

Università degli Studi di Salerno

Corso di Ingegneria del Software

Project: PC Zone
System Design
Document
Versione 3.1



Data: 15/12/2022

Partecipanti

Nome	Matricola
Simone Scermino	0512110611
Roberto Andrei Miron	0512110581

Revision History

Data	Versione	Descrizione	Autore
11/11/2022	1.0	Inizio stesura SDD, indice	Simone Scermino, Roberto Andrei Miron
13/11/2022	1.1	Scopo del sistema, design goals	Simone Scermino, Roberto Andrei Miron
15/11/2022	2.0	Sistema attuale	Simone Scermino, Roberto Andrei Miron
20/11/2022	2.1	Sistema proposto, scomposizione sottosistema	Simone Scermino, Roberto Andrei Miron
23/11/2022	2.2	HW/SW Mapping, Boundary conditions	Simone Scermino, Roberto Andrei Miron
27/11/2022	2.3	Persistent Data management	Simone Scermino, Roberto Andrei Miron
02/12/2022	3.0	Access control, servizi dei sottosistemi	Simone Scermino, Roberto Andrei Miron
15/12/2022	3.1	Rifiniture tabelle	Simone Scermino, Roberto Andrei Miron

1. Introduction

1.1 Purpose of the system

1.2 Design goals

1.3 Definitions, acronyms, and abbreviations

1.4 References

1.5 Overview

2. Current software architecture

3. Proposed software architecture

3.1 Overview

3.2 Subsystem decomposition

3.3 Hardware/software mapping

3.4 Persistent data management

3.5 Access control and security

3.6 Global software control

3.7 Boundary conditions

4. Subsystem services

Glossary

1. Introduction

1.1 Purpose of the system

PC Zone si pone l'obiettivo di facilitare la scelta o la creazione di una configurazione per computer.

Lo scopo principale è quindi quello di unire tutte le informazioni inerenti a questa scelta in una piattaforma sola, senza disperdere l'attenzione ed il tempo dell'utente in varie guide di vari siti, o varie ricerche online.

Il sistema permette l'iscrizione sia di utenti, così da fornire accesso a tutte le principali funzionalità del sito, sia di gestori, per consentire un controllo continuo delle varie sezioni o delle componenti presenti nel catalogo.

Per facilitare l'uso di tale piattaforma la si è scelta di implementare come Web App, aggiungendo la responsività per permettere un uso anche da dispositivi mobile.

Le funzioni fornite dalla nostra piattaforma variano in base al tipo di utenza.

Funzioni inerenti il proprio profilo:

Guest

- Registrazione

Utente registrato

- Login
- Logout
- Ripristina password

Gestore

- Registrazione nuovo gestore
- Visualizza profilo

Utente

- Visualizza profilo
- Modifica profilo

Funzioni inerenti gli ordini/scelta componenti:

Guest, Utente Registrato

- Visualizza elenco guide
- Visualizza guida

- Visualizza catalogo
- Visualizza componenti

Utente registrato

- Ordina componente
- Aggiungi/rimuovi preferiti
- Assembla pc
- Aggiungi recensione a componente

Funzioni inerenti la gestione del sito:

Gestore

- Rimozione recensione
- Inserimento/modifica guida
- Modifica guida esistente

1.2 Design goals

In questa sezione, verranno esplicitati gli obiettivi di design che il sistema mira a raggiungere. La piattaforma sarà progettata per essere intuitiva e facile da utilizzare, attraverso un'interfaccia grafica semplice che evidenzia tutte le operazioni disponibili per l'utente. Inoltre, la piattaforma punta anche a ridurre gli errori che possono essere commessi durante l'utilizzo del sistema, minimizzando così la perdita di dati.

Alla fine di questa sezione, verrà presentata anche una tabella che elenca i diversi compromessi (trade-off). Essi sono necessari poiché non è possibile ottenere un sistema perfetto. Grazie a questi compromessi possiamo migliorare alcune caratteristiche del sistema a discapito di altre.

1.2.1 Criteri di usabilità

- **DG 1.1 - Facilità d'utilizzo**

Il sistema dovrà risultare facilmente comprensibile e utilizzabile anche da utenti meno esperti. Inoltre, è importante che il sistema possa essere utilizzato senza la necessità di consultare documentazioni, ovvero che permetta all'utente di iniziare a utilizzare la piattaforma in modo immediato senza incontrare problemi.

- **DG1.2 - Interfaccia intuitiva**

La piattaforma dovrà fornire un'interfaccia intuitiva per gli utenti, in modo che sia semplice e chiara, evidenziando le funzionalità disponibili

per gli utenti. Il layout dovrà essere organizzato in modo logico, con elementi di navigazione facilmente accessibili e riconoscibili.

- **DG1.3 - Interfaccia responsive**

L'interfaccia del sistema dovrà essere progettata per essere responsive, in modo che si adatti automaticamente alle diverse dimensioni dello schermo e tipologie di dispositivo utilizzati per visualizzare il contenuto.

- **DG1.4 - Segnalazioni di malfunzionamenti**

Il sistema dovrà anche fornire elementi di feedback per le azioni degli utenti, in modo che siano consapevoli di eventuali errori.

1.2.2 Criteri di performance

- **DG 2.1 - Navigazione concorrente**

Il sistema dovrà essere in grado di gestire una media di 100 utenti simultanei, senza compromettere le prestazioni della piattaforma.

- **DG 2.2 - Tempi di risposta**

Il sistema dovrà garantire tempi di risposta non superiori a 5 secondi.

1.2.3 Criteri di affidabilità

- **DG 3.1 - Operazioni autorizzate per ruolo**

Il sistema dovrà garantire una divisione tra i vari ruoli della piattaforma. Le risorse e i dati dovranno essere protetti dal possibile utilizzo o accesso da soggetti non autorizzati.

- **DG 3.2 - Integrità operazioni**

Il sistema deve garantire l'integrità delle operazioni assicurandosi che siano eseguite interamente con successo.

- **DG 3.3 - Controllo input**

Il sistema dovrà assicurarsi che tutti i dati inseriti dagli utenti siano corretti attraverso dei controlli.

- **DG 3.3 - Availability**

Il sistema dovrà essere disponibile agli utenti tutte le ore e tutti i giorni senza nessuna limitazione. Nel caso in cui il sistema deve essere portato offline per un breve lasso di tempo, verranno portate offline solo le funzioni che dovranno essere modificate.

- **DG 3.3 - Sicurezza**

In un eventuale crash del sistema gli unici dati persi dovranno essere gli accessi al sistema. In quest'ultimo caso l'utente dovrà accedere alla

piattaforma. Inoltre, il sistema deve garantire che tutti i dati vengano salvati in maniera sicura.

- **DG 3.4 - Privacy**

Il sistema deve essere conforme alle leggi sulla privacy in vigore, in particolare il Decreto Legislativo n. 101/18 del 10 agosto 2018, noto anche come "GDPR", che è entrato in vigore il 19 settembre 2018 e stabilisce le regole per la protezione dei dati personali.

1.2.4 Criteri di manutenzione

- **DG 4.1 Modificabilità**

Il sistema deve essere progettato in modo tale che le funzioni esistenti possano essere facilmente modificate senza influire sugli altri componenti, al fine di rendere più efficienti e veloci i tempi di manutenzione della piattaforma.

- **DG 4.2 Portabilità**

Il sistema dovrà essere in grado di funzionare correttamente anche quando viene trasferito su hardware più avanzato o su nuove versioni del SO.

- **DG 4.3 Implementazione**

Il sistema dovrà integrare tutte le funzionalità descritte nella fase di analisi e soddisfare tutti i requisiti previsti.

Trade-Offs

Trade-off	Descrizione
Performance vs Scalabilità	Migliorare la concorrenza del sistema, e quindi il numero di utenti simultanei che possono accedervi influenzerebbe sulla performance del sito quindi il sistema dovrà dare precedenza alla Performance.
Complessità vs Semplicità	Un design complesso può fornire più funzionalità e flessibilità ma rende il sistema più difficile da capire, testare e mantenere. Il sistema dovrà dare la precedenza alla semplicità e quindi essere intuitivo e facile da usare per l'utenza meno esperta.

Costo vs Qualità	Utilizzare tecnologie costose garantisce una migliore qualità del sistema ma aumenta i costi del sistema e della manutenzione. Un sistema che usa tecnologie meno costose può ridurre i costi ma può anche compromettere il sistema. Nell'ambito della piattaforma, si è deciso di dare la precedenza al risparmio dei costi.
End User Criteria vs Funzionalità	Un design accattivante e un'interfaccia intuitiva migliora l'esperienza per l'utente ma aumenta anche il tempo di sviluppo e il costo. Un design meno accattivante invece, può permettere di aumentare le funzionalità della piattaforma. Il sistema in questione darà la precedenza alle funzionalità nel caso in cui non si riuscirà ad implementare tutto entro i limiti di tempo.
Usabilità vs Manutenzione	Un design altamente user-friendly può richiedere più tempo per essere sviluppato e può essere più difficile da essere mantenuto. Il sistema cercherà di bilanciare entrambe ma darà la precedenza all'usabilità.

1.3 Definitions, acronyms, and abbreviations

Acronimi:

- RAD: Requirements Analysis Document
- SDD: System Design Document

Abbreviazioni:

- DB: DataBase
- DBMS: DataBase Management System

1.4 References

- B. Bruegge, A. H. Dutoit, Object Oriented Software Engineering - Using UML, Pattern and Java, Prentice Hall, 3rd edition, 2009
- Documento RAD di progetto

1.5 Overview

Il documento è organizzato in quattro punti principali:

- 1) **Introduzione:** introduce sommariamente gli obiettivi del sistema, i design goals, le definizioni e gli acronimi utilizzati ed eventuali riferimenti ad altre fonti utilizzate.
- 2) **Architettura del sistema attuale:** descrive lo stato dell'eventuale sistema attuale per effettuare un confronto con quello proposto.
- 3) **Architettura del sistema proposto:** definisce il partizionamento del sistema in sottosistemi, il mapping HW/SW, la gestione dei dati persistenti, le condizioni limite del sistema.
- 4) **Servizi dei sottosistemi:** presentazione dei servizi offerti dai vari sottosistemi.

Alla fine del documento è presente un **glossario**, per elencare eventuali termini tecnici con relative definizioni

2. Current software architecture

Al momento non è disponibile nessuna piattaforma che unisca tutte le funzionalità di PC Zone in un unico servizio, di conseguenza non è possibile effettuare un confronto con la nostra architettura.

3. Proposed software architecture

3.1 Overview

Il nostro sistema prevede l'uso di una architettura three-tier, mantenendo la classica suddivisione in presentation, application, data tier (per preservare vantaggi quali leggibilità, manutenzione, riuso).

Nel presentation tier saranno presenti pagine che dovranno essere presentate all'utente tramite il browser da lui usato, e saranno realizzate con tecnologie quali HTML, CSS, JS.

Nell'application tier avremo il controllo della logica applicativa, presente lato Server ed implementata tramite Java.

Nel data tier avremo l'uso di un Database relazionale SQL, gestito tramite il DBMS MySQL, al quale verrà effettuato l'accesso tramite apposito driver offerto da JDBC.

3.2 Subsystem decomposition

Come suddivisione iniziale dei sottosistemi, abbiamo pensato di rispettare la categorizzazione data dal tipo di architettura applicata, quindi avremo:

- **livello Presentation**, contenente l'interfaccia grafica del sistema e tutti gli elementi che consentono all'utente di interagire con il sistema;
- **livello Application**, contenente tutto ciò che gestisce input dati dall'utente o funzionalità richieste, fornendo risultati e passandoli al livello Presentation per poterli presentare in modo chiaro;
- **livello Data**, contenente la parte persistente di dati del sistema.

Successivamente si è suddiviso il sistema in **GestioneUtente**, **GestioneSito**, **GestioneGestore**, **InterfacciaStorage**, **Database**.

Nello specifico:

- **GestioneUtente** si occupa di tutto ciò che riguarda il profilo, quindi registrazione/login dell'utente, visualizzazione e modifica dei propri dati, visualizzazione dei favoriti, logout.
- **GestioneSito** si occupa di tutte le funzionalità presenti nel nostro sito, quindi visualizzazione di componenti, guide, post, segnalazione di bug, aggiunta di commenti/post/recensioni.
- **GestioneGestore** si occupa di tutte le funzionalità accessibili solo dal gestore, come registrazione di altri gestori, modifica/aggiunta di componenti/guide, rimozione di commenti/post/recensioni.
- **InterfacciaStorage**, che si pone come collegamento, o meglio interfaccia, tra la logica applicativa e il DBMS. Questo comporta la minimizzazione del coupling tra sottosistemi e data tier, incrementando leggermente la complessità di implementazione. Modificare l'interfaccia risulterà molto più semplice.
- **Database**, che gestisce i dati e la loro persistenza tramite il DBMS.

Da precisare che ogni sottosistema avrà sia componenti nel livello Presentation (componenti GUI per presentare all'utente), che componenti nel livello Application (logica effettiva che gestisce il tutto).

Diagramma delle componenti

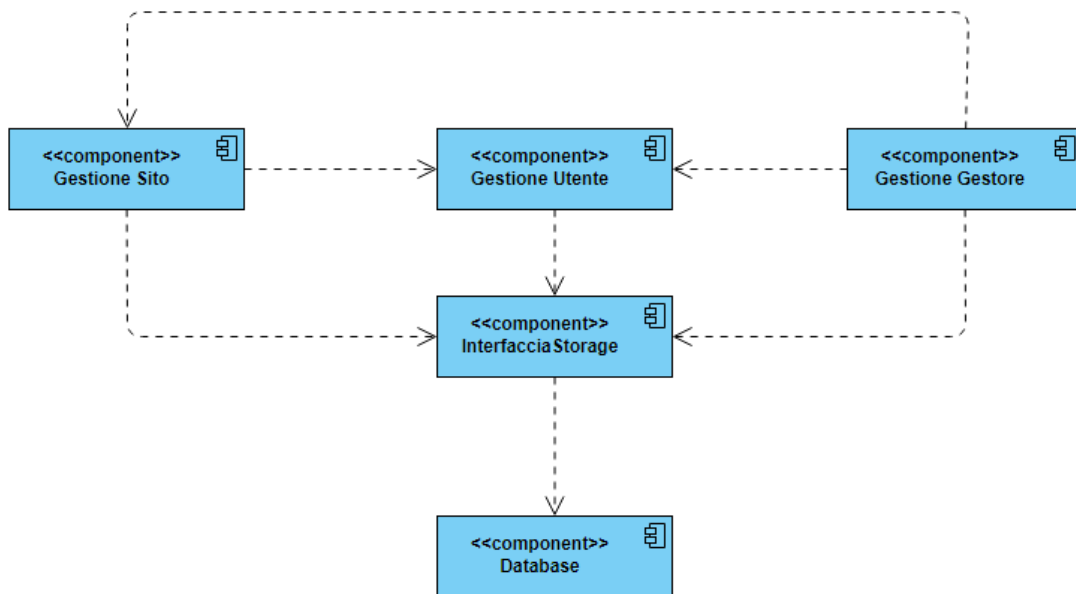
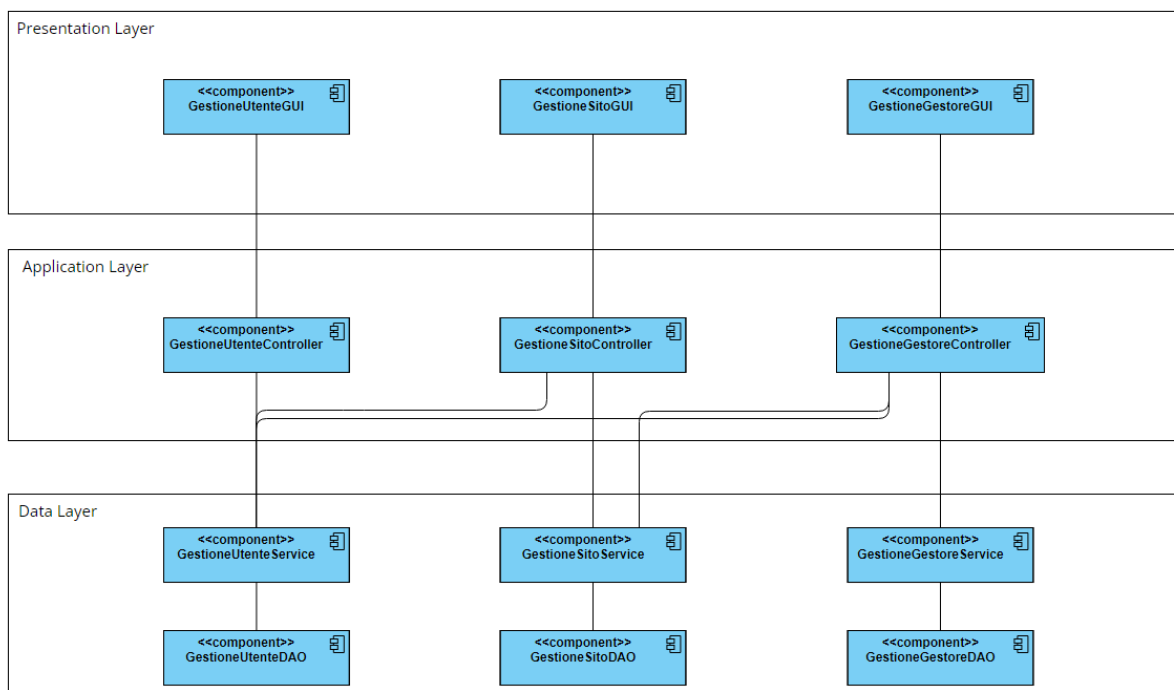
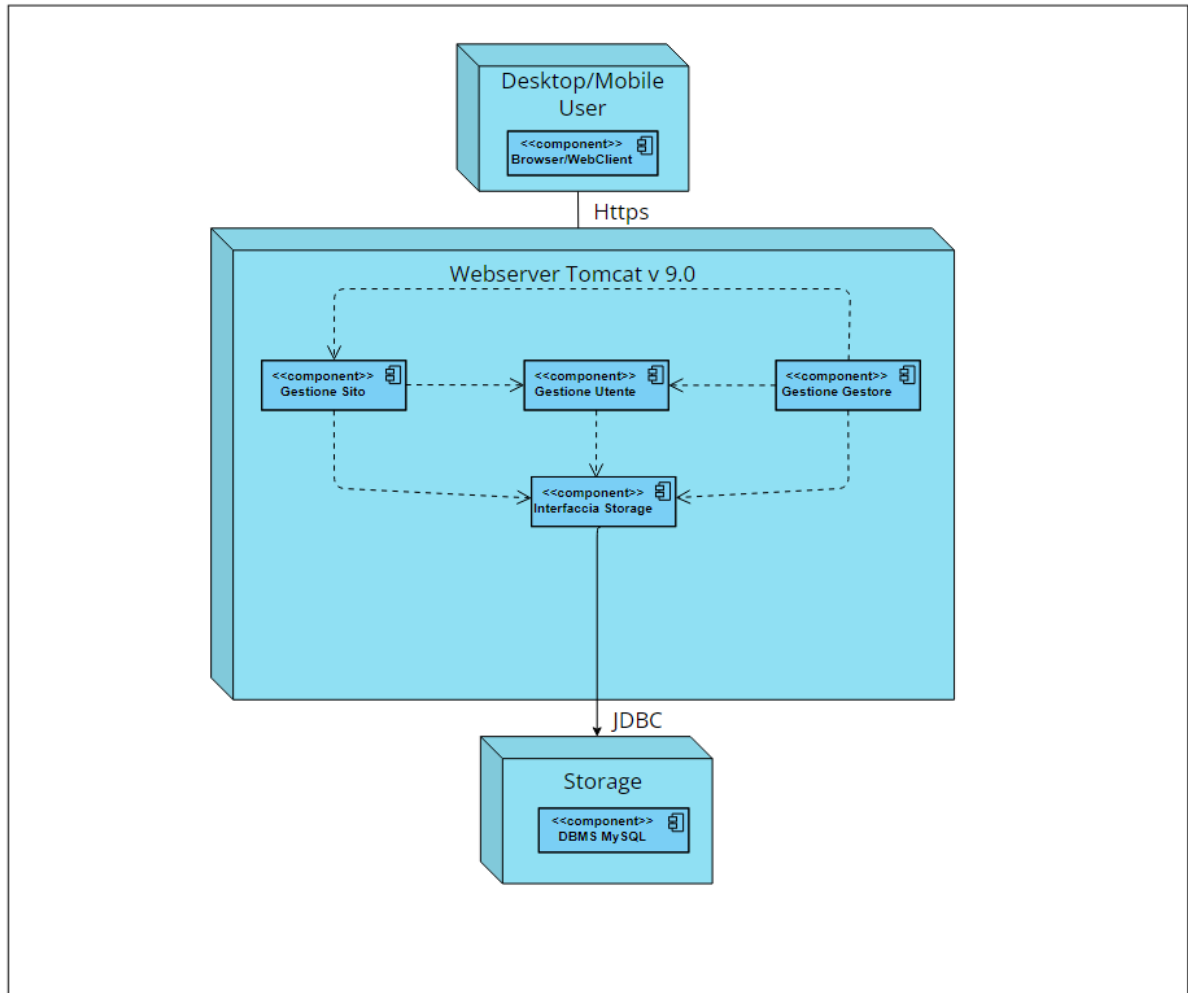


Diagramma architetturale



3.3 Hardware/software mapping



Il sistema sarà organizzato in una architettura a tre livelli, usata tipicamente per la creazione di Web Application; essa è composta da presentation, application, data tier.

Il client è qualsiasi sia il dispositivo con il quale si accede al sistema (ovviamente dotato di un browser ed una connessione ad internet), e si occuperà di presentare all'utente l'interfaccia grafica fornita dal sistema.

Le richieste verranno generate con il protocollo HTTPS, e le eventuali pagine HTML ricevute verranno gestite e mostrate dal browser in uso.

Nel server abbiamo la presenza di un web server, componente che processa le richieste ricevute dal client, collegandosi se necessario alla parte che si occupa di gestire i dati permanenti (DBMS, tramite JDBC).

Il data tier è composto da un database relazionale SQL gestito con il DBMS MySQL.

La scelta di questa architettura porta molti vantaggi: distinzione dei vari tier, consentendo un intervento mirato alle eventuali componenti da modificare; migliore flessibilità del sistema.

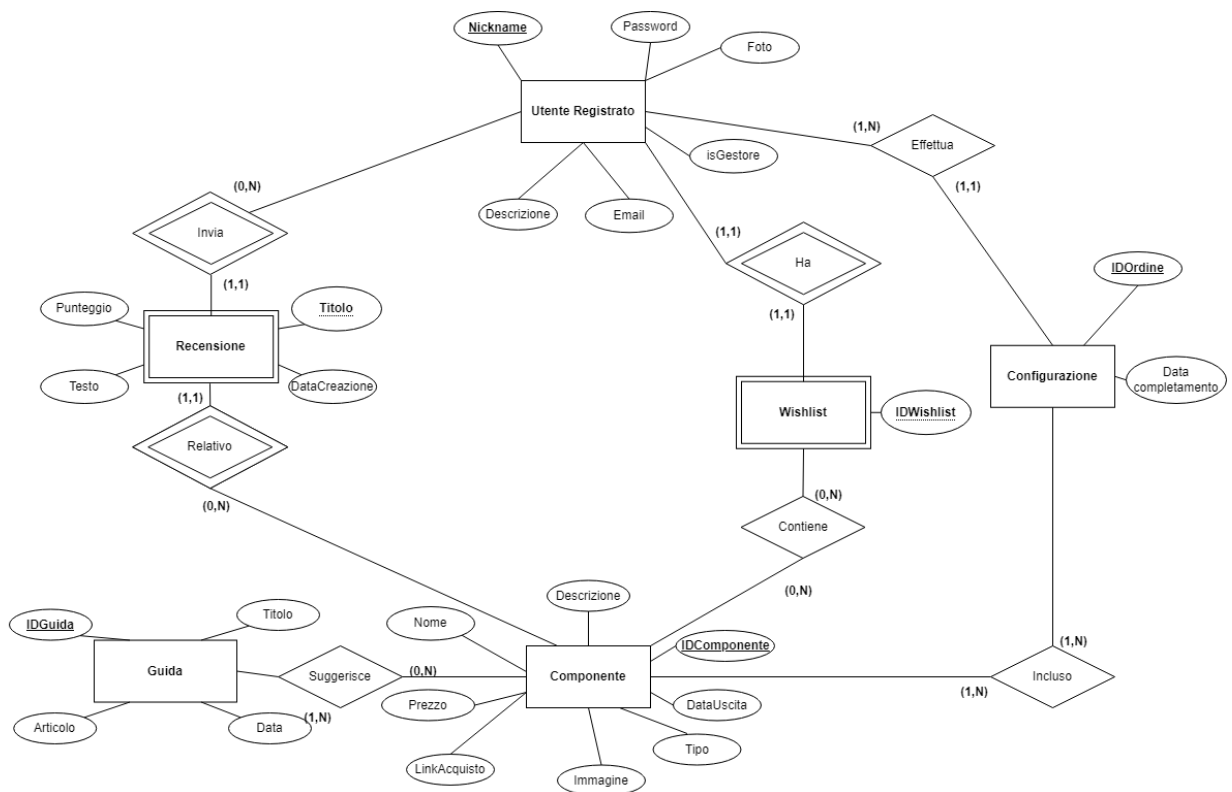
3.4 Persistent data management

Per salvataggio e gestione dei dati, si è scelto di renderli persistenti tramite l'uso dei database relazionali. Tra alcuni dei principali motivi di tale scelta abbiamo:

- Consistenza dei dati grazie all'utilizzo di vincoli, elemento molto utile nella modifica di entità contenenti relazioni con altre esterne.
- Protezione dei dati, possibile grazie alla funzionalità in MySQL di permettere un accesso protetto ai dati, limitando l'accesso degli Utenti senza determinate autorizzazioni.
- Transazioni, operazioni atomiche presenti in MySQL strutturate in modo che il codice al loro interno venga eseguito o per intero o per nulla, senza mezzi termini, garantendo così una più efficiente consistenza dei dati.
- Affidabilità dei dati, i DBMS permettono di effettuare dei backup, ovvero delle copie dei dati attualmente presenti nel database, cosicché in caso di malfunzionamenti hardware o software sarà possibile ripristinare lo stato precedente.

Nello specifico si è optato per MySQL come DBMS anche per l'esperienza dei partecipanti al progetto con esso.

ER Diagram



3.5 Access control and security

Nella tabella sono riportati Attori e Sottosistemi (rispettivamente su colonne e righe), così da indicare quali servizi vengono offerti da essi ai vari attori.

InterfacciaStorage e Database sono stati omessi, dato il loro accesso riservato solo agli oggetti stessi e non agli attori.

GestioneUtente

	Guest	Utente registrato	Gestore
GestioneUtente	registraUtente() login() loginGestore() logout()	login() logout() ripristinaPassword() visualizzaProfilo() modificaInformazioni()	login() loginGestore() logout()

GestioneSito

	Guest	Utente registrato	Gestore
GestioneSito	visualizzaComponente() segnalaBug() visualizzaCatalogo() visualizzaElencoGuide() visualizzaElencoFavoriti() visualizzaHomepage()	recensisciComponente() visualizzaGuida() aggiungiFavoriti() assemblaPC() rimuoviFavoriti() visualizzaComponente() segnalaBug() visualizzaCatalogo() visualizzaElencoGuide() visualizzaElencoFavoriti() visualizzaHomepage()	visualizzaComponente() segnalaBug() visualizzaCatalogo() visualizzaElencoGuide() visualizzaElencoFavoriti() visualizzaHomepage()

GestioneGestore

	Gestore
GestioneGestore	inserisciComponente() rimuoviRecensione() inserisciGuida() modificaGuida() modificaComponente() registrazioneGestore()

3.6 Global software control

Il meccanismo di controllo del nostro sistema è di tipo event-driven.

L'utente avrà accesso ad una interfaccia grafica, nella quale potrà selezionare la funzionalità da avviare, mandando quindi una richiesta all'handler specifico; sarà esso ad occuparsi della sua gestione, dirigendo la richiesta al sottosistema che si occupa della logica di controllo per elaborare tale funzionalità.

Al termine della gestione, da parte dell'handler, della richiesta specifica, al browser del client verrà spedita la risposta alla funzionalità scelta, permettendo di continuare la navigazione nel sistema.

3.7 Boundary conditions

Le boundary conditions identificate nel nostro sistema sono inerenti all'avvio di esso, il suo spegnimento, l'avvio in seguito ad un fallimento, l'errato accesso ai dati persistenti.

Primo avvio

Prima di effettuare il primo avvio del sistema, lo sviluppatore che sta effettuando l'installazione di esso sulla macchina dovrà inserire nel Database MySQL (dopo averne avviato i servizi) almeno un profilo di un gestore, condividendo con esso i dati relativi al login. Questo è necessario in quanto, nella nostra piattaforma, i gestori possono essere registrati solo da altri gestori. Una volta terminata questa operazione, sarà necessario avviare il web server sulla macchina che lo contiene.

I seguenti avvii del sistema comporteranno soltanto l'avvio del servizio DBMS e del web server.

Avvio in caso di fallimento

In caso di errori che portano all'interruzione del sistema improvvisa, la gestione del DBMS tramite transazioni mantiene la consistenza delle informazioni evitando il salvataggio di eventuali modifiche in corso.

Non sono previste soluzioni per ripristinare lo stato precedente all'arresto improvviso, ma le uniche perdite saranno informazioni di sessione (quindi login attuali di gestori/utenti, elementi negli ordini degli utenti).

Spegnimento

Per effettuare lo spegnimento del sistema ci basterà terminare l'esecuzione del web server. Esso infatti gestisce anche il collegamento con il DBMS, di conseguenza non avremo bisogno di dover effettuare manualmente la sua chiusura.

Per mantenere consistenza nei dati persistenti, eventuali transazioni ancora attive non verranno portate a termine, annullando eventuali modifiche in corso.

Errato accesso ai dati persistenti

In caso di eventuali errori nella costruzione del DB, o errori nell'uso del driver per collegarsi ad esso, verrà mostrata a schermo una pagina di errore realizzata dal web server, nella quale verrà descritta la fonte dell'errore e dove esso sarà avvenuto nel codice sorgente.

Lo sviluppatore dovrà risolvere il problema per poi effettuare nuovamente l'avvio del sistema, che si comporterà come di norma in caso di mancanza di altri errori.

4. Subsystem services

Di seguito vengono riportati gli elenchi di servizi offerti dai sottosistemi identificati precedentemente.

Servizi di GestioneUtente

Servizio	Descrizione
registraUtente()	Consente all'utente di registrarsi al sistema.
login()	Consente all'utente di effettuare il login al sistema.
loginGestore()	Consente al gestore di effettuare il login al sistema.
logout()	Consente di effettuare il logout dal sistema.
modificaInformazioni()	Consente di modificare le informazioni del proprio profilo.
visualizzaProfilo()	Consente di visualizzare le informazioni del proprio profilo.
ripristinaPassword()	Consente di accedere alla funzione di ripristino password.

Servizi di GestioneSito

Servizio	Descrizione
recensisciComponente()	Consente all'utente di scrivere una recensione per una componente.
visualizzaGuida()	Consente di visualizzare le informazioni riguardanti la guida selezionata.

aggiungiFavoriti()	Consente all'utente di aggiungere una componente ai favoriti.
visualizzaComponente()	Consente di visualizzare le informazioni riguardanti la componente selezionata.
rimuoviFavoriti()	Consente all'utente di rimuovere una componente dai favoriti.
assemblaPC()	Consente di visualizzare la lista delle componenti selezionate per il PC.
segnalaBug()	Consente all'utente di segnalare un bug.
visualizzaCatalogo()	Consente di visualizzare tutte le componenti presenti nel sistema.
visualizzaElencoGuide()	Consente di visualizzare tutte le guide presenti nel sistema.
visualizzaElencoFavoriti()	Consente di visualizzare l'elenco delle componenti che l'utente ha aggiunto ai favoriti.
visualizzaHomepage()	Consente di visualizzare la homepage del sistema.

Servizi di GestioneGestore

Servizio	Descrizione
inserisciComponente()	Consente di inserire una nuova componente nel catalogo.
rimuoviRecensione()	Consente di rimuovere una recensione da una componente.
inserisciGuida()	Consente di inserire una nuova guida nell'elenco guide.
modificaGuida()	Consente di modificare una guida esistente.

modificaComponente()	Consente di modificare una componente esistente.
registrazioneGestore()	Consente di registrare un nuovo gestore nel sistema.

Glossary

Terminologia	Definizione
Sistema	Base software o hardware su cui sono sviluppate o eseguite applicazioni.
Three-Tier	Architettura software composta da tre strati disaccoppiati. Gli strati sono: Presentazione, Logica, Persistenza
DBMS	Un Database Management System è un sistema software progettato per consentire la creazione, la manipolazione e l'interrogazione efficiente di database.
MySQL	DBMS relazionale open source, utilizza SQL come linguaggio.
Diagramma ER	Modello utilizzato per la rappresentazione grafica dei dati. In questo modello vengono definite entità e relazioni tra le entità.
Bug	Anomalia riscontrata durante l'utilizzo del sistema. Utile da segnalare per migliorare la permanenza su esso.