



SDD

System Design Document

Riferimento	C033_SDD_ver.2.0
Versione	2.0
Data	14/02/2025
Destinatario	Prof. C. Gravino
Presentato da	C033

Revision History

Data	Versione	Descrizione	Autori
15/11/2024	0.1	Prima stesura	EF
22/11/2024	0.2	Individuazione dei Sottosistemi e dei loro Sottosistemi	TUTTI
29/11/2024	0.3	Design Goal e Trade-off	TUTTI
06/12/2024	0.4	Diagramma Architeturale	EF - PS
13/12/2024	0.5	Descrizione e Scopo del Sistema	CI
20/12/2024	1.0	Entity Class Diagram Ristrutturato	EF - PS
27/12/2024	1.1	Mapping Hardware-Software	PS - CI
03/01/2025	1.2	Schema ER del Database	PS - CI
10/01/2025	1.3	Attributi per Entità in Forma Tabellare	EF - AD
17/01/2025	1.4	Controllo degli Accessi e Sicurezza	EF - AD
24/01/2025	1.5	Diagramma Sottosistemi Specificati	TUTTI
31/01/2025	1.6	Individuazione e Realizzazione Casi Limite	PS - AD - EF
07/02/2025	1.7	Design Pattern Façade	AD
14/02/2025	2.0	Revisione Finale	TUTTI

Team members

Nome	Ruolo nel progetto	Acronimo	Informazioni di contatto
Alessandro Dell’Atti	Team Member	AD	a.dellatti@studenti.unisa.it
Carmine Ioanna	Team Member	CI	c.ioanna@studenti.unisa.it
Eustachio Fontana	Team Member	EF	e.fontana4@studenti.unisa.it
Pietro Somma	Team Member	PS	p.somma15@studenti.unisa.it

Sommario

- 1 Introduzione.....
 - 1.1 Scopo del Sistema.....
 - 1.2 Obiettivi di Design (Design Goals).....
 - 1.3 Definizioni, acronimi, e abbreviazioni.....
 - 1.4 Riferimenti.....
 - 1.5 Organizzazione del Documento.....
- 2 Architettura del Sistema Corrente.....
- 3 Architettura del Sistema Proposto.....
 - 3.1 Panoramica sulla Sezione.....
 - 3.2 Decomposizione in Sottosistemi.....
 - 3.3 Mapping hardware/software.....
 - 3.4 Gestione dei dati persistenti.....
 - 3.5 Controllo degli accessi e sicurezza.....
 - 3.6 Controllo globale del software.....
 - 3.7 Condizioni limite.....
- 4 Servizi dei Sottosistemi.....
- 5 Design Pattern.....
- 6 Glossario.....

1 Introduzione

1.1 Scopo del Sistema

Il sistema che si vuole realizzare ha come obiettivo principale quello di facilitare la gestione del guardaroba degli utenti attraverso una piattaforma online.

Wearly permetterà di inserire i capi di abbigliamento, associando a ciascuno uno stato (come “in armadio” o “a lavare”), per una gestione semplificata e visiva del guardaroba.

L'obiettivo principale è quello di aiutare gli utenti a comprendere a colpo d'occhio quali capi siano disponibili, facilitando la scelta di cosa indossare in una determinata situazione.

1.2 Obiettivi di Design (Design Goals)

Nella presente sezione si andranno a presentare i Design Goals, ovvero le qualità sulle quali il sistema deve essere focalizzato, formalizzati esplicitamente così che qualsiasi importante decisione di design può essere fatta consistentemente seguendo lo stesso insieme di design goal.

Seguendo le linee guida del libro Bernd Bruegge – Object Oriented Software Engineering i design goal sono stati suddivisi nelle seguenti categorie:

- **Performance:** includono i requisiti di spazio e velocità imposti sul sistema.
- **Dependability:** determinano quanto sforzo deve essere speso per minimizzare i fallimenti del sistema (crash, falle di sicurezza) e le loro conseguenze.
- **Maintenance:** determina quanto sforzo è necessario per modificare il sistema dopo il suo rilascio.
- **End User:** includono qualità che sono desiderabili dal punto di vista dell'utente, ma che non sono state coperte dai criteri di Performance e Dependability.

Ciascun design goal è descritto da:

- **Rank**, che ne specifica un valore di priorità compreso tra 1 e 8 (1 massima e 8 minima).
- **ID Design Goal**, un identificatore univoco e un nome esplicativo.
- **Descrizione**, una descrizione del design goal.
- **Categoria**, ovvero la categoria di appartenenza del design goal.
- **RNF di origine**, che specifica il requisito non funzionale da cui il design goal è stato creato.

Rank	ID Design Goal	Descrizione	Categoria	RNF di origine
1	DG_1: Tempi di risposta	La velocità del sistema è essenziale per garantire un'esperienza fluida agli utenti. Un sistema lento potrebbe scoraggiare gli utenti e compromettere l'usabilità. Per questo, i tempi di risposta sono stati considerati la priorità assoluta.	Performance	RNF_P_5
2	DG_2: Affidabilità delle operazioni	Gli utenti devono avere fiducia nel sistema, soprattutto per operazioni sensibili come aggiungere, modificare o eliminare capi d'abbigliamento. Gli errori frequenti potrebbero portare alla perdita di dati o alla frustrazione degli utenti, rendendo l'affidabilità fondamentale.	Dependability	RNF_A_1
3	DG_3: Protezione dei dati utente	La sicurezza dei dati è una priorità elevata per garantire conformità alle normative (es. GDPR) e mantenere la fiducia degli utenti.	Dependability	RNF_A_2
4	DG_4: Scalabilità	Man mano che il numero di utenti e di dati cresce, il sistema deve continuare a funzionare senza perdita di prestazioni.	Performance	RNF_P_3

5	DG_5: Usabilità del sistema	La semplicità e l'intuitività sono importanti per favorire l'adozione del sistema da parte degli utenti.	End-User	RNF_U_4
6	DG_6: Manutenibilità	La capacità di aggiornare o correggere il sistema in modo rapido è essenziale per affrontare eventuali bug o modifiche.	Maintenance	RNF_S_1
7	DG_7: Disponibilità del servizio	Garantire un uptime elevato è fondamentale per evitare interruzioni del servizio.	Dependability	RNF_P_4
8	DG_8: Prestazioni sotto carico	La capacità del sistema di continuare a mantenere prestazioni alte anche con molti utenti connessi contemporaneamente.	Performance	RNF_P_2

Trade-off:

Trade-Off	Descrizione
Tempi di Risposta vs Costi	Nel processo di sviluppo del sistema, si è scelto di dare priorità alle prestazioni e alla velocità di risposta, ritenendole aspetti fondamentali per garantire un'esperienza ottimale. Questa decisione è stata presa consapevolmente, accettando l'incremento dei costi derivanti dall'implementazione di soluzioni tecniche più avanzate e performanti.
Sicurezza vs Usabilità	Nel processo di sviluppo del sistema, si è scelto di privilegiare la sicurezza, ritenendola un aspetto prioritario rispetto ad altri fattori, inclusa l'usabilità. Questa decisione è stata presa con l'obiettivo di garantire un elevato livello di protezione, anche a costo di introdurre misure che potrebbero rendere l'interazione con il sistema meno immediata o intuitiva.

1.3 Definizioni, acronimi, e abbreviazioni

Vengono riportati di seguito alcune definizioni presenti nel documento corrente:

- **Sottosistema:** un sottoinsieme dei servizi del dominio applicativo, formato da servizi legati da una relazione funzionale.
- **Design Goal:** le qualità sulle quali il sistema deve essere focalizzato.
- **Dati Persistenti:** dati che sopravvivono all'esecuzione del programma che li ha creati e che dunque vengono salvati.
- **Mapping Hardware/Software:** studio della connessione tra parti fisiche e logiche di cui si compongono il sistema.
- **SDD:** System Design Document
- **RAD:** Requirements Analysis Document

1.4 Riferimenti

Di seguito una lista di riferimenti ad altri documenti utili durante la lettura:

- [Requirements Analysis Document;](#)
- [Test Case Specification;](#)
- [Test Incident Report;](#)
- [Test Plan;](#)
- [Test Summary Report;](#)

1.5 Organizzazione del Documento

Il presente documento di System Design consiste di quattro sezioni:

Introduzione: Viene descritto in generale lo scopo del sistema, gli obiettivi di design che il sistema propone di raggiungere.

Architettura software corrente: Viene descritto lo stato attuale dell'architettura del software già presente.

Architettura software proposta: Viene descritto come il sistema sarà definito e partizionato in sottosistemi, il loro mapping Hardware/Software, la gestione dei dati persistenti. Verranno poi presentate la struttura dei singoli sottosistemi e le boundary conditions riguardanti l'intero sistema.

Glossario: Contiene la lista dei termini usati nel documento con annessa spiegazione.

2 Architettura del Sistema Corrente

Al momento, non esiste alcun software che condensi l'interezza delle funzionalità di Wearly in un unico servizio. Il mercato delle possibili alternative a questo software è pertanto incredibilmente frammentato e non esiste una reale architettura a cui è possibile confrontare in maniera ragionevole il sistema.

3 Architettura del Sistema Proposto

3.1 Panoramica sulla Sezione

Il sistema proposto è basato sullo stile architetturale Three Tier.

Il motivo della presente scelta è che tale architettura è perfetta per lo sviluppo di web application come la nostra, poiché la separazione della logica di presentazione da quella di elaborazione, migliora una serie di qualità, tra le quali:

- Leggibilità
- Manutenzione
- Riutilizzo

Nello sviluppo del sistema verranno usati HTML5, CSS3 e Javascript per la parte di front-end e la generazione delle view.

Per la logica applicativa e quindi il back-end sarà utilizzato Java.

Per la gestione del database saranno usati:

- MySQL come software per lo sviluppo del database in linguaggio SQL

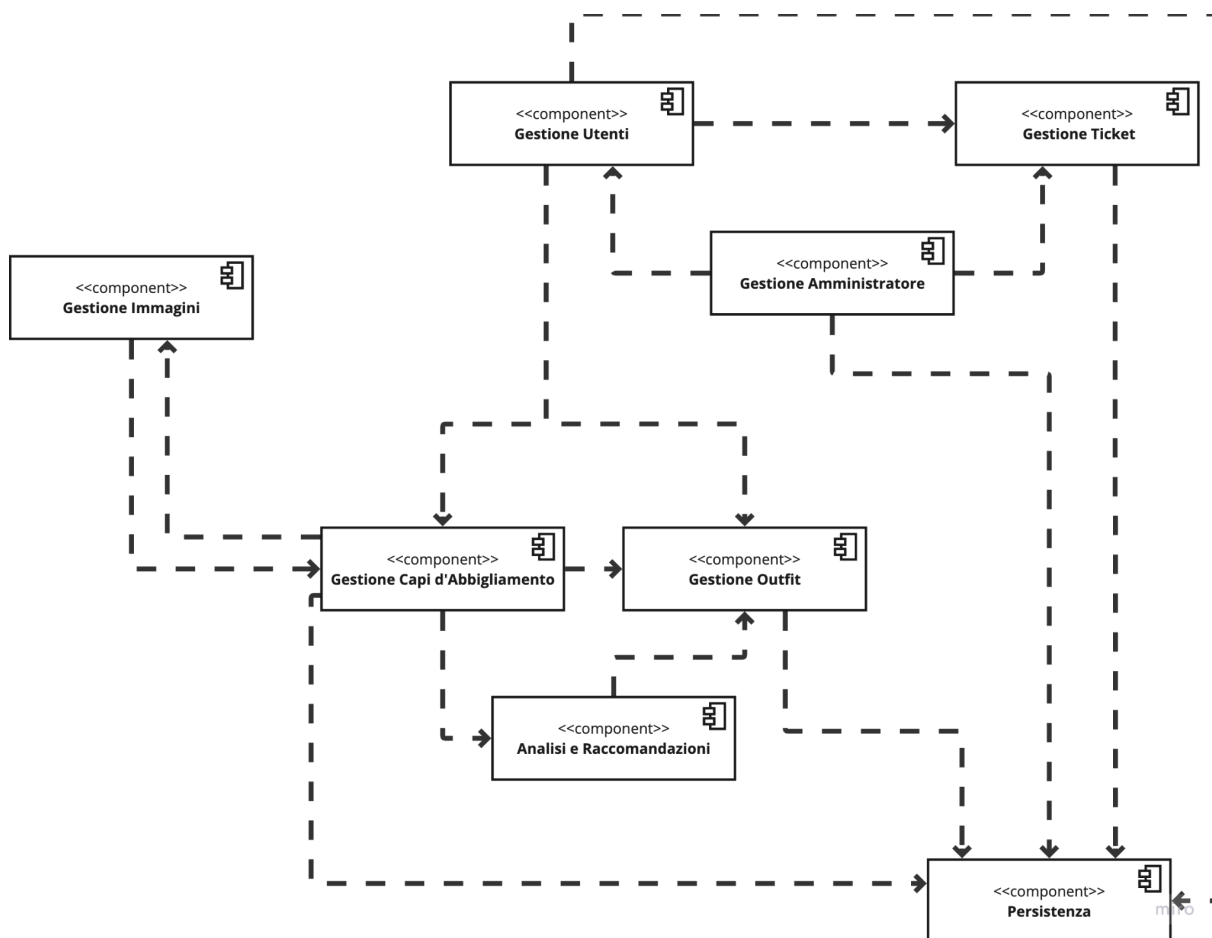
3.2 Decomposizione in Sottosistemi

- **Gestione Utente:** Responsabile della registrazione, autenticazione e gestione dei dati personali degli utenti.
- **Gestione Amministratore:** Responsabile della gestione degli utenti.
- **Gestione Capi d'Abbigliamento:** Gestisce il caricamento, modifica, visualizzazione dei capi di abbigliamento nel guardaroba digitale. Controlla la validità dei dati inseriti e memorizza le immagini associate ai capi.
- **Gestione Outfit:** Permette di creare, cancellare e modificare outfit combinando i capi presenti nel guardaroba.
- **Analisi e Raccomandazioni:** Utilizza algoritmi di AI e/o filtri per suggerire

outfit sulla base del guardaroba di ogni utente.

- **Gestione Ticket:** Consente agli utenti di fornire ticket di assistenza per gli amministratori, ed agli amministratori la possibilità di gestirli.
- **Gestione Immagini:** Gestisce il caricamento, compressione e memorizzazione delle immagini dei capi e degli outfit. Può essere un modulo separato per garantire scalabilità.
- **Persistenza:** Si occupa di gestire la persistenza dei dati con un database.

Sono mostrate di seguito le dipendenze tra i sottosistemi attraverso un component diagram UML.

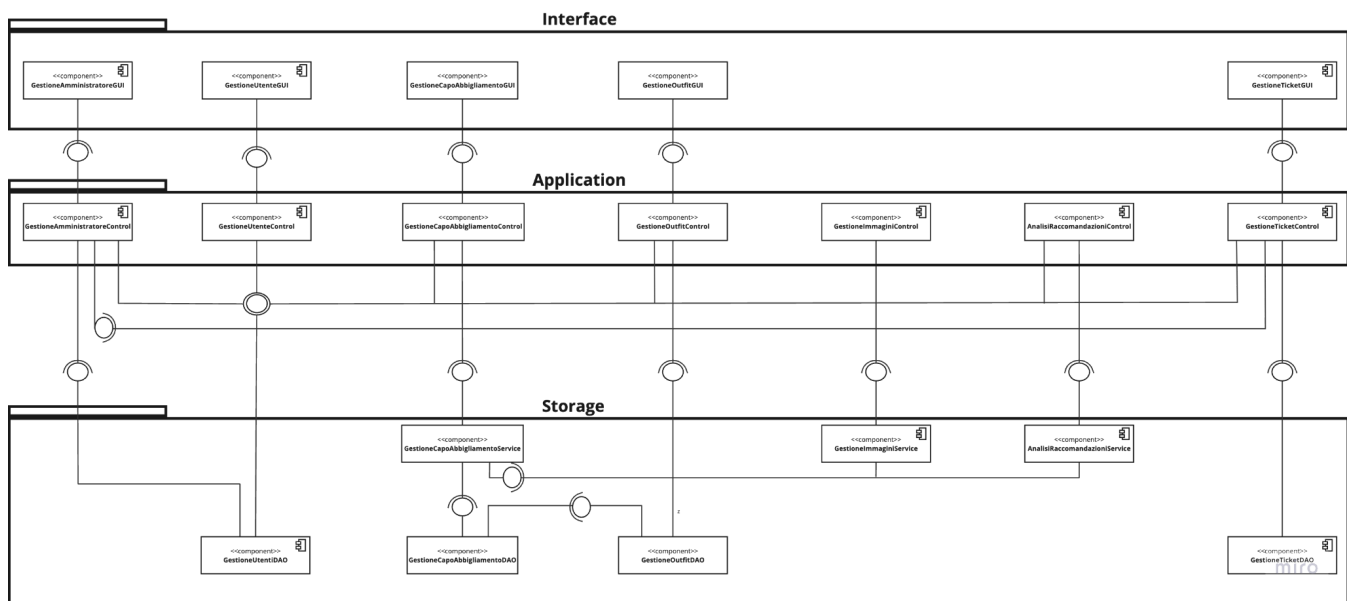


Di seguito una vista dettagliata di ciascun sottosistema evidenziando le componenti

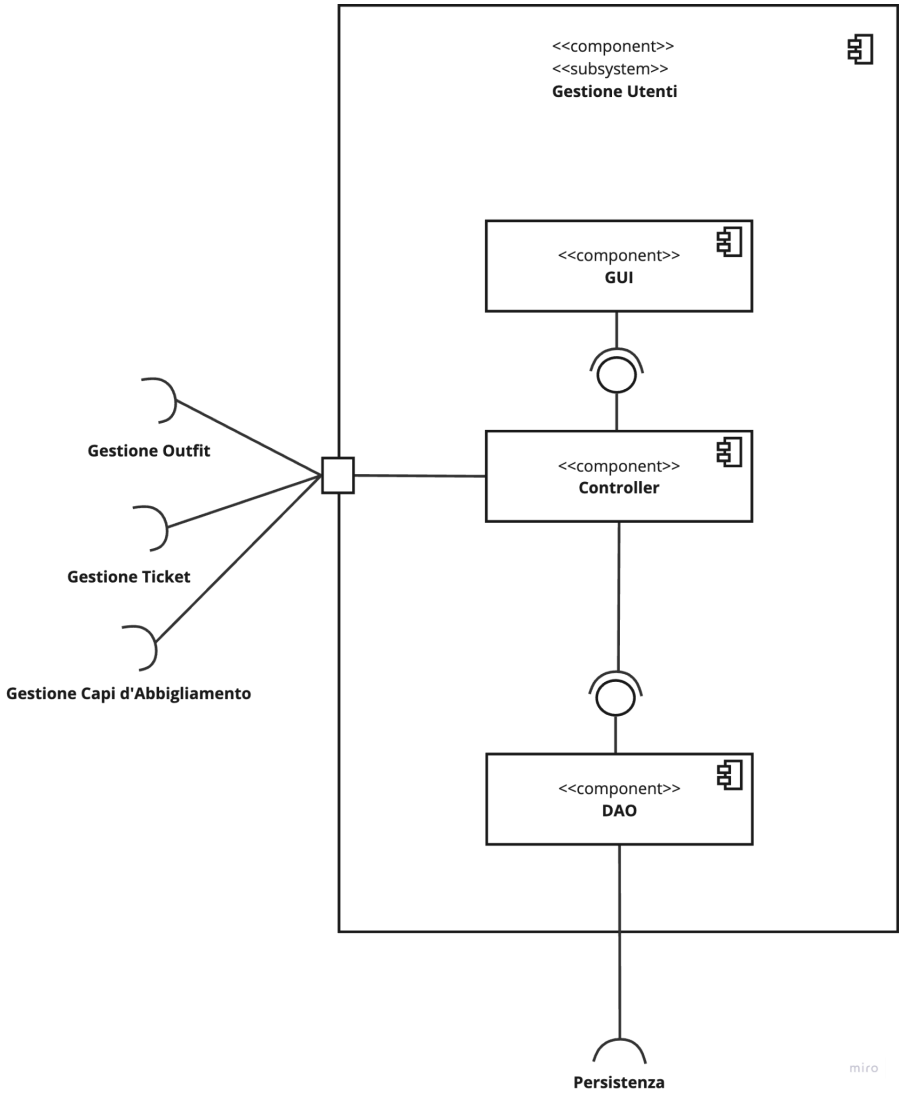
principale:

- **GUI:** Graphic User Interface, che contiene le varie view che saranno renderizzate per creare le pagine web da mostrare al cliente.
- **Controller:** si occupa della logica per il controllo del sistema.
- **DAO:** Data Access Object, che si occupa della logica di business e di fornire accesso ai dati persistenti.

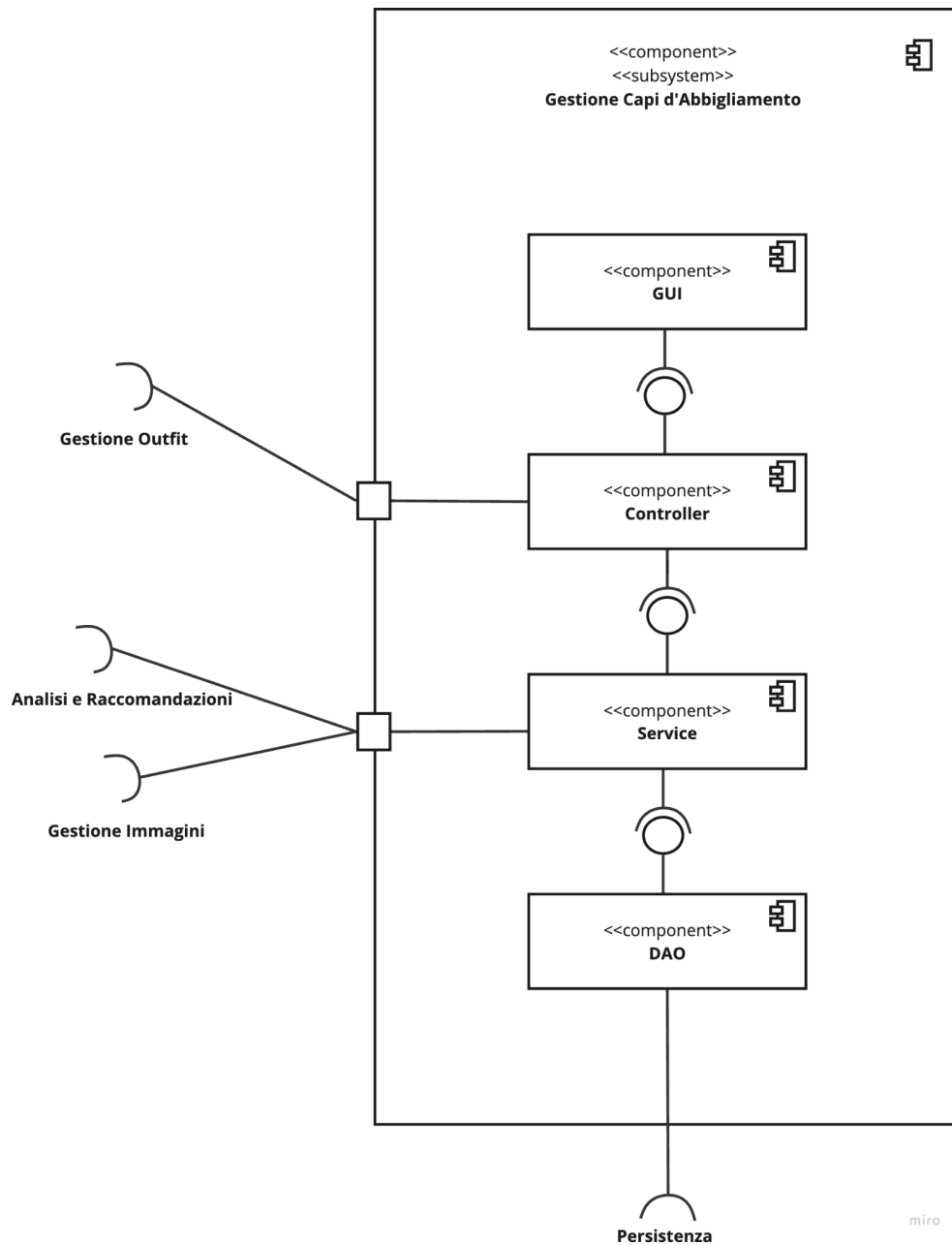
Diagramma Architeturale



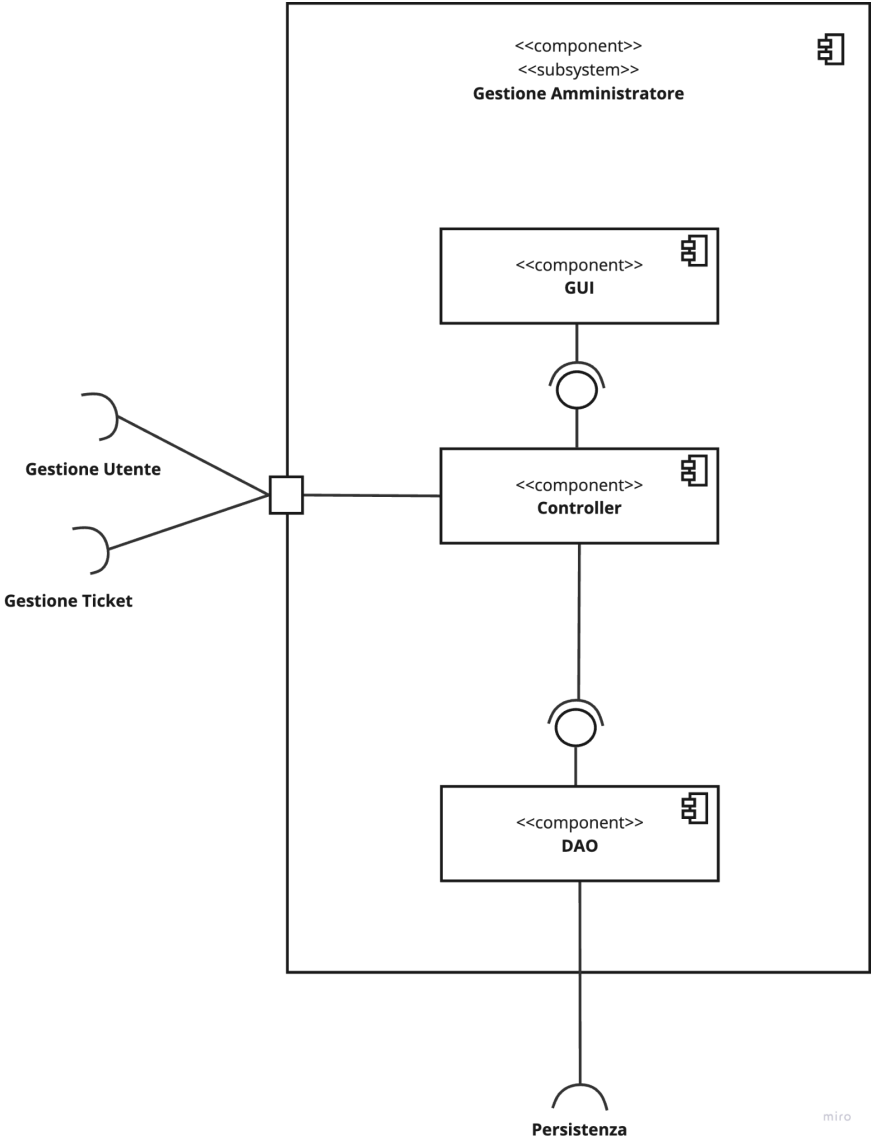
Sottosistema Gestione Utenti



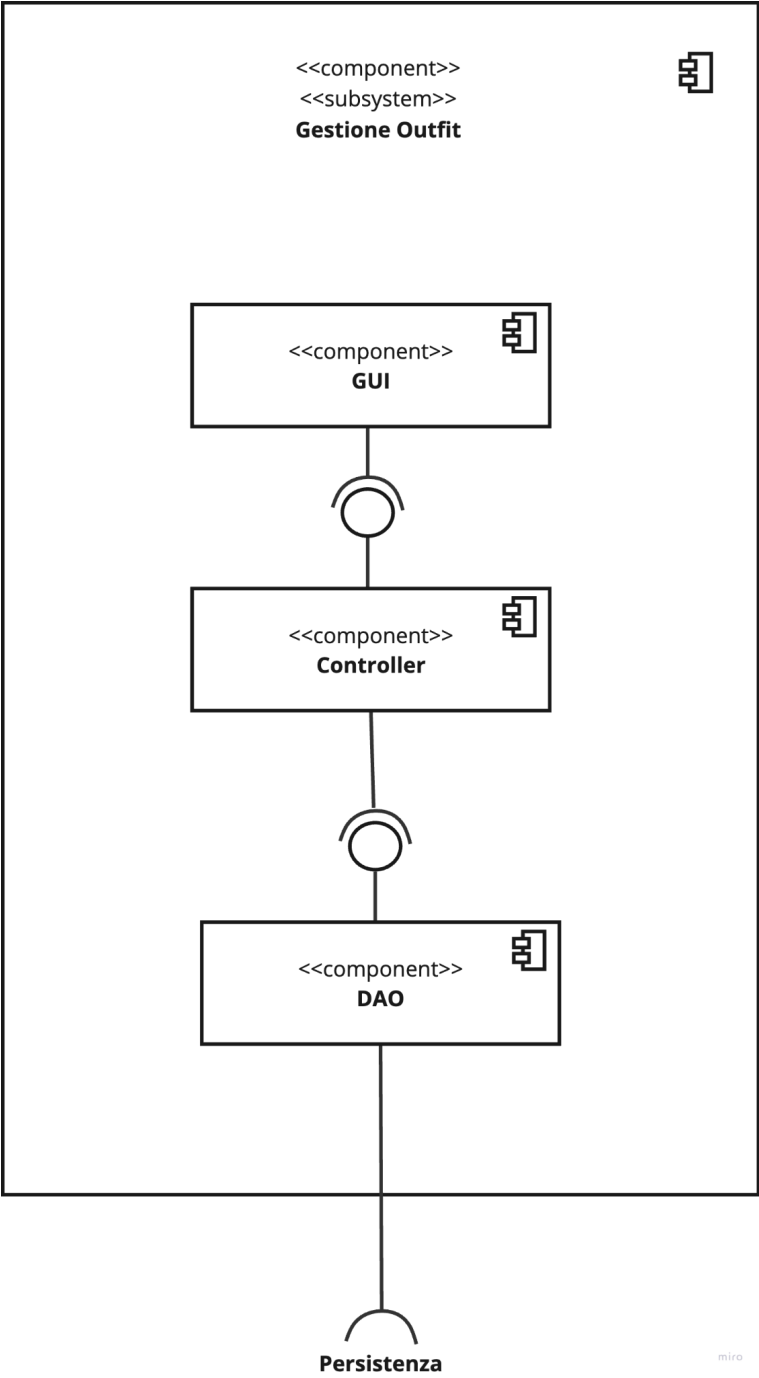
Sottosistema Gestione Capi d'Abbigliamento



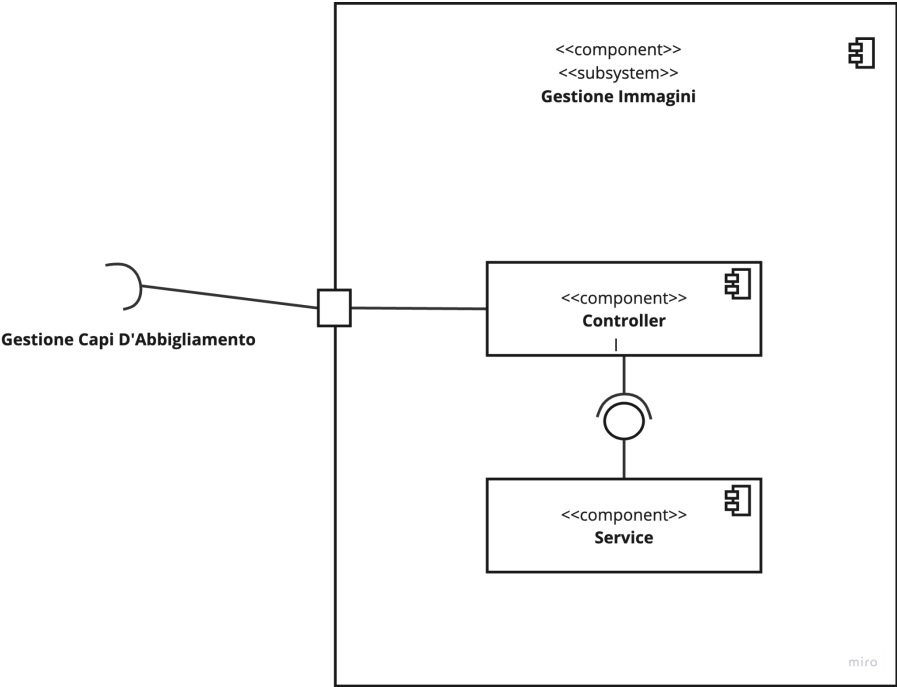
Sottosistema Gestione Amministratore



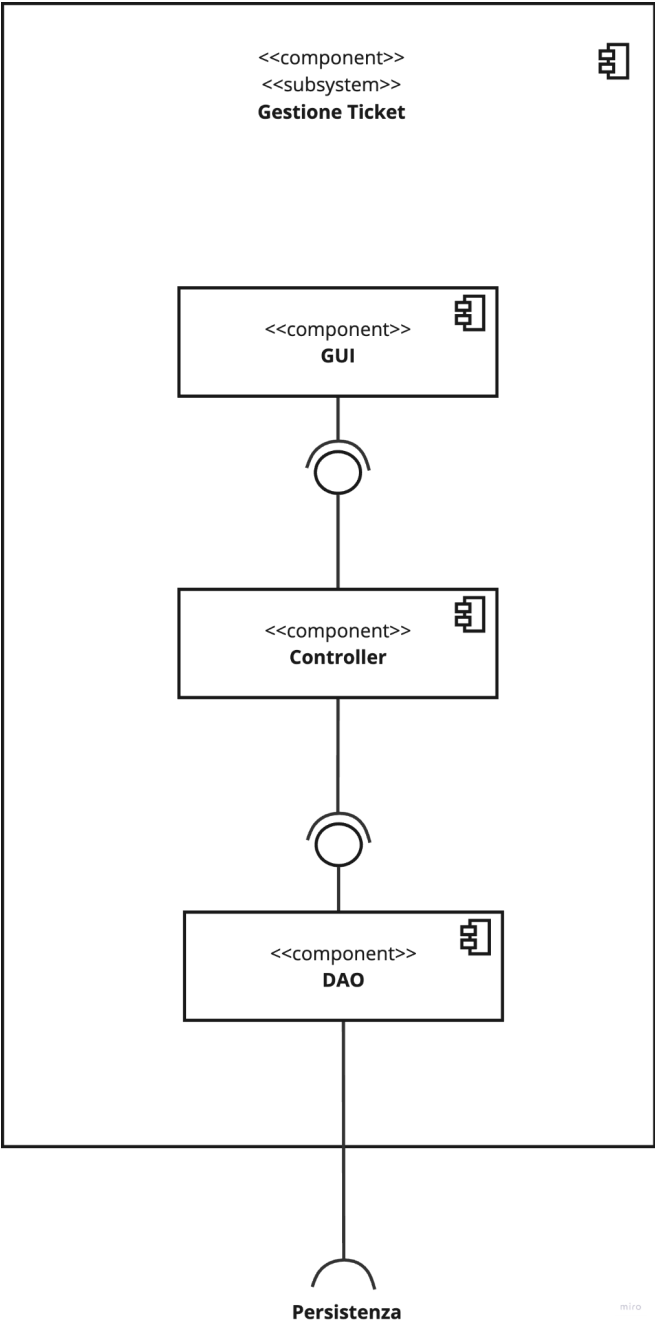
Sottosistema Gestione Outfit



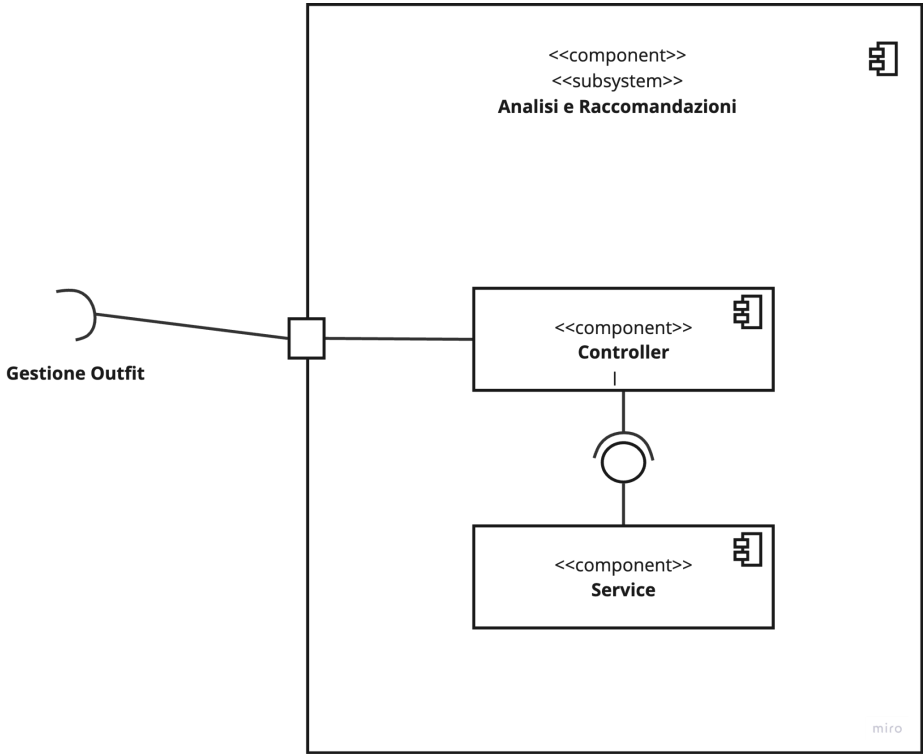
Sottosistema Gestione Immagini



Sottosistema Gestione Ticket



Sottosistema Analisi e Raccomandazioni

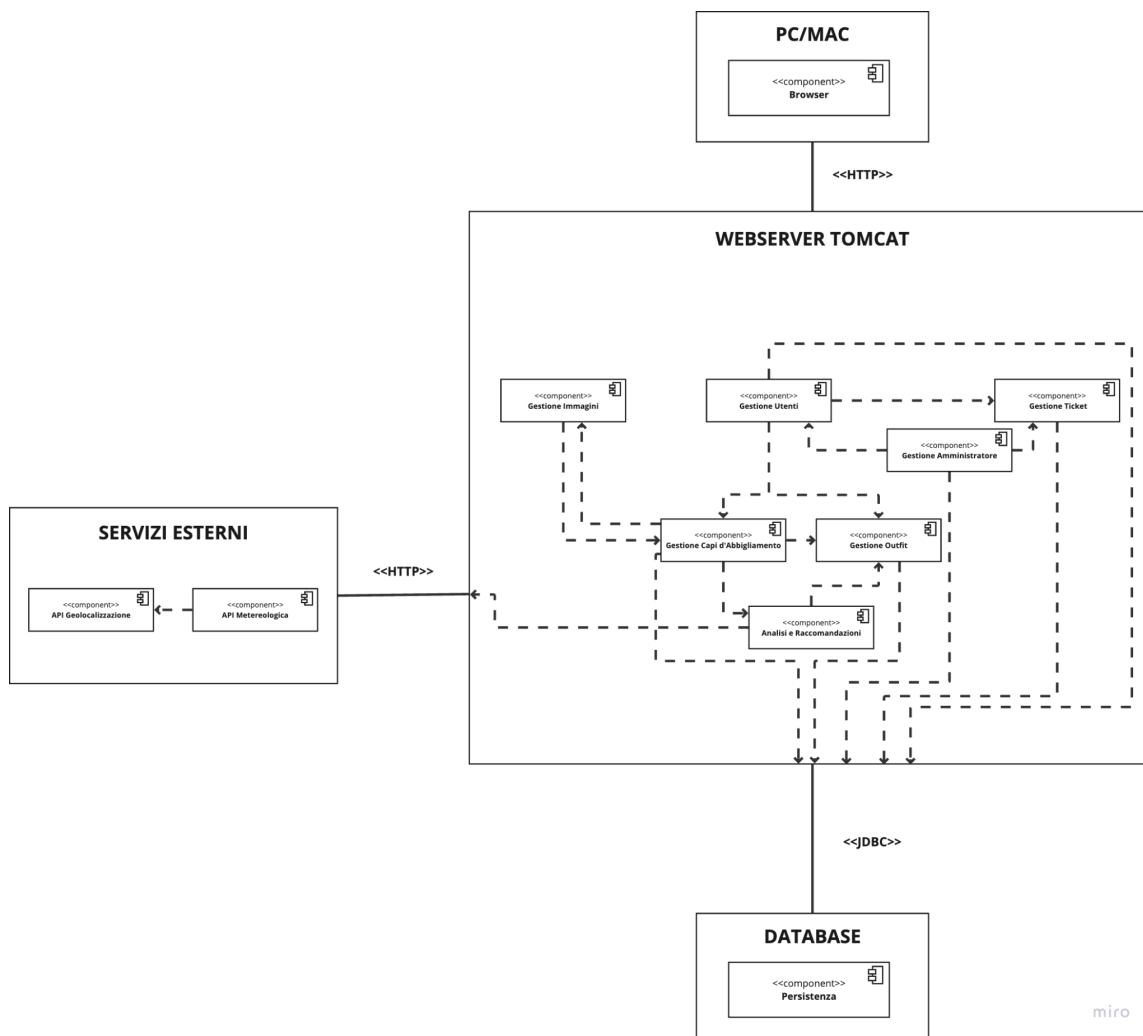


3.3 Mapping hardware/software

L' applicazione web che verrà sviluppata si basa su una piattaforma hardware costituita da un server che risponde alle richieste effettuate dai client da una qualsiasi macchina con un browser ed una connessione ad Internet.

Essendo che il nostro sistema è una web application e risiede su un web server, e che si basa su un'architettura non distribuita, risiede su un solo nodo.

Di seguito un UML deployment diagram che descrive il mapping hardware/software.



3.4 Gestione dei dati persistenti

Introduzione

Per la gestione del salvataggio dei dati persistenti del sistema si è deciso di utilizzare un database relazionale, al fine di gestire agevolmente l'accesso concorrente ai dati e contemporaneamente garantire la consistenza dei dati tramite l'utilizzo di un DBMS. La scelta di utilizzo di un DBMS è stata presa al fine di mantenerci quanto più possibile coerenti con i design goals stabiliti, potendo contare su:

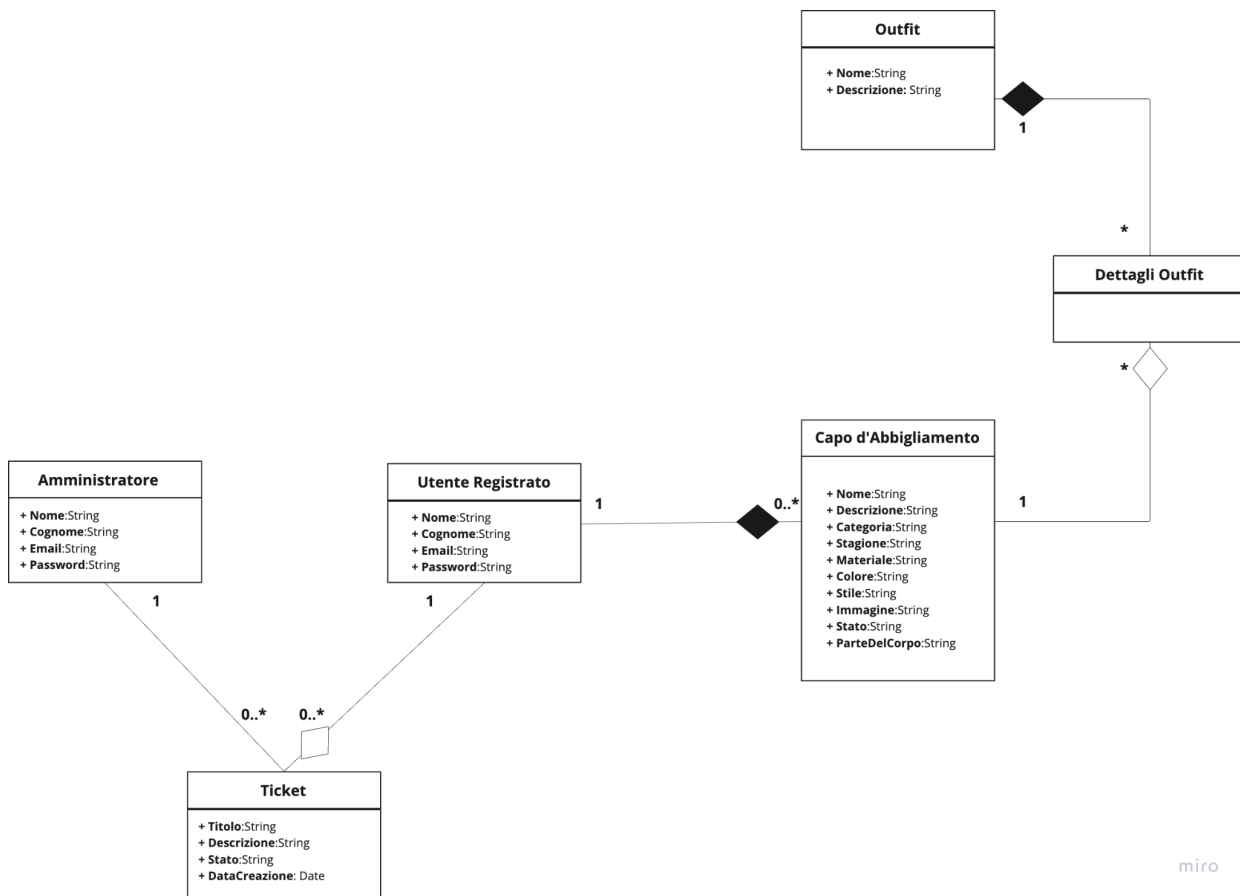
- **Imposizioni di vincoli di integrità sui dati**, poiché un DBMS permette di specificare diversi tipi di vincoli per mantenere l'integrità dei dati e controlla che tali vincoli siano soddisfatti quando la base di dati cambia.
- **Privatezza dei dati**, garantita dal fatto che un DBMS permette un accesso protetto ai dati. Utenti diversi possono avere accesso a diverse porzioni della base di dati e possono essere abilitati a diverse operazioni su di esse.
- **Affidabilità dei dati**, infatti un DBMS offre dei metodi per salvare copie dei dati e per ripristinare lo stato della base di dati in caso di guasti software e hardware.
- **Atomicità delle operazioni**, data dal fatto che un DBMS permette di effettuare sequenze di operazioni in modo atomico. Ciò significa che l'intera sequenza di operazioni viene eseguita con successo oppure nessuna di queste operazioni ha alcun effetto sui dati della base. L'atomicità delle transazioni permette di mantenere uno stato della base di dati consistente con la realtà modellata.

È stato scelto di utilizzare un database in locale, avvalendoci dell'utilizzo di un server MySQL. Il linguaggio di programmazione utilizzato per lo sviluppo del database è SQL.

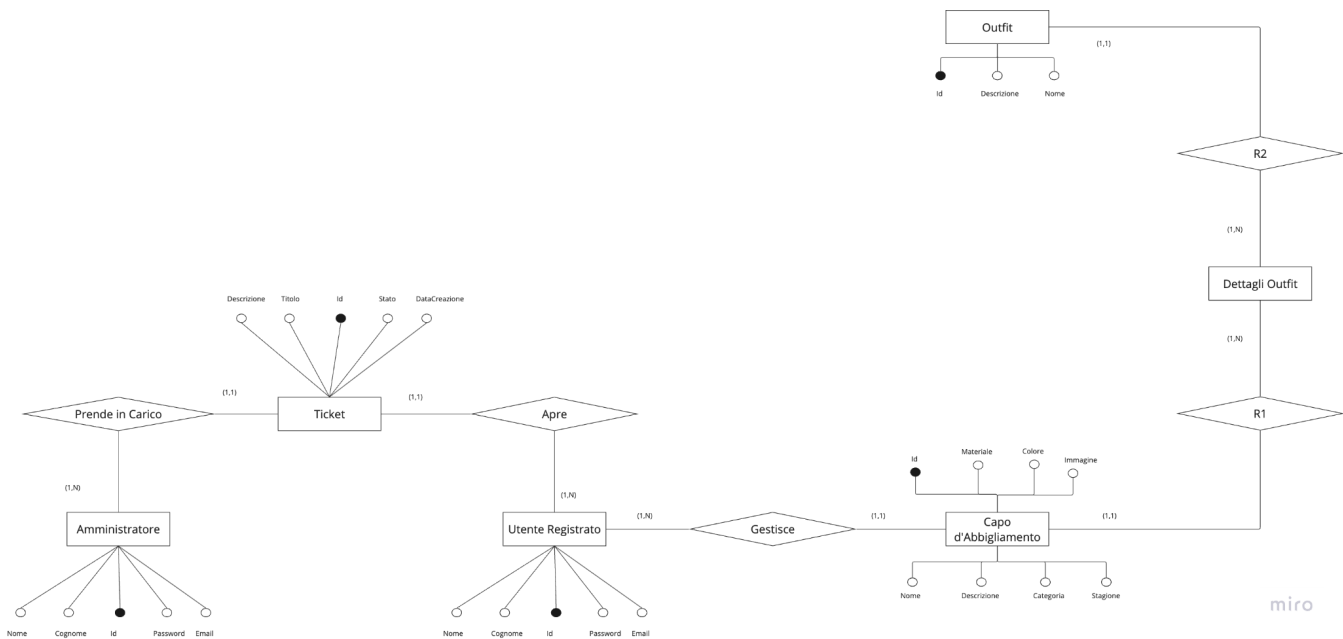
CD_SDD: Entity Class Diagram ristrutturato

La ristrutturazione del nostro Class Diagram presenta una modifica:

- L'introduzione della tabella "Dettagli Outfit" per una gestione più scalabile della relazione tra outfit e capi di abbigliamento, migliorando la flessibilità del database per future estensioni, come l'aggiunta di attributi specifici alla relazione (ad esempio, quantità di capi usati in un outfit, priorità, varianti, ecc.).



ER: Schema ER del database



Dizionario dei dati

Di seguito si mostrano gli attributi per ogni entità individuata.

Nome Entità	Capo d'Abbigliamento		
Descrizione	Contiene i dati relativi ad un capo d'abbigliamento inserito da un utente		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	int	PRIMARY KEY	NOT NULL
id_utente	int	FOREIGN KEY	NOT NULL
Descrizione	varchar[100]		
Nome	varchar[30]		NOT NULL
Categoria	varchar[30]		NOT NULL
Stagione	varchar[15]		NOT NULL
Colore	varchar[20]		NOT NULL
Immagine	varchar[150]		NOT NULL
Materiale	varchar[40]		NOT NULL
Stato	varchar[40]		NOT NULL
Stile	varchar[40]		NOT NULL
Parte del corpo	varchar[40]		NOT NULL

Nome Entità	Amministratore		
Descrizione	Contiene i dati relativi ad un amministratore di Wearly		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	int	PRIMARY KEY	NOT NULL
Email	varchar[319]		UNIQUE
Password	varchar[20]		NOT NULL
Nome	varchar[30]		NOT NULL
Cognome	varchar[30]		NOT NULL

Nome Entità	Utente Registrato		
Descrizione	Contiene i dati relativi ad un utente registrato a Wearly		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	int	PRIMARY KEY	NOT NULL
Email	varchar[319]		UNIQUE
Password	varchar[20]		NOT NULL
Nome	varchar[30]		NOT NULL
Cognome	varchar[30]		NOT NULL

Nome Entità	Dettagli Outfit		
Descrizione	Contiene la corrispondenza relativa ad un outfit e un capo d'abbigliamento		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id_outfit	int	PRIMARY KEY, FOREIGN KEY	NOT NULL
id_capo	int	PRIMARY KEY, FOREIGN KEY	NOT NULL

Nome Entità	Outfit		
Descrizione	Contiene i dati relativi ad un outfit		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
id	int	PRIMARY KEY	NOT NULL
Descrizione	varchar[100]		
Nome	varchar[20]		NOT NULL

Nome Entità	Ticket		
Descrizione	Contiene i dati relativi ad un utente registrato a Wearly		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli

Nome Entità	Ticket		
id	int	PRIMARY KEY	NOT NULL
Titolo	varchar[319]		NOT NULL
Descrizione	varchar[319]		NOT NULL
Stato	varchar[30]		NOT NULL
DataCreazione	TimeStamp		DEFAULT CURRENT_TIMESTAMP
idUtenteRegistrato	int	FOREIGN KEY	
idAmministratore	int	FOREIGN KEY	

3.5 Controllo degli accessi e sicurezza

Di seguito viene mostrata la matrice degli accessi per poter tenere traccia di quali attori possono accedere a quali dei servizi offerti dal sistema.

OGGETTO	Utente Registrato	Amministratore
Gestione Utenti	<ul style="list-style-type: none"> •Login •Logout •VisualizzaAreaUtente •ModificaDatiPersonali 	<ul style="list-style-type: none"> •Login •Logout •VisualizzaAreaAdmin
Gestione Amministratore		<ul style="list-style-type: none"> •ModificaUtente •CancellaUtente •CreazioneUtente

OGGETTO	Utente Registrato	Amministratore
Gestione Capi d'Abbigliamento	<ul style="list-style-type: none"> •AggiungiCapo •EliminaCapo •ModificaCapo •VisualizzaGuardaroba 	
Gestione Outfit	<ul style="list-style-type: none"> •CreaOutfitManuale •CreaOutfitAI •ModificaOutfit •EliminaOutfit •VisualizzaAreaOutfit 	
Gestione Immagini		
Gestione Ticket	<ul style="list-style-type: none"> •VisualizzaAreaAssistenza •InviaTicket 	<ul style="list-style-type: none"> •VisualizzaAreaTicket •PresaleInCaricoDelTicket •CancellaTicket
Analisi e Raccomandazioni		

3.6 Controllo globale del software

Il sistema Wearly è un sistema interattivo per cui ogni funzionalità viene avviata in seguito ad un comando impartito dall'utente tramite l'uso di un'interfaccia grafica.

Quindi quando un utente vuole accedere ed utilizzare una funzionalità del sistema può farlo tramite l'interfaccia grafica la quale selezionerà il controllo corrispondente.

L'azione scatenerà un evento il quale verrà gestito dal suo handler. L'handler a sua volta indirizzerà il controllo del flusso di eventi al sottosistema che si occupa della logica e gestione di controllo, che poi si rivolge ai servizi per la logica applicativa.

Per tali motivi il sistema utilizzerà un meccanismo di controllo del flusso di tipo event-driven essendo una web-application.

3.7 Condizioni limite

Nel presente paragrafo verranno presentate le boundary conditions inerenti all'avvio del sistema, spegnimento del sistema, fallimento del sistema ed errore di accesso ai dati persistenti.

Avvio del Sistema

Identificativo	UCBC_1 – Avvio del Sistema	Data	14/02/2025
		Versione	2.0
		Autori	Fontana Eustachio
Descrizione	Lo UC permette l'avvio del sistema.		
Attore principale	Amministratore.		
Attori secondari	NA		
Entry condition	L'Amministratore accede al Server.		
Exit condition On success	Il sistema viene avviato correttamente.		
Exit condition On failure	Il sistema non viene avviato.		
Flusso di eventi principale			
1	Amministratore	Esegue sulla macchina il comando che avvia il sistema.	
2	Sistema	Verifica la sanità dei dati persistenti e, se sani, rende disponibili i suoi servizi e rende le sue funzionalità disponibili agli utenti.	
I Flusso di Eventi Alternativo: I Dati Persistenti sono danneggiati			
2.a1	Sistema	Notifica l'Amministratore di problemi ai dati persistenti e non effettua l'avvio.	
2.a2	Amministratore	Corregge i dati persistenti	
2.a3	Amministratore	Esegue il Passaggio 1	

Spegnimento del sistema

Identificativo	UCBC_2 – Spegnimento del Sistema	Data	14/02/2025
		Versione	2.0
		Autori	Somma Pietro
Descrizione	Lo UC permette lo spegnimento del sistema		
Attore principale	Amministratore		
Attori secondari	NA		

Entry condition		L'Amministratore accede al Server AND Il Sistema è stato precedentemente avviato AND Il Sistema non è stato ancora spento
Exit condition On success		Il sistema viene spento correttamente
Exit condition On failure		Il sistema non viene spento
Flusso di eventi principale		
1	Amministratore	Invia un segnale di spegnimento al Sistema
2	Sistema	Controlla che non ci siano connessioni ancora aperte da o verso l'esterno e, se non ci sono, termina l'esecuzione del sistema.
I Flusso di eventi alternativo: Ci sono connessioni ancora aperte		
2.a1	Sistema	Notifica all'Amministratore che ci sono ancora connessioni aperte verso l'esterno.
2.a2	Sistema	Attende una quantità di tempo per rispondere a eventuali richieste dall'esterno, non generando nuove connessioni se non allo scopo di rispondere a richieste già in corso.
2.a3	Sistema	Controlla che non ci siano connessioni ancora aperte da o verso l'esterno e, se non ci sono, termina l'esecuzione del sistema.
2.a4	Sistema	Notifica l'Amministratore dell'avvenuto spegnimento del sistema.
II Flusso di eventi alternativo: Ci sono connessioni ancora aperte		
2.a3.a1	Sistema	Recide le connessioni verso l'esterno.
2.a3.a2	Sistema	Notifica l'Amministratore dell'avvenuto spegnimento del sistema e del numero di connessioni recise.

Fallimento del sistema

Identificativo	UCBC_3 – Fallimento del Sistema	Data	14/02/2025
		Versione	2.0
		Autori	Ioanna Carmine
Descrizione	L'UC definisce il comportamento del Sistema in caso di fallimento.		
Attore principale	Amministratore		
Attori secondari	NA		
Entry condition	Il Sistema viene terminato inaspettatamente		

Exit condition On success		Il Sistema viene riavviato correttamente
Exit condition On failure		Il Sistema non viene riavviato
Flusso di eventi principale		
1	Amministratore	Include UCBC_1

Errore di Accesso ai Dati Persistenti

Identificativo	UCBC_4 – Errore di Accesso ai Dati Persistenti	Data	14/02/2025
		Versione	2.0
		Autori	Dell’Atti Alessandro
Descrizione	L’UC descrive il comportamento del sistema qualora fosse impossibile accedere ai dati persistenti o questi risultassero corrotti.		
Attore principale	Amministratore		
Attori secondari	NA		
Entry condition	Il Sistema non può accedere ai dati persistenti OR I dati persistenti risultano corrotti		
Exit condition On success	Il Sistema riprende il normale funzionamento		
Exit condition On failure	Il Sistema non riprende il normale funzionamento		
Flusso di eventi principale			
1	Sistema	Notifica l'amministratore dell'impossibilità di accedere ai dati persistenti	
2	Sistema	Cessa di processare eventuali richieste dall'esterno e risponde a tutte le richieste con un messaggio di errore.	
3	Amministratore	Include UCBC_2	
4	Amministratore	Ripristina l'accessibilità o la sanità dei dati persistenti.	
5	Amministratore	Include UCBC_1	

4 Servizi dei Sottosistemi

In questa sezione vengono descritti i servizi di ogni sottosistema precedentemente elencati.

Sottosistema Gestione Utenti

SERVIZIO	DESCRIZIONE	INTERFACCIA
Login	Il servizio di login permette di accedere al sistema in modo sicuro, autenticando l'utente tramite le credenziali fornite e avviando una nuova sessione attiva.	LoginServlet
Logout	Il servizio di logout permette di disconnettersi dal sistema in modo sicuro, terminando la sessione attiva.	LogoutServlet
VisualizzaAreaUtente	Questo servizio consente agli utenti autenticati di visualizzare la propria area riservata	AuthenticationServlet
ModificaDatiPersonali	Questo servizio permette di aggiornare in modo sicuro i dati personali dell'utente, garantendo l'integrità e la protezione delle informazioni memorizzate nel sistema.	UserPageUpdateServlet
VisualizzaAreaAdmin	Questo servizio permette di accedere all'area di gestione utenti, consentendo la gestione e l'amministrazione degli utenti registrati.	AuthenticationServlet

Sottosistema Gestione Amministratore

SERVIZIO	DESCRIZIONE	INTERFACCIA
ModificaUtente	Questo servizio consente di cambiare le informazioni associate a un account di un utente registrato.	UserPageUpdateServlet
CancellaUtente	Questo servizio consente di rimuovere un account utente dal sistema, eliminando tutte le informazioni relative all'utente.	AdminPageCRUDServlet
CreazioneUtente	Questo servizio consente di aggiungere un account utente al sistema.	AdminPageCRUDServlet

Sottosistema Gestione Capi D'Abbigliamento

SERVIZIO	DESCRIZIONE	INTERFACCIA
AggiungiCapo	Questo servizio consente di aggiungere un nuovo capo di abbigliamento nel sistema.	AddCapoServlet
EliminaCapo	Questo servizio consente di rimuovere un capo di abbigliamento dal sistema.	DeleteCapoServlet
ModificaCapo	Questo servizio consente di aggiornare i dettagli di un capo di abbigliamento già presente nel sistema.	EditCapoServlet

SERVIZIO	DESCRIZIONE	INTERFACCIA
VisualizzaGuardaroba	Questo servizio permette di visualizzare il guardaroba virtuale, dove sono archiviati i capi di abbigliamento registrati nel sistema.	ToWardrobeServlet

Sottosistema Gestione Outfit

SERVIZIO	DESCRIZIONE	INTERFACCIA
CreaOutfitManuale	Questo servizio consente di creare un outfit manualmente selezionando i capi di abbigliamento dal guardaroba disponibile.	OutfitCRUDServlet
CreaOutfitAI	Questo servizio permette di creare un outfit automaticamente tramite l'intelligenza artificiale, che suggerisce una combinazione di capi in base al clima e alle caratteristiche dei capi.	OutfitAIServlet
ModificaOutfit	Questo servizio consente di aggiornare un outfit già creato.	OutfitCRUDServlet
EliminaOutfit	Questo servizio consente di rimuovere un outfit.	OutfitCRUDServlet

SERVIZIO	DESCRIZIONE	INTERFACCIA
VisualizzaAreaOutfit	Questo servizio permette di visualizzare l'area outfit, dove sono mostrati gli outfit disponibili e le relative informazioni salvate nel sistema.	toOutfitServlet

Sottosistema Gestione Ticket

SERVIZIO	DESCRIZIONE	INTERFACCIA
VisualizzaAreaAssistenza	Questo servizio consente agli utenti di accedere all'area di supporto dove possono visualizzare i ticket aperti e inviarne di nuovi.	ToSupportPage
inviaTicket	Questo servizio permette agli utenti di creare e inviare un nuovo ticket di supporto fornendo dettagli come nome del problema e descrizione.	TicketServlet
VisualizzaAreaTicket	Questo servizio consente agli amministratori di accedere a una vista dedicata per monitorare e gestire tutti i ticket inviati dagli utenti.	AuthenticationServlet
PresalInCaricoDelTicket	Questo servizio consente agli amministratori di prendere in carico un ticket specifico aggiornando lo stato del ticket in "in corso".	AdminTicketServlet

SERVIZIO	DESCRIZIONE	INTERFACCIA
CancellaTicket	Questo servizio permette agli amministratori di eliminare un ticket specifico dal sistema, rimuovendolo definitivamente dalla base dati.	AdminTicketServlet

5 Design Pattern

5.1 Façade Pattern

Si è scelto di utilizzare il Façade Pattern che deve assicurarsi di fornire un'interfaccia unificata semplificata, per l'accesso a sottosistemi che espongono interfacce complesse e diverse fra loro.

L'utilizzo del Façade Pattern permette di nascondere la complessità delle operazioni, fornendo un'interfaccia di alto livello per rendere più facile l'utilizzo del sottosistema.

Nel nostro caso, il Façade Pattern è realizzato attraverso le interfacce manager.

6 Glossario

Nella presente sezione sono raccolte le sigle o i termini del documento che necessitano di una definizione.

SIGLA / TERMINE	DEFINIZIONE
Wearly	Nome dell'applicativo che si andrà a realizzare.
Three Tier	Architettura software a tre livelli che separa l'applicazione in tre componenti distinte: il front-end (interfaccia utente), il back-end (logica di business), e il database (memorizzazione dei dati).

Front-End	La parte dell'applicazione con cui l'utente interagisce direttamente, ossia l'interfaccia grafica (UI) e le funzionalità visibili e accessibili dall'utente.
Back-End	La parte dell'applicazione che gestisce la logica di business, l'elaborazione dei dati e la comunicazione con il database. Non è visibile all'utente finale, ma supporta il funzionamento del front-end.
View	Componente che si occupa di visualizzare i dati all'utente, rappresentando l'interfaccia grafica dell'applicazione.
Database	Sistema di gestione dei dati che archivia, gestisce e fornisce accesso ai dati in modo strutturato, attraverso tabelle, query e operazioni di lettura/scrittura.