



Università degli studi di Napoli Parthenope

**Progetto Ingegneria del Software e Interazione
Uomo Macchina**

A.A. 2023/2024

Documento di Progettazione

“GamersHarem”

Proponenti:

Porritiello Francesco	0124002486
Esposito Francesco	0124002629

1- Introduzione.....	3
1.1 - Scopo del Sistema.....	3
1.2 - Obiettivi di progettazione.....	3
1.3 - Riferimenti.....	4
1.4 - Panoramica.....	4
2 - Sistema Corrente.....	4
3 - Sistema Proposto.....	5
3.1 - Panoramica.....	5
3.2 - Decomposizione del sistema.....	6
3.2 - Hardware/Software Mapping.....	7
3.4 - Gestione dei dati persistenti.....	7
3.5 - Controllo accessi e sicurezza.....	7

1- Introduzione

1.1 - Scopo del Sistema

Gamers Harem è un marketplace dedicato all'acquisto di videogiochi.

Esso mira ad offrire ai consumatori una vasta gamma di titoli, prezzi convenienti ed un'esperienza utente coinvolgente creando un ambiente che favorisca la soddisfazione del cliente e la fidelizzazione con lo scopo di divenire la destinazione preferita per gli appassionati di videogiochi.

Gli utenti di Gamers Harem sono appassionati di videogiochi appartenenti a diverse fasce d'età. Il marketplace mira a coprire le esigenze di videogiocatori occasionali o hardcore, offrendo un'ampia varietà di generi e titoli per diverse piattaforme.

1.2 - Obiettivi di progettazione

La progettazione di Gamers Harem avviene in modo user-centered rispettando i criteri quali:

- **Usabilità:**
 - Garantire che l'apprendimento del linguaggio visivo sia intuitivo e rapido.
 - Adottare un design semplice, arricchito da immagini e simboli familiari al pubblico di destinazione.
- **Estensibilità:**
 - Progettare le componenti in modo che possano essere facilmente modificate o ampliate, senza richiedere modifiche radicali al sistema.
- **Portabilità:**
 - Sviluppare il sistema utilizzando Html, CSS, JavaScript e Python per garantire la compatibilità come progressive web application, accessibile da qualsiasi browser e da qualsiasi dispositivo sia desktop che mobile rendendolo responsive.
- **Sicurezza dei Dati:**
 - Implementare la crittografia per proteggere i dati sensibili, inclusi nome, e-mail e password.
 - Garantire la sicurezza dei dati inseriti dagli utenti, con particolare attenzione alle credenziali di accesso.
- **Performance**
 - L'interfaccia utente deve rispondere prontamente agli input degli utenti, garantendo una navigazione fluida e senza intoppi.

1.3 - Riferimenti

Si faccia riferimento al RAD per una maggiore comprensione del lavoro esposto.

1.4 - Panoramica

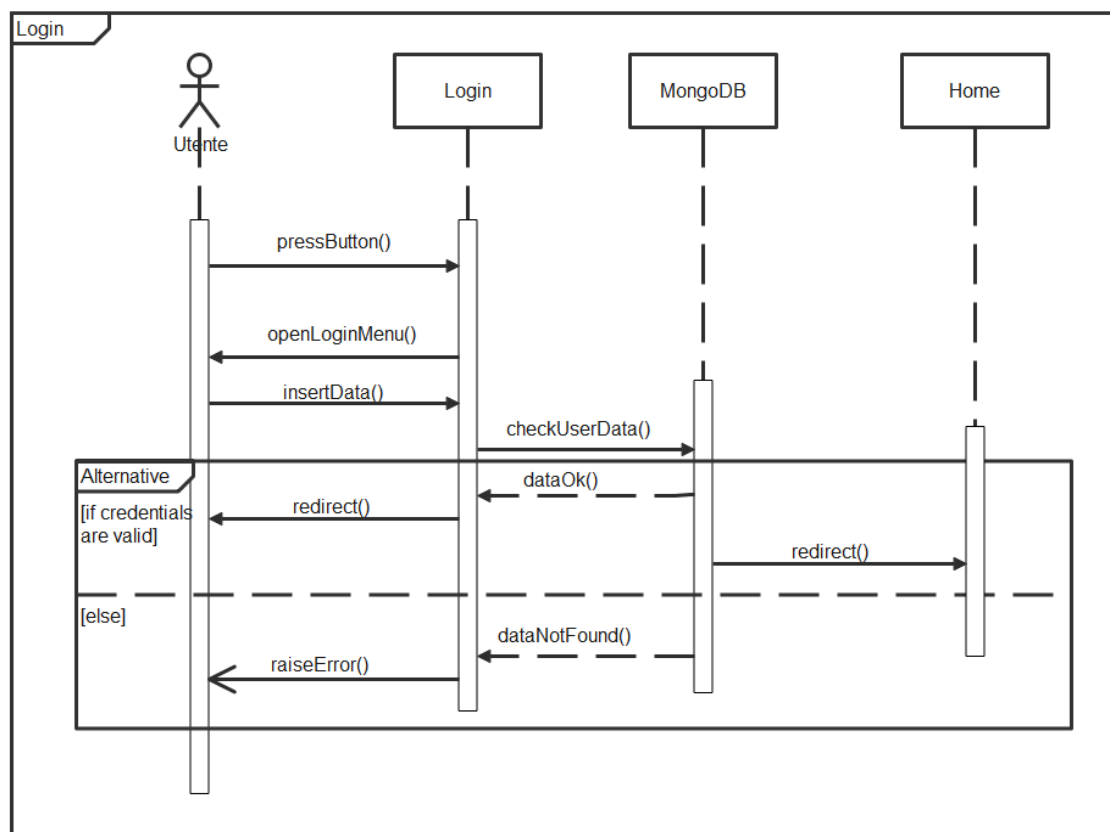
La WebApp è dedicata all'esperienza degli utenti nel mondo dei videogiochi.

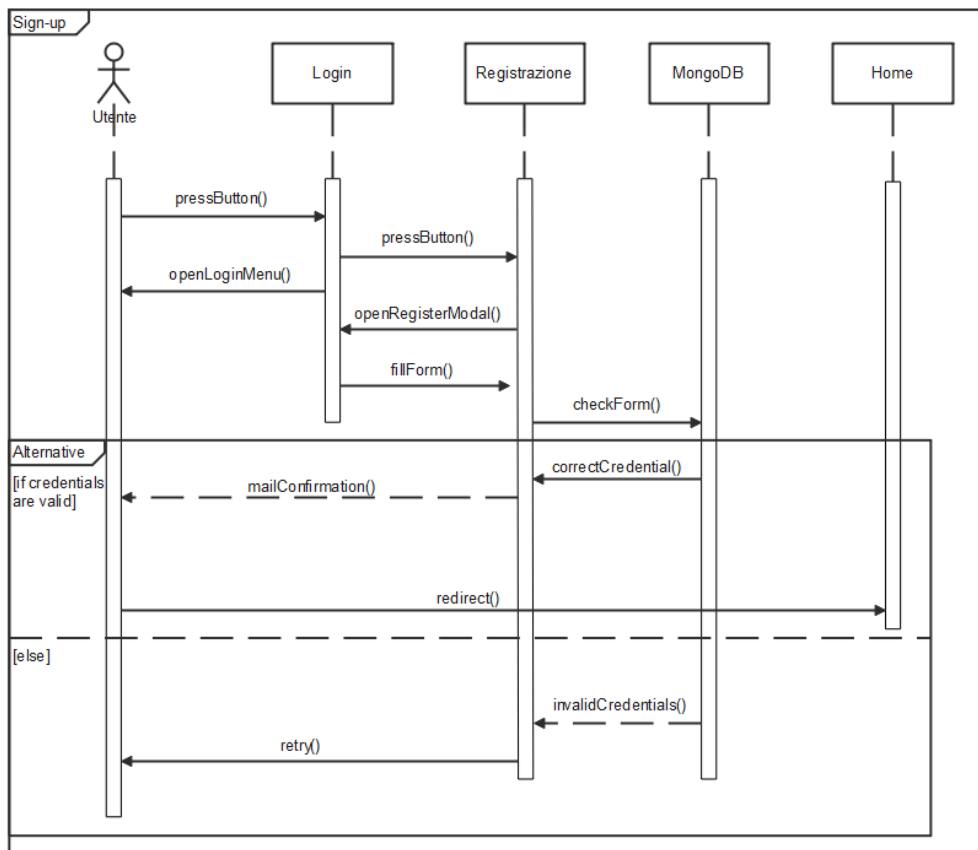
Il sistema è sviluppato utilizzando tecnologie come Python, Flask, MongoDB, HTML, CSS, e JavaScript.

L'interfaccia è stata resa minimale per consentirne l'utilizzo in modo del tutto intuitivo, richiamando icone comuni e conosciute con pattern di interazioni appartenenti al framework di Navbar, Navigation Drawer e Drop Down Menu.

2 - Sistema Corrente

Diagramma delle sequenze:





3 - Sistema Proposto

3.1 - Panoramica

Il sistema sviluppato è una WebApp per la vendita di videogiochi.

La piattaforma presenta due tipologie di utenze:

- Amministrative
- Consumer

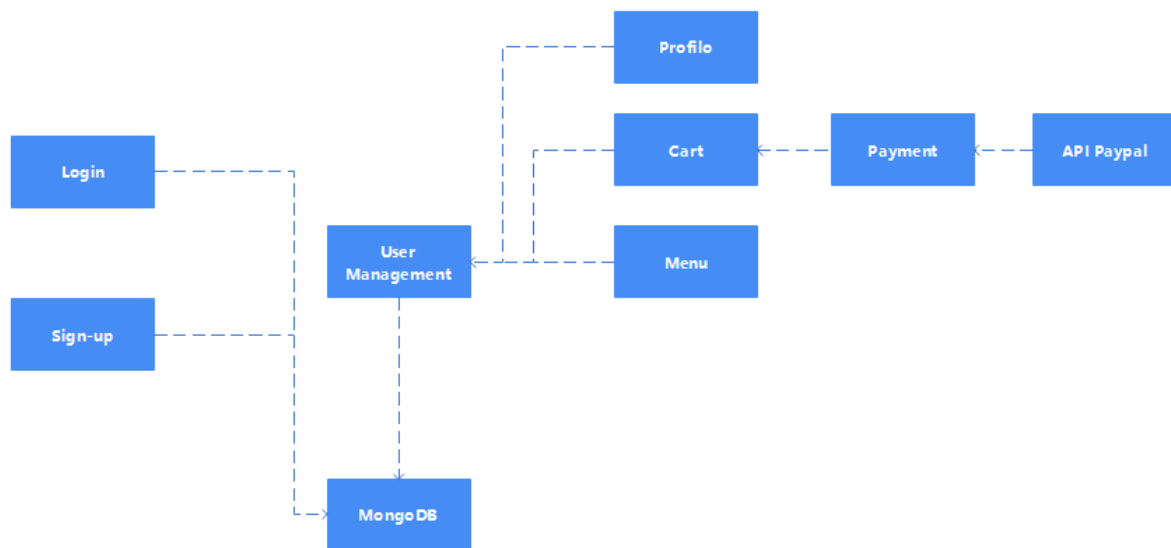
I consumatori sono appassionati di videogiochi alla ricerca di una nuova avventura digitale. Una vasta scelta di titoli e di piattaforme disponibili consentirà loro di avventurarsi nei generi da loro preferiti oppure potranno lasciarsi incuriosire da quelli inesplorati.

Gli utenti potranno:

- Registrarsi ed accedere al sito.
- Visionare la propria area personale.
- Visionare lo storico acquisti.
- Visionare il carrello.
- Aggiungere un gioco al carrello.
- Comprare un gioco.
- Cercare attraverso la barra di ricerca.
- Visualizzare i giochi a secondo della piattaforma.

Per gestire i pagamenti è stata implementata l'API Paypal in quanto ad oggi rappresenta il metodo di pagamento più diffuso ed affidabile per coloro che vogliono effettuare acquisti online.

3.2 - Decomposizione del sistema



User Management:

- **login**: Gestisce il processo di accesso degli utenti.
- **sign-up**: Gestisce la registrazione di nuovi utenti.
- **logout**: Gestisce la disconnessione degli utenti.
- **profilo**: Pagina per visualizzare informazioni sull'utente.

Database e MongoDB:

- Configurazione di Flask e PyMongo per connettersi a MongoDB.
- Interazione con il database per gestire le informazioni degli utenti (registrazione, accesso).

Cart:

- Pagina per visualizzare il carrello degli acquisti.

3.2 - Hardware/Software Mapping

L'applicazione segue un modello client/server, in cui il server svolge il ruolo di fornire le risorse e gestire la logica di back-end, mentre il client (browser web) interagisce con il server per ottenere e visualizzare le pagine web e inviare dati attraverso richieste HTTP.

1. Client:

- Il client è rappresentato dai browser web degli utenti che accedono al sito. Quando gli utenti interagiscono con l'applicazione attraverso il browser, inviano richieste HTTP al server.

2. Server:

- Il server è il componente che ospita l'applicazione. L'applicazione web basata su Flask gestisce le richieste provenienti dai client, elabora le informazioni richieste o invia dati al client come risposta.

3. Database:

- Il server utilizza MongoDB come database per archiviare e recuperare informazioni, ad esempio dettagli sugli utenti ed eventualmente altre informazioni correlate ai giochi.

4. Pagamenti:

- Per la gestione dei pagamenti, il server interagisce con l'API di PayPal. Quando un utente avvia il processo di pagamento, il server comunica con l'API di PayPal per creare un ordine e ottenere un URL di approvazione.

3.4 - Gestione dei dati persistenti

La gestione dei dati persistenti nel sistema è affidata a MongoDB, un database NoSQL. Flask-PyMongo facilita l'interazione tra il server Flask e il database, consentendo operazioni come l'inserimento e la ricerca degli utenti. Per garantire la sicurezza delle informazioni sensibili, il modulo Werkzeug è impiegato per eseguire l'hashing delle password degli utenti prima del salvataggio nel database. Questa pratica migliora la sicurezza complessiva del sistema, proteggendo le credenziali utente da accessi non autorizzati.

3.5 - Controllo accessi e sicurezza

Gli utenti avranno la possibilità di registrarsi e di effettuare l'accesso al sito attraverso email e password.

La sicurezza sarà garantita dalla cifratura delle credenziali di accesso attraverso l'implementazione di Werkzeug.security cioè un algoritmo di hashing SHA-256.