



iSnapGaming

SDD

SDD	3
Introduction	4
Design Goals	4
Proposed Software Architecture	5
Subsystem Decomposition	5
Mapping Hardware-Software	7
Persistence Data Management	7
Access Control and Security	8
Global Control Flow	8
Boundary Conditions	8
Installazione e avvio del sistema	8
Spegnimento del sistema	9

SDD

Introduction

Design Goals

Dependability

Security

- NRF_SEC_1
 - Prevenzione SQL injection: Il sistema deve garantire la sicurezza e la protezione dei dati da attacchi di tipo SQL injection, assicurando che tutte le query SQL e i parametri siano correttamente validati e sanificati prima dell'esecuzione.

Robustness

- NRF_ROB_1
 - Gestione eccezioni: Gestione delle eccezioni client-side. In caso di eccezioni e dati errati, il sistema dovrà essere in grado di gestirli e non dovrà andare in crash.

Usability

- NRF_USA_1
 - Design responsive: Il sistema deve implementare un design responsive, in modo che l'interfaccia si adatti in maniera fluida a diverse dimensioni dello schermo, consentendo agli utenti di navigare e fare acquisti sia da desktop che da dispositivi mobile.
- NRF_USA_2
 - Immagini di alta qualità: Il sistema utilizza immagini chiare e di alta qualità per favorire l'appeal dei prodotti e dell'intera applicazione.
- NRF_USA_3
 - Selezione di colori: Il sistema deve adottare una specifica palette di colori, definita come un insieme ben preciso di valori esadecimali nel formato #RRGGBB, al fine di assicurare una chiara distinzione visuale tra le funzionalità di registrazione, acquisto e le altre funzioni principali dell'applicazione. I colori utilizzati dovrebbero essere selezionati in modo tale da consentire agli utenti di identificare rapidamente e senza ambiguità ciascuna funzionalità.
- NRF_USA_4
 - Utilizzo di font della famiglia sans serif: I font senza grazie tendono ad essere più leggibili su schermi digitali, come quelli dei computer e dei dispositivi mobili. La mancanza di grazie semplifica la visualizzazione dei caratteri su pixel, contribuendo a una maggiore chiarezza e nitidezza.

Proposed Software Architecture

La piattaforma è basata su un'architettura *three-tier*, dunque verranno distinti il livello di Presentazione, il livello di Business e il livello di Persistenza. Inoltre, l'applicazione sarà di tipo *client-server*, per cui gli utenti (i clients) usufruiranno dei servizi dell'applicativo che verranno ospitati sul server.

Subsystem Decomposition

Analizzando i servizi e raggruppandoli in base alle funzionalità, sono stati individuati i seguenti sottosistemi:

Presentation layer:

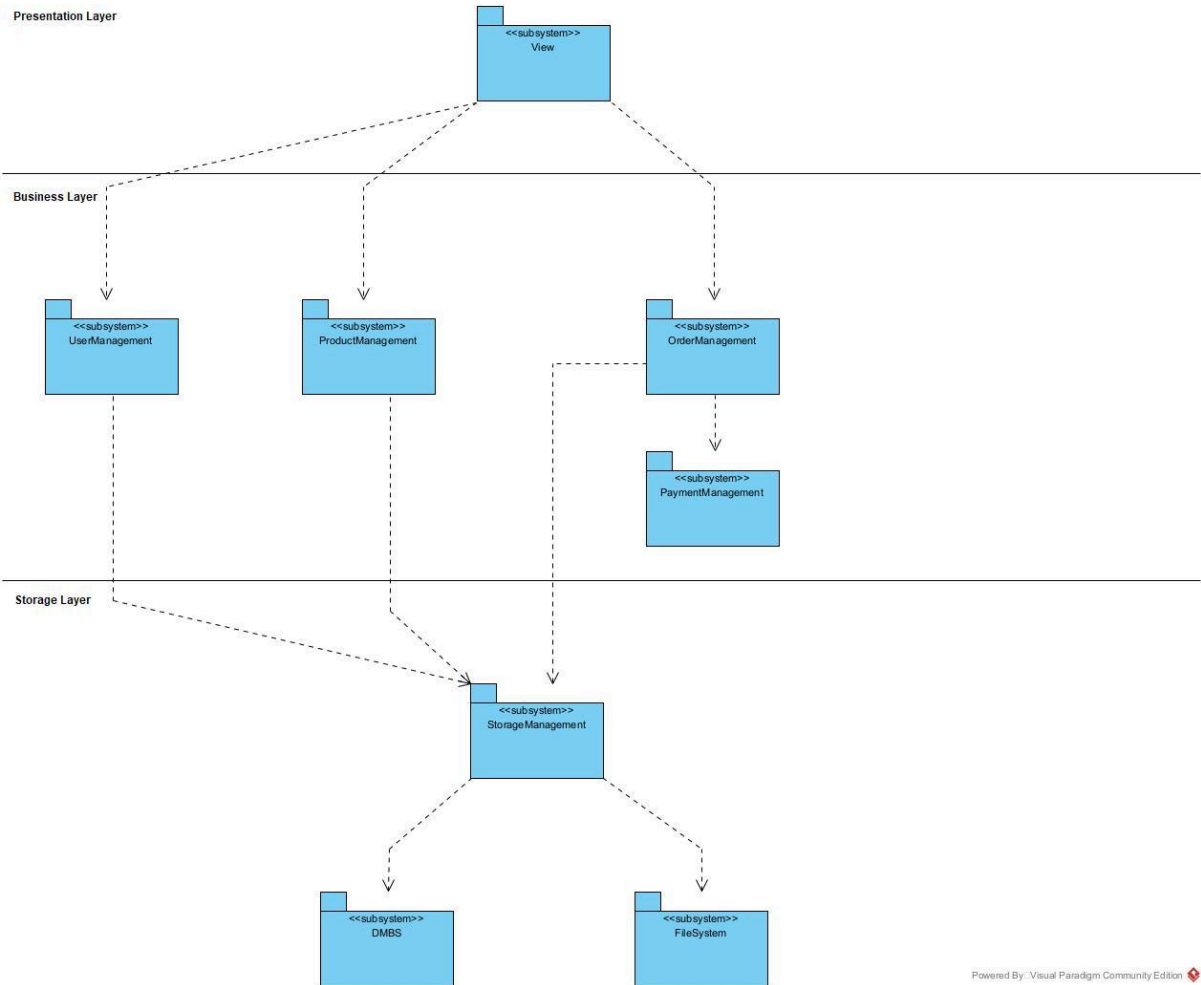
- **View:** tale sottosistema, raggruppa le pagine web della piattaforma e le servlet. Inoltre, interagisce con tutti i sottosistemi del Business layer e li coordina in quanto la logica di controllo è implementata nelle servlet.

Business layer:

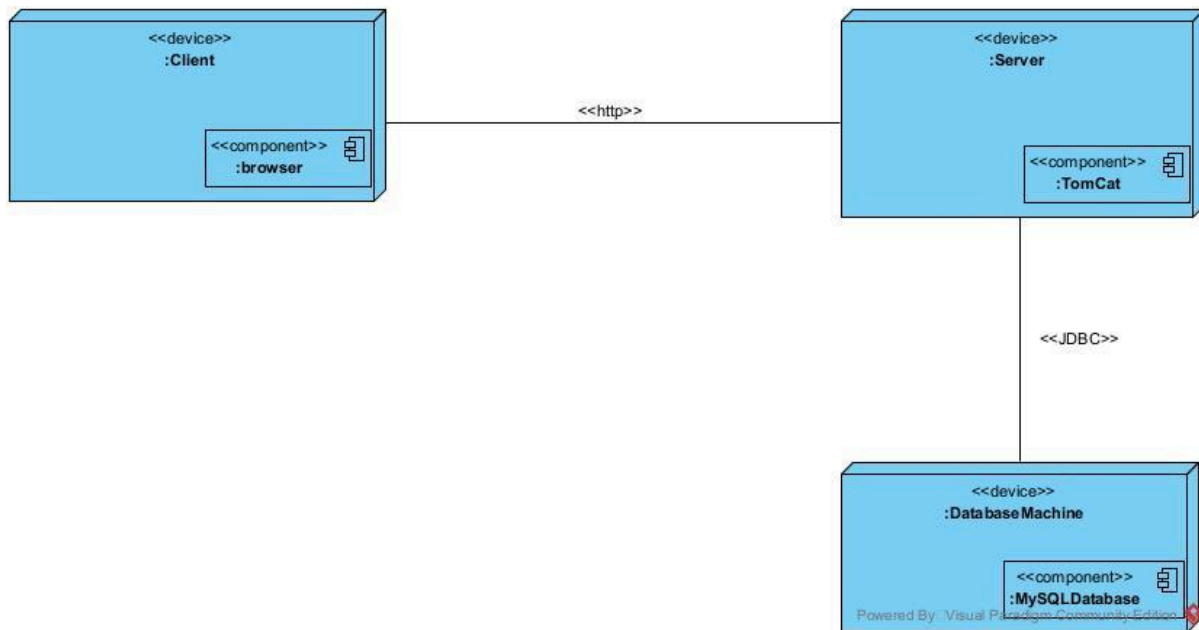
- **UserManagement Subsystem:** tale sottosistema raggruppa tutti i servizi che offrono le funzionalità di registrazione, gestione e autenticazione degli utenti.
- **ProductManagement Subsystem:** tale sottosistema raggruppa tutti i servizi che permettono la gestione dei prodotti.
- **OrderManagement Subsystem:** tale sottosistema raggruppa tutti i servizi che permettono la gestione degli ordini, inclusa la creazione di questi nonché l'aggiunta di prodotti al carrello da parte del Customer.
- **PaymentManagement Subsystem:** tale sottosistema raggruppa tutti i servizi che permettono di effettuare un pagamento.

Persistence layer:

- **StorageManagement Subsystem:** tale sottosistema utilizza il servizio di persistenza offerto dal DBMS; aggiunto per ridurre il grado di accoppiamento dei sottosistemi del Business Layer con il Database del Persistence Layer.
- **DBMS**
- **File System**



Mapping Hardware-Software



Persistence Data Management

Sono stati individuati i seguenti dati che necessitano di essere resi persistenti:

- User, Manager e le rispettive sottoclassi e classi associate (Address);
- Product;
- Order;
- Cart.

Per la gestione della persistenza, il sistema utilizzerà un database relazionale che garantisce accesso e utilizzo dei dati facile, sicuro e consistente. Il DBMS scelto è MySQL. Le immagini, invece, saranno rese persistenti nel File System.

Nota: Cart non verrà reso persistente nel database, ma la sua persistenza sarà garantita attraverso la sessione.

Access Control and Security

Attori Oggetti	Guest	Customer	Product Manager	Order Manager
User	makeUser()	login() logout() setCurrentRole()	login() logout() setCurrentRole()	login() logout() setCurrentRole()
Product Manager			addProduct() updateProduct() removeProduct()	
Order		pay()		checkProduct() packProduct() replaceProduct() contactCourier() restoreOrder() confirmDelivery()
Cart	addToCart() removeFromCart() decreaseQuantityCart()	addToCart() removeFromCart() decreaseQuantityCart()		
Address		makeAddress()		
Product			makeProduct()	

Global Control Flow

Il sistema prevede un controllo del software globale centralizzato di tipo event-driven, in cui c'è un componente centrale che gestisce tutte le attività del sistema in base agli eventi che si verificano. Ovviamente, per permettere a più attori di utilizzare la piattaforma e i suoi servizi contemporaneamente, lato server verrà utilizzato un control flow basato su threads.

Boundary Conditions

Installazione e avvio del sistema

La fase di installazione del sistema verrà effettuata da un addetto che innanzitutto provvederà alla configurazione e al popolamento della base di dati con i dati dei manager forniti dal cliente. Successivamente, installerà il server container su una macchina remota. L'addetto avvierà, a questo punto, il sistema e il database MySQL che verranno connessi tra loro tramite driver JDBC.

Spegnimento del sistema

Il sistema verrà spento da un addetto al quale sarà sufficiente spegnere il DBMS e il server web. Verrà garantita la consistenza dei dati, annullando eventuali operazioni ancora in corso.