fondamentali:

- Consentire la visualizzazione dei videogiochi più rilevanti sul mercato, facilitando la ricerca del prodotto.
- Rendere possibile acquistare prodotti e visualizzare tutti i prodotti acquistati in una libreria virtuale.
- Consentire la gestione degli ordini effettuati da parte del singolo utente.

Per la progettazione e per lo sviluppo di Infinity Games è stata scelta un'architettura MVC (Model-View-Controller) che separa la logica di business da quella di presentazione.



Model: Si occupa di gestire i dati persistenti, contiene metodi di lettura e scrittura sul Database.

View: Si occupa di far visualizzare i dati all'utente e si occupa dell'interazione tra quest'ultimo ed il Sistema.

Controller: Si occupa della logica di controllo dell'applicazione interagendo con gli altri due componenti.

3.2. Decomposizione sottosistemi

Presentiamo una decomposizione del nostro sistema in 3 layer (o livelli: Presentation Layer, Application Logic Layer e Data Access Layer che si occupano di gestire funzionalità ed aspetti differenti del nostro sistema:

Presentation Layer(View): Livello che gestisce la comunicazione con le entità esterne al sistema tramite delle componenti che si occupano di rappresentare l'informazione verso i client, consentendo essi di interagire con il sistema.

Utente non registrato: GUI che gestisce l'interfaccia degli utenti che non hanno effettuato l'accesso e/o non hanno effettuato la registrazione alla piattaforma, i quali sono limitati alla visualizzazione dei prodotti dello store.

Utente registrato: GUI che gestisce l'interfaccia degli utenti che sono registrati e che hanno effettuato l'accesso alla piattaforma, e che sono abilitati ad acquistare prodotti, visualizzare la propria area personale, contattare l'assistenza ed inserire prodotti nella propria lista dei desideri.

Amministratore: GUI che gestisce l'interfaccia dedicata ai responsabili della piattaforma, e che sono abilitati all'aggiunta e alla rimozione di prodotti e categorie, promuovere utenti ad amministratori, declassare utenti dallo stato di amministratore, visualizzare gli ordini di ogni singolo utente e rimuovere utenti dalla piattaforma.

Application Layer(Control): E' il livello del sistema che si occupa del processamento dei dati necessario per produrre i risultati da inoltrare al Presentation Layer.

Gestione Utente: Sottosistema che permette di registrarsi, effettuare il login e il logout, accedere ai servizi relativi all'area personale e la visualizzazione di quest'ultima.

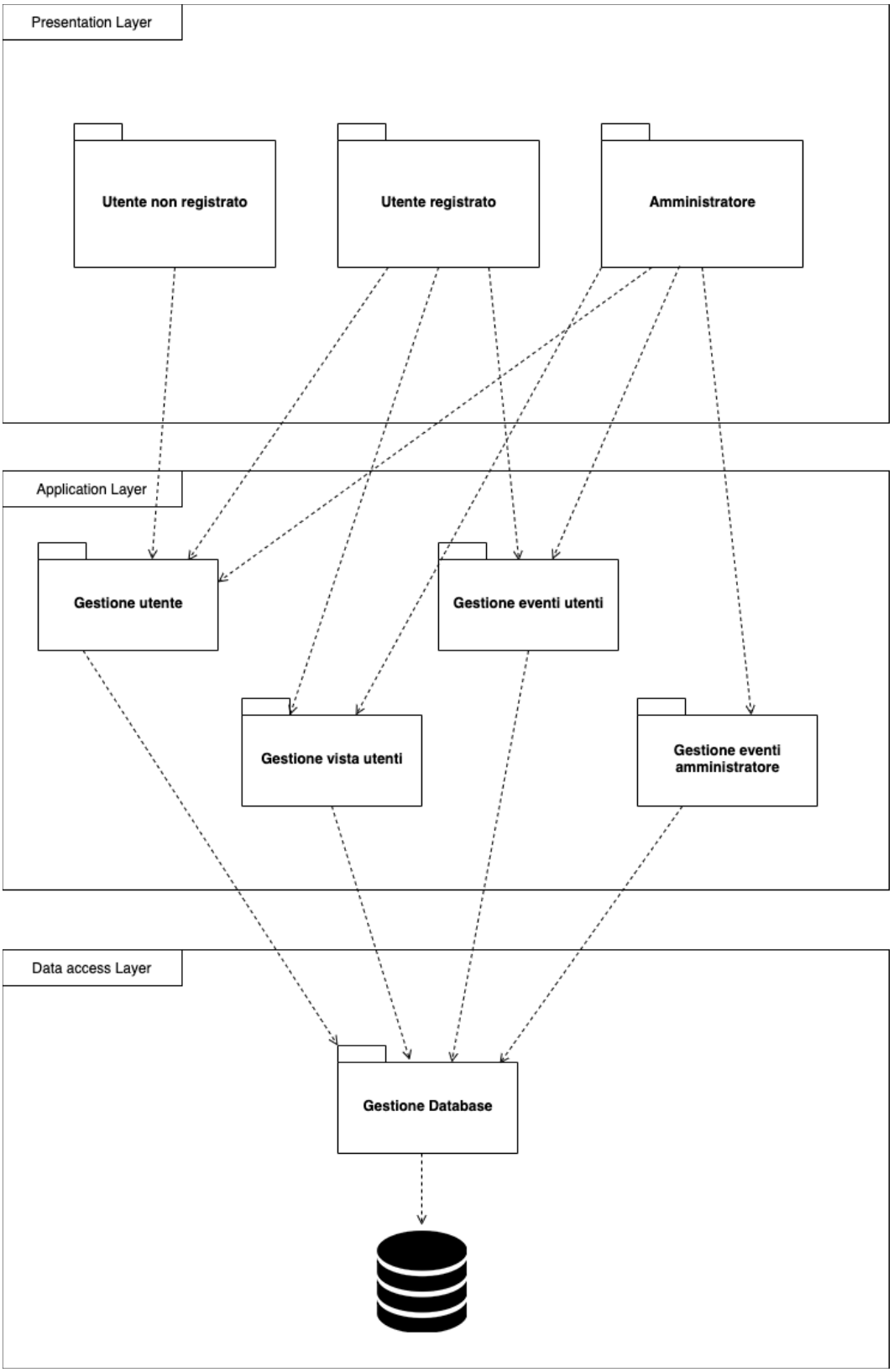
Gestione vista utenti: Questo sottosistema permette la visualizzazione della libreria, dello storico degli ordini effettuati, il carrello e la lista dei desideri.

Gestione eventi utenti: sottosistema che permette l'acquisto di un videogioco.

Gestione eventi Amministratore: sottosistema che permette la gestione di tutte le funzionalità di un amministratore della piattaforma, quali: aggiunta e rimozione dei prodotti allo store, aggiunta e rimozione di categorie di videogiochi, promozione e declassamento di utenti e rimozione di utenti dalla piattaforma.

Data Access Layer(Model) : Livello che gestisce i dati necessari al funzionamento dell'intero sistema ovvero i dati persistenti.

Gestione DataBase(DB): sottosistema che si occupa di immagazzinare e di prelevare i dati persistenti dal nostro database.


Presentation Layer
Utente non registrato
Utente registrato
Amministratore
Application Layer
Gestione utente
Gestione eventi utenti
Gestione vista utenti
Gestione eventi amministratore
Data access Layer
Gestione Database

3.3. Mapping hardware/software

Per il sistema che si vuole realizzare ci interessa un framework o librerie per applicazioni web.

Il framework che andremo ad utilizzare è jQuery, Javascript, AJAX, JSON, JSTL.

jQuery è una libreria JavaScript per accelerare lo sviluppo della business logic delle interfacce web.

Javascript è un linguaggio di scripting orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione Web lato client.

AJAX, è una tecnica di sviluppo software per la realizzazione di applicazioni web interattive (Rich Internet Application), basandosi su uno scambio di dati in background fra web browser e server.

JavaScript Object Notation, è un formato adatto all'interscambio di dati fra applicazioni client/server. Viene usato in AJAX come alternativa a XML/XSLT.

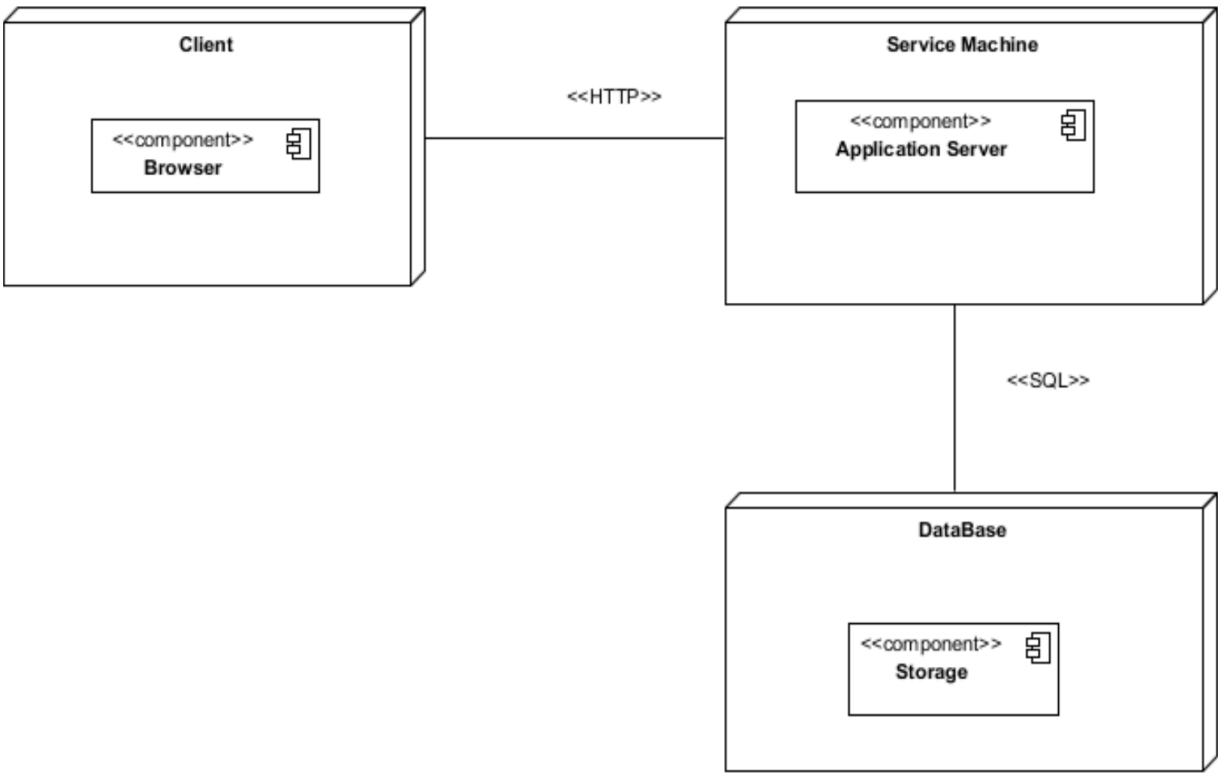
JSTL è un'estensione di JSP ed incorpora un insieme di tag HTML definiti tramite file XML e programmati in linguaggio Java.

Per il sistema, basato su un’architettura MVC, sono state scelti questi tipi di configurazioni:

- per la parte di Interface (View) usiamo le JSP, HTML, CSS, JavaScript.
- per la parte di business logic (Control) usiamo Apache Tomcat e Java.
- per la parte di Data access (Model) usiamo MySQL con JDBC.
- Come Protocollo di comunicazione usiamo HTTP, TCP/IP.

3.3.1. Deployment

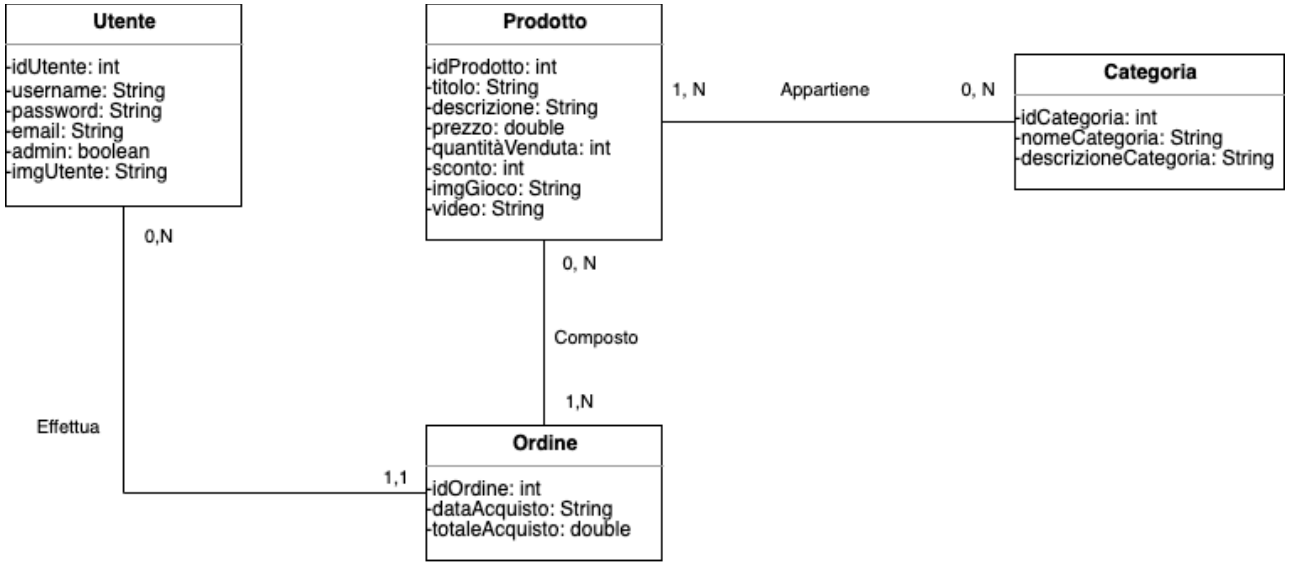
Gli utenti potranno interagire con Infinity Games tramite un WebBrowser collegandosi all'indirizzo del nostro sito. Il Control si occupa di: interpretare gli eventi generati dall'utente, richiedere e prelevare le opportune risorse dalla Base di Dati ed inviare le risposte tramite protocollo HTTP.



3.4. Gestione Dati Persistenti

Il sistema che si vuole sviluppare sarà installato su un solo computer e utilizzerà un DBMS MySQL installato sullo stesso. Il sistema sarà diviso in client e server che inizialmente saranno lo stesso pc, ma in caso di futura necessità potranno essere facilmente divisi in quanto i servizi saranno progettati separatamente. Sul server ci sarà un DBMS per la gestione dei dati persistenti di nostro interesse.

3.4.1. Schema Logico



3.4.2. Struttura delle Tabelle

Tabella Utente

Attributo	Tipo	Vincoli
idUtente	int	primary key
username	string	not null unique
password	string	not null
email	string	not null unique
admin	boolean	not null
imgUtente	string	

Tabella Ordine

Attributo	Tipo	Vincoli
idOrdine	int	primary key
idUtente	int	foreign key Utente(idUtente)
dataAcquisto	string	not null
totaleAcquisto	double	not null

Tabella Prodotto

Attributo	Tipo	Vincoli
idProdotto	int	primary key
titolo	string	not null
descrizione	string	not null
prezzo	double	not null
quantitàVenduta	int	not null
sconto	int	
imgGioco	string	
video	string	

Tabella Categoria

Attributo	Tipo	Vincoli
idCategoria	int	primary key
nomeCategoria	string	not null
descrizione	string	not null