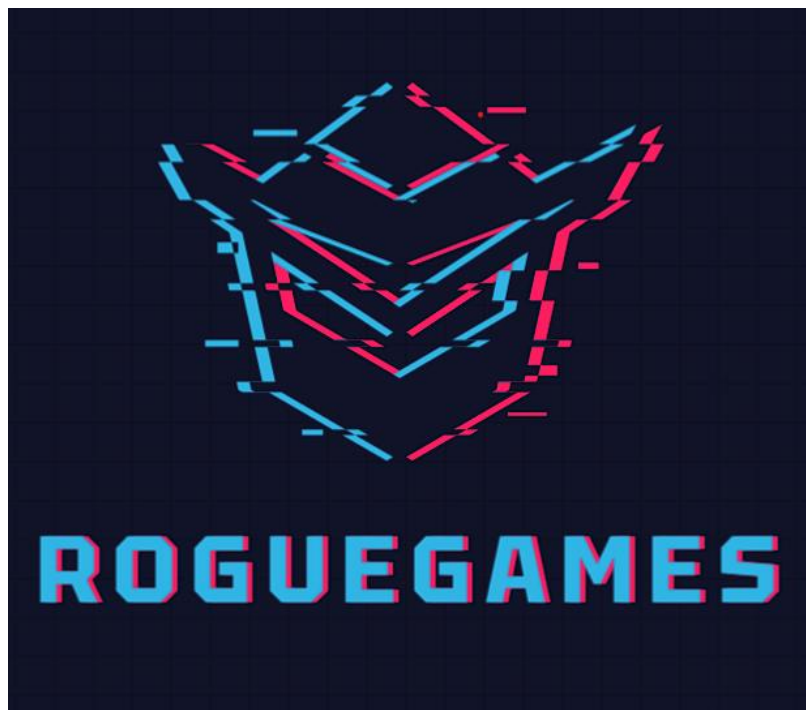


**RogueGames**  
**System Design Document**  
**Versione 0.1**



Data: 24/11/2024

Progetto: RogueGames	Versione: 0.1
Documento: System Design Document	Data: 24/11/2024

**Coordinatore del progetto:**

Nome	Matricola

**Partecipanti:**

Nome	Matricola
Marco Battaglia	0512116374
Giovanni Lavorato	0512116371
di Caprio Domenico	0512118543
De Vita Antonio	0512116944

<b>Scritto da:</b>	Giovanni Lavorato Battaglia Marco di Caprio Domenico De Vita Antonio
--------------------	---

**Revision History**

Data	Versione	Descrizione	Autore
24/11/2024	0.1	Stesura del System Design Document (SDD)	Giovanni Lavorato Battaglia Marco di Caprio Domenico De Vita Antonio

Progetto: RogueGames	Versione: 0.1
Documento: System Design Document	Data: 24/11/2024

## Indice

1. INTRODUCTION .....	4
1.1. Purpose of the system.....	4
1.2. Design goals .....	4
1.2.1. Criteri End User .....	4
1.2.1.1. Usabilità .....	4
1.2.2. Criteri di Affidabilità .....	4
1.2.2.1. Disponibilità .....	4
1.2.2.2. Robustezza .....	4
1.2.2.3. Tolleranza ai Fault .....	5
1.2.2.4. Protezione .....	5
1.2.3. Criteri di Performance.....	5
1.2.3.1 ThroughPut .....	5
1.2.3.2 Tempo di risposta.....	5
1.2.4. Criteri di Supportabilità .....	5
1.2.4.1 Mantenibilità .....	5
1.2.4.2 Estensibilità .....	5
1.2.4.3 Portabilità .....	5
1.3. Definition, acronym and abbreviations .....	5
1.4. Reference.....	5
2. CURRENT SOFTWARE ARCHITETURE .....	6
3. PROPOSED SOFTWARE ARCHITETURE.....	6
3.1. Overview .....	6
3.2. Subsystem decomposition .....	6
3.3. Hardware/software mapping .....	8
3.4. Persistent data management .....	8
3.5. Access control and security .....	9
3.5.1. Matrice degli accessi Manager Utente .....	9
3.5.2. Matrice degli accessi Manager Prodotto.....	9
3.5.3. Matrice degli accessi Manager Gestore .....	10
3.6. Global software control .....	10
3.7. Boundary conditions .....	10
4. SUBSYSTEM SERVICES .....	11
4.1. Servizi della Manager Utente .....	11
4.2. Servizi della Manager Prodotto.....	12
4.3. Servizi della Manager Gestore .....	13

Progetto: RogueGames	Versione: 0.1
Documento: System Design Document	Data: 24/11/2024

# 1.Introduzione

## 1.1 Purpose of the system

Lo obiettivo del sistema è di creare una piattaforma user-friendly che permette a tutti gli appassionati di videogiochi di poter trovare i titoli che desiderano per le loro piattaforme preferite.

Si tratta di una e-commerce online basato sulla vendita di videogiochi e di tutti i prodotti inerenti a quel mondo.

Il sito offre agli utenti una serie di funzionalità per poter cercare i loro videogiochi in modo rapido e veloce, mettendo a loro disposizione una serie di metodi per poter trovare i loro titoli. Inoltre, gli verrà permesso di visualizzare tutti i dettagli di un gioco e di poterlo acquistare con tanta semplicità, solo con previo accesso al loro account del sito.

Agli utenti loggati viene offerto anche la possibilità di gestire e visualizzare il proprio profilo, di poter usufruire di sconti e di poter accedere al proprio storico degli ordini.

Infine, garantisce un'elevata sicurezza per le transazioni online.

Per poter raggiungere il sito, basterà loro avere un qualsiasi dispositivo con connessione ad internet e con un qualsiasi motore di ricerca. Basterà digitare il nome del sito e si verrà direttamente indirizzati sulla homepage.

Ai gestori, invece, gli verranno forniti una serie di interfacce per potere: gestire il catalogo, consentendo di poter aggiungere, modificare o eliminare prodotti da esso; gestire e visualizzare lo storico degli ordini effettuati da tutti gli utenti del sito.

## 1.2 Design goals

### 1.2.1 Criteri di End User

#### 1.2.1.1 Usabilità

L'interfaccia utente deve essere intuitiva, seguendo convenzioni standard, con guide online e documentazione per facilitare la navigazione e l'uso del sito da parte di utenti di ogni livello.

### 1.2.2 Criteri di Affidabilità

#### 1.2.2.1 Disponibilità

Il sito deve essere attivo 24 ore su 24, 7 giorni su 7, garantendo un'alta disponibilità per gli utenti.

#### 1.2.2.2 Robustezza

Il sito deve garantire che il sistema funzioni anche se l'utente inserisce input sbagliati

Progetto: RogueGames	Versione: 0.1
Documento: System Design Document	Data: 24/11/2024

#### **1.2.2.3 Tolleranza ai fault**

Il sito deve gestire correttamente tutte le eccezioni, mantenendo la stabilità del servizio anche in caso di errori imprevisti.

#### **1.2.2.4 Protezione**

Il sito deve garantire la capacità di resistere ed evitare attacchi malevoli.

### **1.2.3 Criteri di Performance**

#### **1.2.3.1 Throughput**

Il sito deve consentire a più utenti di collegarsi simultaneamente senza problemi di sovraccarico, garantendo la scalabilità e robustezza del sistema.

#### **1.2.3.2 Tempo di risposta**

Il sito deve rispondere alle richieste dell'utente entro due secondi, garantendo tempi di risposta rapidi per una migliore esperienza di navigazione.

### **1.2.4 Criteri di Supportabilità**

#### **1.2.4.1 Mantenibilità**

Il sito deve essere facilmente aggiornabile e mantenibile per permettere future modifiche o aggiustamenti con minimo impatto sull'operatività.

#### **1.2.4.2 Estensibilità**

Il sito deve poter permettere di aggiungere nuove funzionalità in modo facile.

#### **1.2.4.3 Portabilità**

Il sito deve poter essere visualizzabile su qualsiasi dispositivo, sia esso un computer o dispositivo mobile, che abbia una connessione ad internet e che faccia uso di un qualsiasi motore di ricerca.

## **1.3 Definition, acronyms and abbreviations**

RAD: Requirements Analysis Documentation

DBMS: DataBase Managment System

## **1.4 References**

RAD\_RogueGames

Progetto: RogueGames	Versione: 0.1
Documento: System Design Document	Data: 24/11/2024

## 2. Current software architecture

L'architettura di RogueGames è progettata con un approccio modulare e orientato ai servizi per garantire scalabilità e facilità di manutenzione. Il sistema si basa su un'architettura three-tier con pattern MVC che include componenti per l'interfaccia utente, la logica del gioco e la gestione dei dati persistenti.

## 3. Proposed software architecture

### 3.1 Overview

Il sistema si basa sul modello Three-tier caratterizzato dalla separazione di tre livelli, ciascuno con responsabilità distinte:

• **Presentation Layer:**

Questo livello è responsabile di presentare i dati all'utente in un formato comprensibile, e interagisce con l'application layer per inviare richieste e ricevere dati.

• **Application Layer:**

Questo livello si occupa di gestire la logica di business. È responsabile della manipolazione dei dati, dell'esecuzione delle operazioni richieste dagli utenti.

• **Data Access Layer:**

Questo livello si occupa dell'accesso ai dati e della persistenza.

### 3.2 Subsystem decomposition

#### Presentation Layer

- GestoreGUI: Interfacce disponibili esclusivamente al gestore.
- UtenteGUI: Interfacce front-end utilizzate da tutti gli utenti.
- ProdottiGUI: Interfacce front-end dedicate ai prodotti.

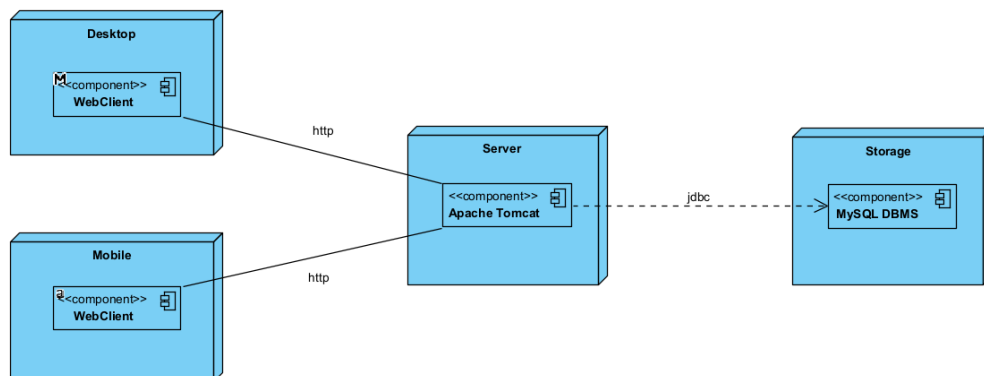
#### Application Layer

- GestioneUtente: Coordina la logica dei dati, operazioni e stati dell'utente.
- GestioneGestore: Coordina la logica per le operazioni effettuabili dal gestore.
- GestioneProdotto: Coordina la logica per le operazioni effettuabili sui prodotti e i loro stati
- GestioneOrdine: Coordina la logica per le operazioni effettuabili sugli ordini
- GestioneCarrello: Coordina la logica per le operazioni CRUD riguardanti il carrello
- GestionePreferiti: Coordina la logica per le operazioni CRUD riguardanti i preferiti



Progetto: RogueGames	Versione: 0.1
Documento: System Design Document	Data: 24/11/2024

### 3.3 Mapping hardware/software



Il sistema è suddiviso in tre parti abbiamo:

1. **Utenti/Desktop e Mobile:** Gli utenti interagiscono con il sistema attraverso un WebClient. Questo avviene grazie all'utilizzo del protocollo HTTP.
2. **Server:** Sistema ha come suo centro il server, che ospita il Web Server Apache Tomcat. Questa applicazione gestisce le richieste degli utenti e si interfaccia con lo storage del sistema.
3. **Storage:** Lo storage del sistema contiene un DBMS MySQL. Il server accede al DBMS usando il protocollo JDBC per recuperare o memorizzare i dati.

### 3.4 Persistent data management

I dati persistenti sono gestiti tramite DBMS relazionali e tramite **Spring Data JPA**, che utilizza **Hibernate** come ORM (Object-Relational Mapping) per mappare le entità Java alle tabelle del database. Questo approccio ci permette di semplificare la gestione dei dati e astrarre la complessità del database sottostante.

L'ORM semplifica le operazioni di accesso ai dati e consente di rappresentare gli oggetti dell'applicazione in modo più diretto nella base di dati relazionale. Inoltre, agisce come uno strato intermedio tra le entità del sistema e la base di dati, semplificando le operazioni di creazione, lettura, scrittura e modifica (Create, Read, Update, Delete).



Progetto: RogueGames	Versione: 0.1
Documento: System Design Document	Data: 24/11/2024

### 3.5 Access control and security

Per l'autenticazione al sistema l'utente inserirà l'email e la password, di quest'ultima verrà eseguita una funzione di hash dopodiché verrà salvata nel Database.

Di seguito riportiamo la matrice degli accessi.

ATTORI	CLASSE	FUNZIONI
Gestore	Utente	R, visualizzaProfilo() U, modificaProfilo()
Utente	Utente	R, visualizzaProfilo() U, modificaProfilo()
Utente	Ordine	U, VisualizzaOrdine()
Gestore	Ordine	R, FiltraggioOrdiniPerData() R, FiltraggioOrdiniPerEmail()
Utente	Carrello	C, aggiungialCarrello() D, rimuoviprodottoCarrello() U, modificaQnt()
Utente	Preferiti	C, aggiungiaiPreferiti() D, rimuoviprodottoPreferiti()
Gestore	Catalogo	C, saveProdotto() U, updateProdotto() D, removeProdotto() R, ordinamentoProdottiData() R, filtraggioProdottiPiattaforma()
Utente	Catalogo	R, ordinamentoProdottiData() R, filtraggioProdottiPiattaforma()

Progetto: RogueGames	Versione: 0.1
Documento: System Design Document	Data: 24/11/2024

## 3.6 Global software control

Il nostro sistema avrà un controllo esplicito di tipo centralizzato, il controllo risiede in un dispatcher. Le richieste sono gestite attraverso il Web Server, che funge da intermediario per il sistema. Inizialmente, vengono sottoposte a filtri dedicati per operazioni di autenticazione e gestione delle sessioni. Successivamente le richieste vengono smistate ai rispettivi controller. Questi controller interagiscono con i servizi che contengono la logica di business del sistema. Una volta completate le elaborazioni, le risposte vengono inviate dal controller al client.

## 3.7 Boundary conditions

### 3.7.1 Start-up

Per l'inizializzazione del sistema bisogna avviare il DBMS, ovvero MySQL, così da poter accedere ai dati persistenti. L'apertura della connessione al database MySQL viene gestita attraverso JDBC. Dopodiché mediante l'utilizzo di Spring Boot, bisogna avviare il Web Server, nonché Tomcat dove avviene il deployment dell'applicazione, la quale espone le funzionalità del sistema in modo trasparente agli utenti.

### 3.7.2 Shutdown

In questa fase avviene lo spegnimento del DBMS e del Web Server, prima però l'applicazione si assicura che tutte le attività in corso siano completate prima della terminazione

### 3.7.3 Error Behavior of the system

In caso di problemi di connessione al database, il sistema ritenta la connessione. In caso di problemi di alimentazione viene effettuato il ripristino all'ultimo stato persistente.

## 4. Subsystems services

### 4.1 Servizi della Gestione Utente

Servizio	Descrizione
Autenticazione	Un servizio che ci permette di controllare le credenziali dell'utente e ci permette di poter controllare quali operazioni può eseguire

Progetto: RogueGames	Versione: 0.1
Documento: System Design Document	Data: 24/11/2024

<b>ProfileUpdate</b>	Servizio che permette all'utente di modificare tutti i campi del suo profilo
<b>IndirizzoSpedizioneCRUD</b>	Fornisce tutte le operazioni per gestire i suoi indirizzi di spedizione
<b>CartaCreditoCRUD</b>	Fornisce tutte le operazioni per gestire le sue carte di credito

## 4.2 Servizi di GestioneOrdine

<b>Servizio</b>	<b>Descrizione</b>
<b>Acquisto</b>	Un servizio che permette di poter gestire il check-out
<b>OrdiniProvider</b>	Un servizio che fornisce una lista di ordini (di un singolo utente o di più utenti)

## 4.3 Servizi di GestioneProdotto

<b>Servizio</b>	<b>Descrizione</b>
<b>ProductProvider</b>	Un servizio che fornisce una lista di prodotti e che permette di poter visualizzare un prodotto particolare

## 4.4 Servizi di GestioneGestore

<b>Servizio</b>	<b>Descrizione</b>
<b>ProductCRUD</b>	Un servizio che permette al gestore di poter creare e organizzare i prodotti

Progetto: RogueGames	Versione: 0.1
Documento: System Design Document	Data: 24/11/2024

#### 4.5 Servizi di GestioneCarrello

Servizio	Descrizione
CarrelloCRUD	Un servizio che permette di poter gestire tutte le operazioni di carrello

#### 4.6 Servizi di GestionePreferiti

Servizio	Descrizione
PreferitiCRUD	Un servizio che permette di poter gestire tutte le operazioni di preferiti