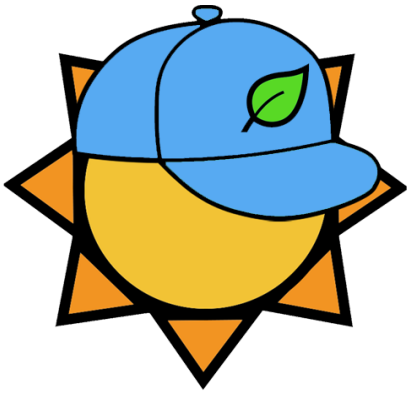
SDD

Progetto

WeatherStyle



Versione	0.4
Data	16/12/2022
Destinatario	Professore Carmine Gravino.
Presentato da	Aurucci Raffaele, Miglino Annalaura, Palmieri Angelo, Zammarrelli Francesco Giuseppe.

Revision History

Data	Versione	Descrizione	Autore
24/11/2022	0.1	Effettuata Scomposizione in sottosistemi e scelta dell'architettura	Tutto il team
29/11/2022	0.1	Effettuata mappatura della scomposizione nell'architettura three tier	Tutto il team
30/11/2022	0.1	Effettuato mapping hardware e software	Tutto il team
06/12/2022	0.1	Revisione Design Goal	Tutto il team
07/12/2022	0.2	Aggiunti Gestione dei Dati Persistenti e Controllo degli accessi e sicurezza	Tutto il team
07/12/2022	0.3	Aggiunti Condizioni limite	Tutto il team
17/12/2022	0.4	Revisione completa	Tutto il team

Sommario

1. Introduction

1.1 Purpose Of System

WeatherStyle ha come obiettivo principale quello di mostrare le previsioni meteo di qualsiasi città del mondo e, sulla base di queste, suggerire agli utenti dei capi d'abbigliamento ritenuti più adatti da indossare.

Il sistema permette ad un Utente di aggiungere dei capi d'abbigliamento al suo guardaroba digitale, e fornire quindi dei suggerimenti basandosi su questi ultimi.

Inoltre, offre la possibilità agli Utenti di inviare una richiesta di promozione ad Ecologista, la quale sarà valutata da un Admin, che potrà quindi approvare o rifiutare tale richiesta.

Gli Ecologisti avranno la possibilità di pianificare eventi a favore dell'ambiente in una determinata città e, sempre per lo stesso scopo, potranno pubblicare dei post.

Il cuore del sistema è determinato dalla componente di Intelligenza Artificiale che necessita delle informazioni meteo e delle informazioni sui capi d'abbigliamento posseduti dall'Utente, e dalla componente Ambientale che offre le funzionalità sopra citate ad un Ecologista.

1.2 Design Goal

Rank	ID	Descrizione	Categoria	RNF di origine
5	DG_1 Usabilità	Il sistema deve essere conforme ai criteri di successo stabiliti nella WCAG 2.0	End User Criteria	RNF_1 RNF_2
7	DG_2 Throughput	Il sistema deve garantire la fruizione dei suoi contenuti concorrentemente ad un massimo di 1000 utenti.	Performance	RNF_3
3	DG_3 Fault Tolerance	Il sistema deve notificare l'utente mediante appositi messaggi in caso di servizi non fruibili o in caso di errori durante un'operazione, supportandolo verso il suo completamento.	Dependability	RNF_4 RNF_5
6	DG_4 Response Time	Il sistema deve garantire un tempo di risposta agli input degli utenti di massimo 20 secondi.	Performance	RNF_6
4	DG_5 Modifiability	Il sistema deve essere facilmente modificabile,	Maintenance	RNF_7

		garantendo la modularità negli aggiornamenti, poiché seguirà lo standard ISO/IEC 25011.		
2	DG_6 Development cost	L'utilizzo delle API da cui il sistema acquisisce informazioni sulle città e il meteo devono essere open source e gratuite.	Cost Criteria	RNF_8 RNF_9
8	DG_7 Utility	Il sistema dovrebbe fornire una console amministrativa che permetta di vedere informazioni su tutti gli utenti	End User Criteria	RNF_10
1	DG_8 Modifiability	Il sistema deve essere progettato in logica Object Oriented.	Maintenance	RNF_11
9	DG_9 Extensibility	Il sistema deve poter essere esteso dando la possibilità di utilizzarlo tramite una progressive Web App.	Maintenance	RNF_12

1.2.1 Trade-off

Trade-off	Razionale
Tempo di distribuzione vs Funzionalità	Per ridurre al minimo i tempi di sviluppo ci limiteremo ad implementare solo le funzionalità a priorità elevata come definito nelle specifiche iniziali ai fini di rientrare nei tempi prestabiliti per il rilascio e con meno bug.
Costi di sviluppo vs Velocità	Per ridurre i costi di sviluppo si è disposti ad utilizzare delle API per il meteo e le città gratuite anche se queste implicano un maggiore tempo di risposta.
Velocità vs Spazio	Nelle operazioni che non dipendono dalle API esterne, per rispettare il vincolo dei tempi di risposta si è disposti ad occupare più spazio di memoria per aumentare la velocità.

1.3 Definizioni Acronimi e abbreviazioni

Nel Documento vengono utilizzate i seguenti acronimi e abbreviazioni:

- DG : Design Goal
- DB : Database

1.4 References

- Statement Of Work
- RAD
- Matrice di Tracciabilità
- Test Plan
- Test Case specification

[I link saranno aggiunti nella versione finale]

1.5 Overview del documento

Il documento prevede le seguenti sezioni:

- **Introduzione:** Presentazione generale dello scopo del sistema e degli obiettivi di design che si intende rispettare.

- **Architettura del sistema corrente:** Viene descritto lo stato attuale dell'architettura dei sistemi già sul mercato.

- **Architettura proposta:** Presenta una overview generale di come il sistema sarà strutturato e scomposto in sottosistemi; a seguire vi è il mapping Hardware/Software rispetto ai sottosistemi precedentemente definiti, e una descrizione di come i dati persistenti saranno gestiti. Infine vi è la definizione e le motivazioni del flusso di controllo che caratterizza il sistema, oltre che gli use case legati alle boundary conditions.

2. Architettura del sistema corrente

Al momento della scrittura di questo documento non vi è alcun sistema software che integri, come WeatherStyle, la visualizzazione delle previsioni metereologiche con dei suggerimenti intelligenti sui capi d'abbigliamento più adeguati.
Fatta questa premessa viene naturale dire che non c'è la possibilità di comparare con criterio il sistema proposto ad un'altra architettura corrente poiché quest'ultima appunto non esiste.
Ovviamente ci sono molti sistemi che permettono di visualizzare le informazioni meteorologiche, che con ogni probabilità utilizzano un' architettura di tipo MVC o Three Tier.

3. Architettura Proposta

3.1 Overview

Il sistema si basa su un'architettura Three Tier. Facendo uso di questa architettura, vi sono dei sottosistemi che integrano le funzionalità dell'intero applicativo. Innanzitutto, per amministrare la registrazione e l'accesso degli utenti vi è il sottosistema Gestione Utenti,

La funzionalità principale del sistema è quella di suggerire dei capi d'abbigliamento sulla base delle condizioni meteo di una città, per questo sono stati inseriti i seguenti moduli: Gestione Meteo,

Gestione Città e Gestione Suggerimento IA; quest'ultimo si basa su una componente intelligente che, combinando i dati degli altri sottosistemi; è in grado di suggerire gli indumenti più adatti. Per poter usufruire di questa funzionalità, è necessario che l'utente informi il sistema dei capi d'abbigliamento che possiede: è stato quindi inserito il sottosistema Gestione Guardaroba.

Tra gli obiettivi del sistema vi è anche la volontà di sensibilizzare gli utenti sui cambiamenti climatici e sul rispetto dell'ambiente; è stato, quindi, inserito il sottosistema Gestione Ambiente che permette agli utenti promossi ad Ecologisti di pubblicare post ed eventi a favore dell'ambiente, in modo che qualsiasi utente possa rispettivamente leggere e partecipare per avere una maggiore sensibilità rispetto alle questioni ambientali.

3.2 Scomposizione in sottosistemi

Di seguiti tutti i sottosistemi individuati e una breve descrizione delle funzionalità offerte da ognuno:

- **Gestione Meteo:** fornisce la funzionalità di visualizzare le previsioni meteo interfacciandosi ad una API Meteo esterna.

- **Gestione Città:** fornisce la funzionalità di ricercare le città del mondo interfacciandosi ad una API Città esterna, oltre che salvare e rimuovere le città tra quelle preferite da un Utente.

- **Gestione Guardaroba:** fornisce la funzionalità di visualizzare il guardaroba di un Utente, oltre che inserire e rimuovere un nuovo capo d'abbigliamento da esso.

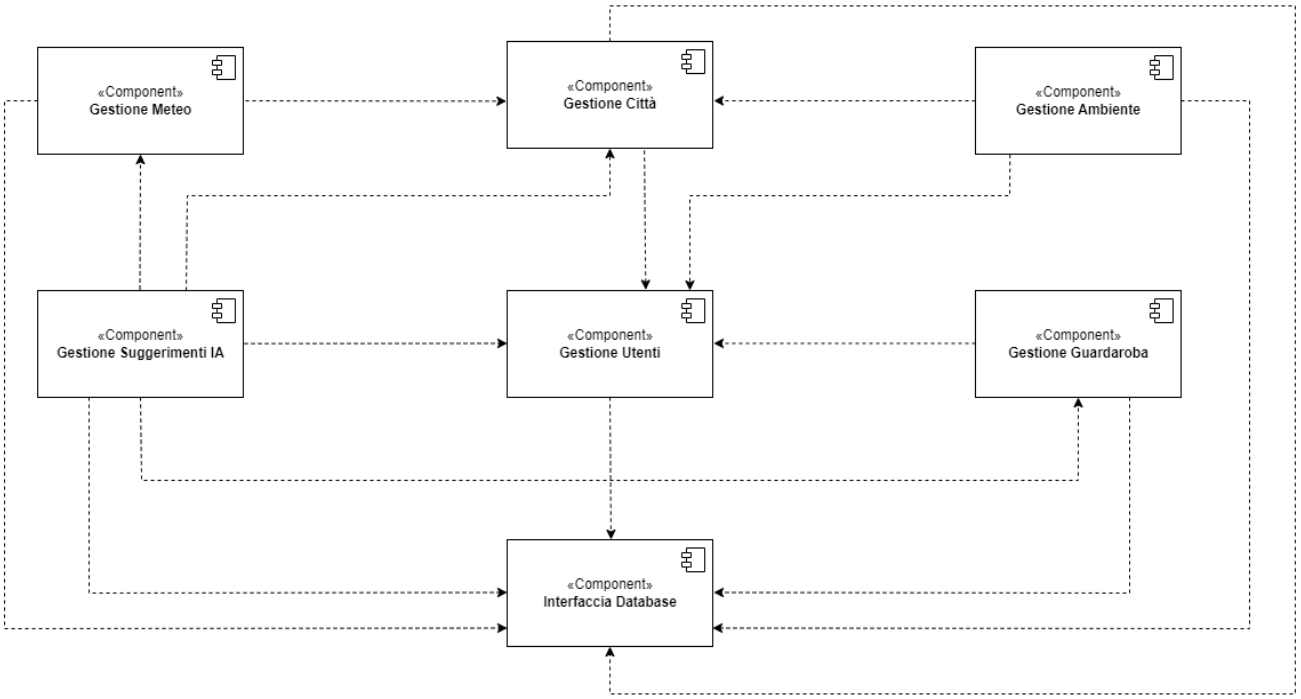
- **Gestione Suggerimento IA:** fornisce la funzionalità di suggerire capi d'abbigliamento su richiesta di un Utente in base alle previsioni meteo di una data città, dando la possibilità di fornire un feedback al suggerimento, e infine di visualizzare la cronologia dei suggerimenti accettati da un Utente.

- **Gestione Ambiente:** fornisce la funzionalità di pubblicare post e pianificare eventi a favore dell'ambiente da parte di un Ecologista.
  Inoltre fornisce le funzionalità di inviare una richiesta di promozione ad Ecologista da parte di un Utente, e di farla valutare da un Admin ai fini dell'approvazione.
  Infine, fornisce consigli giornalieri agli Utenti su tematiche per la riduzione dell'impatto ambientale.

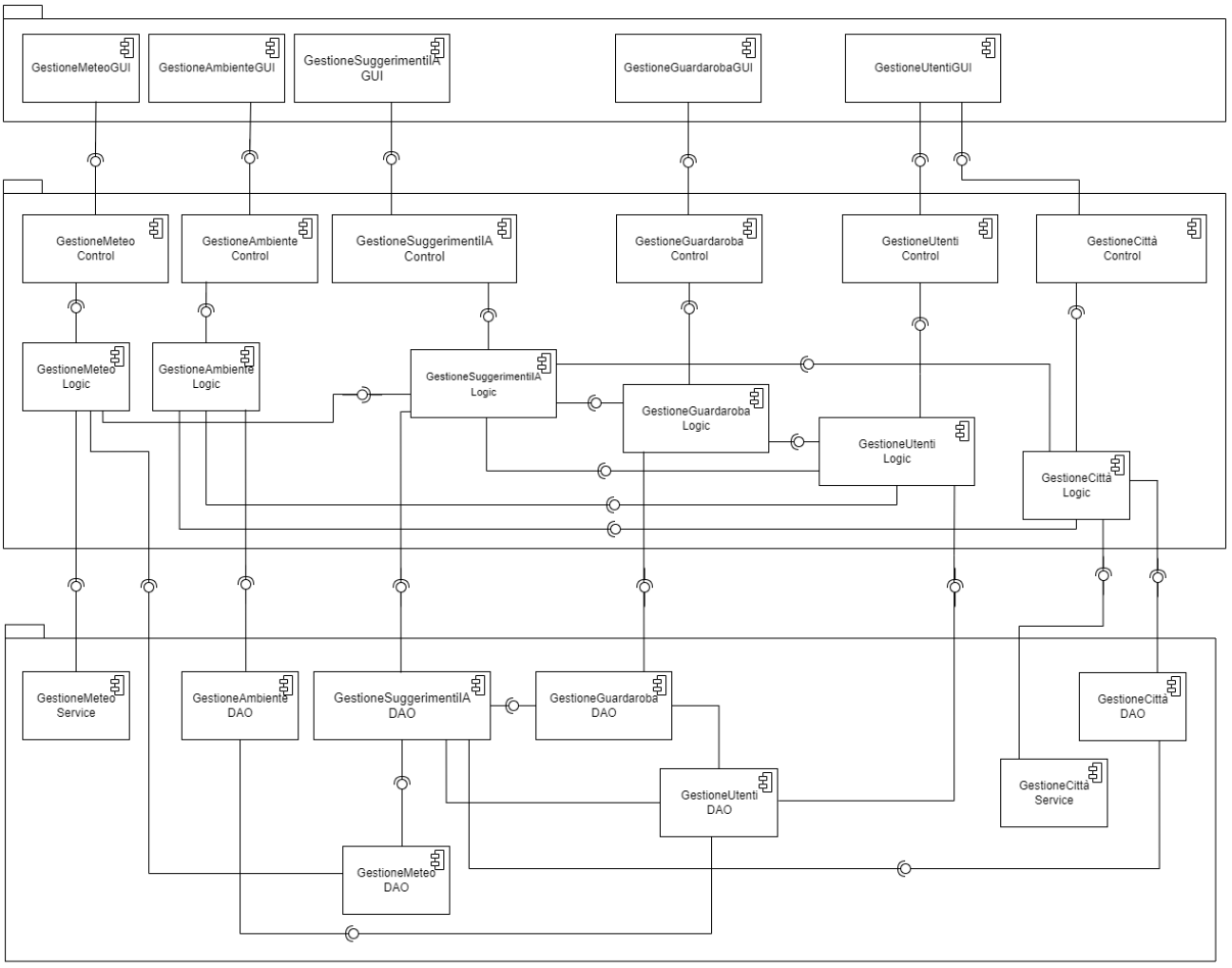
- **Gestione Utenti:** fornisce le funzionalità di registrazione di un Utente, e di modifica dei dati di quest'ultimo. Il sottosistema si occuperà di fornire, ai sottosistemi che lo necessitano, il servizio di utilizzo di un Utente, Ecologista o Admin, per portare a termine le proprie funzionalità.

- **Interfaccia Database:** fornisce un'interfaccia verso la componente Database, offrendo servizi di persistenza e di recupero dati alle componenti che lo necessitano.

A seguire l'UML Component Diagram:



Architettura del sistema
Three Tier

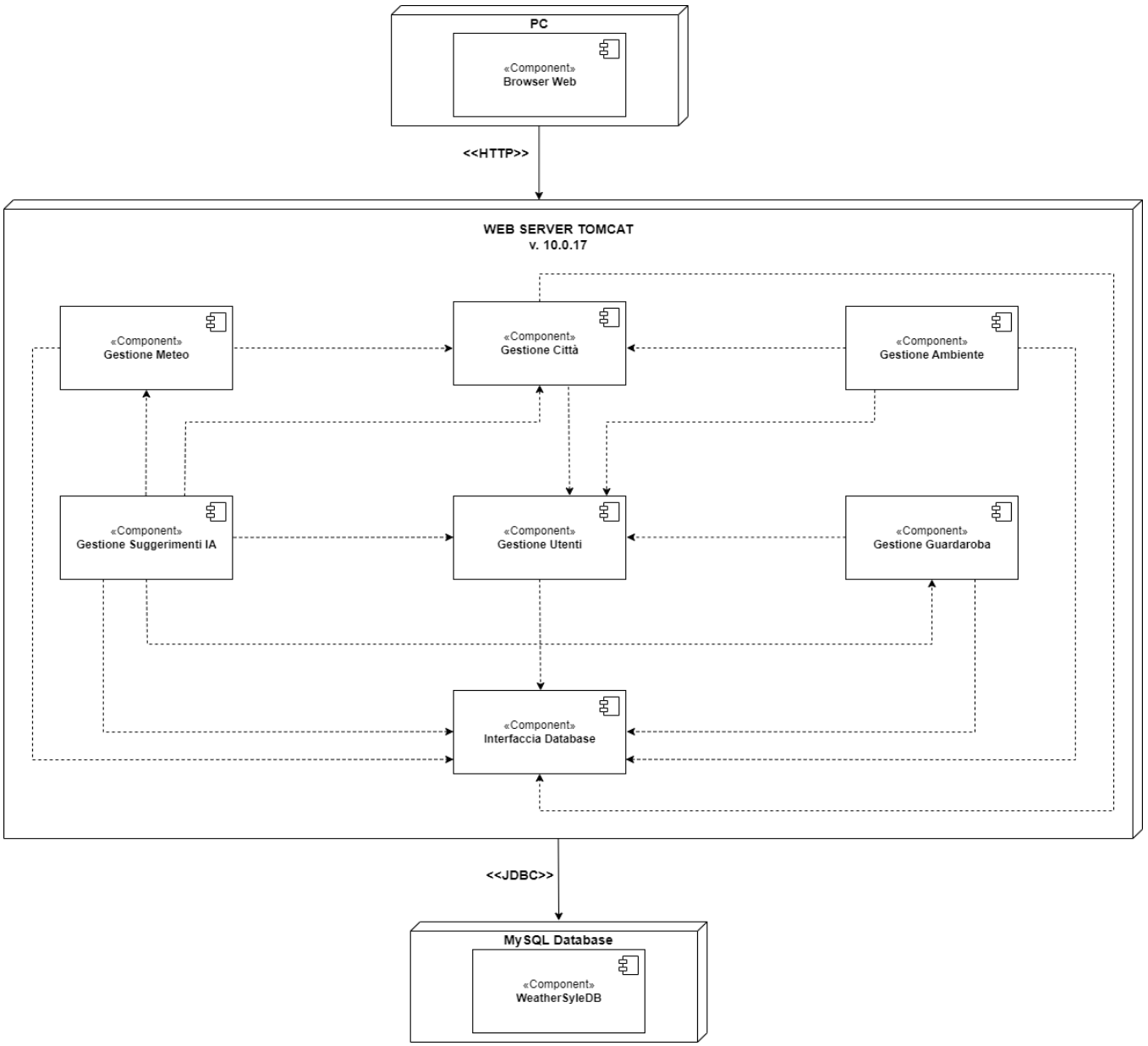


3.3 Mapping Hardware e Software

Il sistema che sarà sviluppato, trattandosi di un applicativo web, possiamo rappresentarlo attraverso tre nodi:

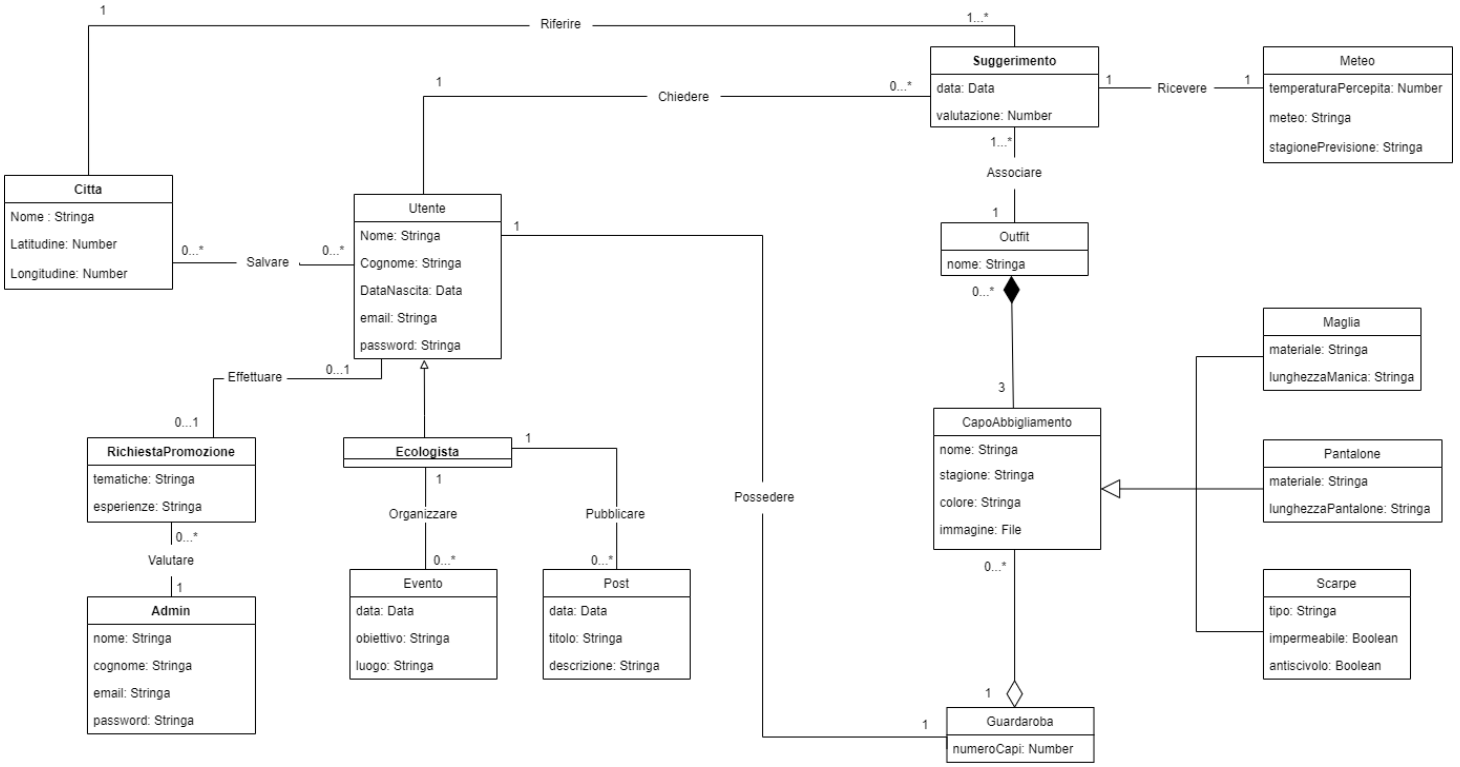
- **PC:** dove sarà installato un web browser con il quale sarà possibile inoltrare richieste HTTP al server su cui sarà installato l'applicativo.
- **WebServer Tomcat:** dove sarà installato l'applicativo che a sua volta comunicherà mediante Driver JDBC ad un Database.
- **MySQL Database:** dove sarà installato il database.

A seguire l'UML deployment diagram:



3.4 Gestione dei Dati Persistenti

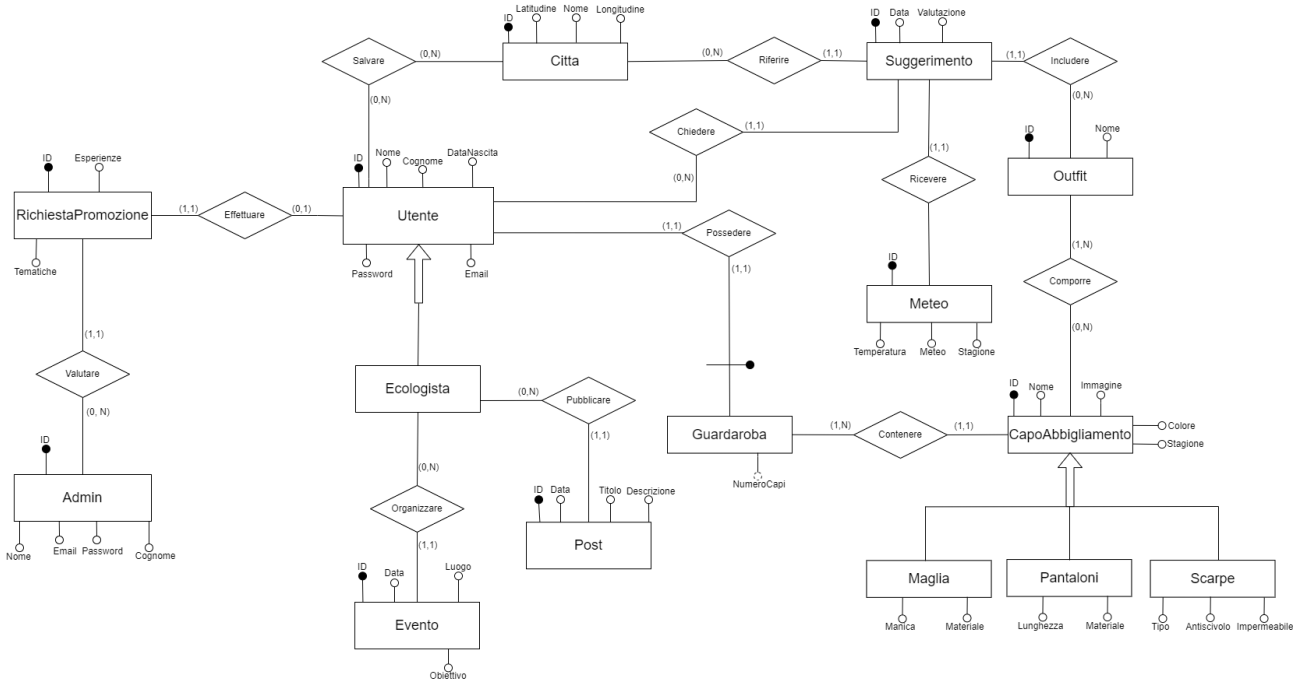
La gestione dei dati persistenti viene effettuata tramite un Database MySQL. Di seguito il Class Diagram, da cui viene poi estratto il diagramma E-R:



Tutte le classi sono state trasformate in Entità nel modello Entità - Relazione, e sono state apportate le seguenti modifiche:

- Per tutti gli oggetti Entity è stato aggiunto l'attributo ID, che costituisce la chiave primaria tranne per l'entità Guardaroba che è un'entità debole e usa come identificatore l'ID dell'entità Utente.
- Sono state trasformate le relazioni di ereditarietà tra le classi in generalizzazioni.

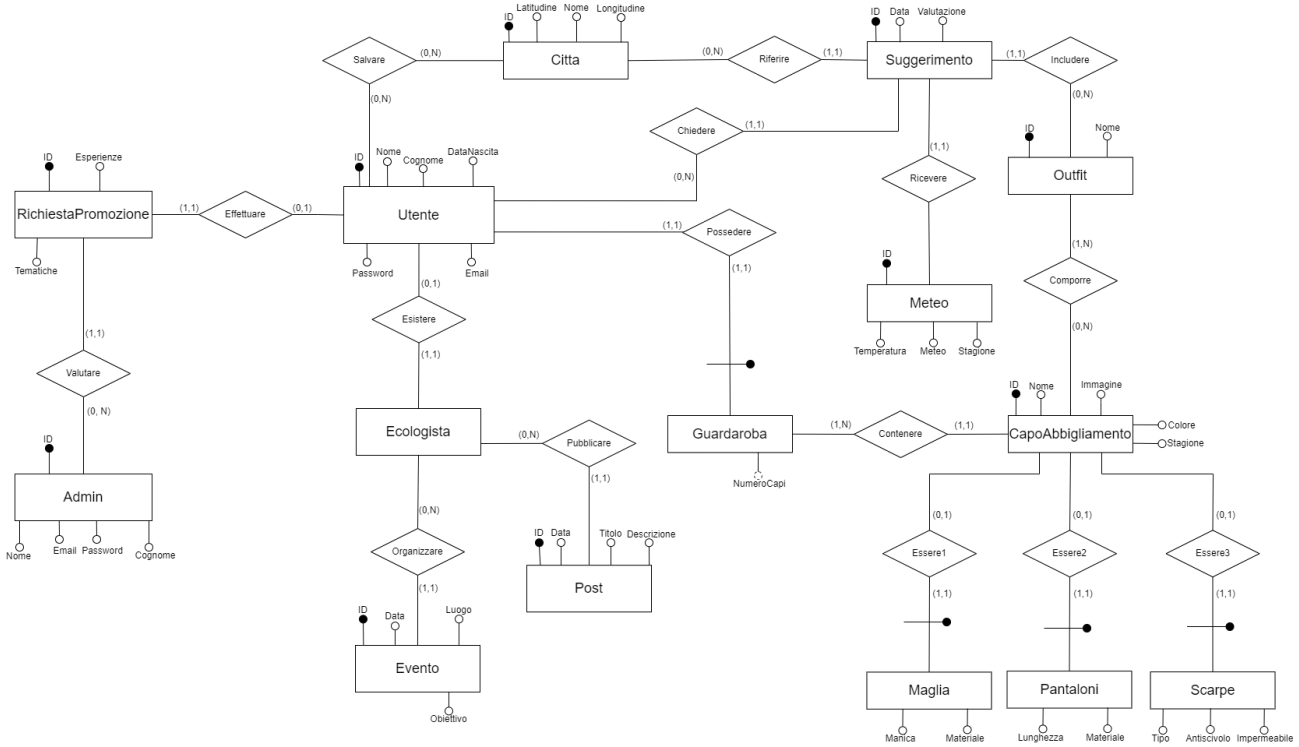
Il diagramma relazionale risultante è il seguente:



Le business rules a cui è soggetto il modello sono:

- L'Utente può effettuare una sola richiesta di promozione ad Ecologista.
- Un Outfit è composto da minimo tre e massimo tre capi d'abbigliamento.

Successivamente, a seguito della ristrutturazione, lo schema risultante è il seguente:



L'unica operazione di ristrutturazione effettuata è stata l'eliminazione delle generalizzazioni, aggiungendo le relazioni, in quanto le entità figlie hanno attributi o relazioni in più dell'entità genitore.

Schema logico

UTENTE (ID, nome, cognome, dataNascita, email, password)
CITTA (ID, nome, latit, longit)
SALVARE (IDutente, IDcitta)
SUGGERIMENTO (ID, data, valutazione, IDutente*,IDcitta*, IDoutfit*, IDMeteo*)
METEO(IDmeteo, temperatura, meteo, stagione)
OUTFIT (ID, nome)
CAPOABBIGLIAMENTO (ID, nome, immagine, stagione, colore, IDguardaroba*)
MAGLIA(IDcapoAbbigliamento, manica, materiale)
PANTALONI(IDcapoAbbigliamento, lunghezza, materiale)
SCARPE(IDcapoAbbigliamento, tipo, antiscivolo, impermeabile)
COMPORRE (IDoutfit, IDcapoAbbigliamento)
GUARDAROBA (ID, nome, numeroCapi)
RICHIESTAPROMOZIONE (ID, tematiche, esperienze, IDutente, IDadmin*)
ADMIN (ID, nome, cognome, email, password)
POST (ID, data, titolo, descrizione, IDutente*)
ECOLOGISTA (IDutente)
EVENTO (ID, data, luogo, obiettivo, IDutente*)

Vincoli di integrità referenziale
GUARDAROBA (IDguardaroba) VIR UTENTE (ID)
SALVARE (IDutente) VIR UTENTE (ID)
SALVARE (IDcitta) VIR CITTA (ID)
SUGGERIMENTO (IDutente) VIR UTENTE (ID)
SUGGERIMENTO (IDcitta) VIR CITTA (ID)
SUGGERIMENTO (IDoutfit) VIR OUTFIT (ID)
CAPOABBIGLIAMENTO (IDguardaroba) VIR GUARDAROBA (ID)
COMPORRE (IDoutfit) VIR OUTFIT (ID)
COMPORRE (IDcapoAbbigliamento) VIR CAPOABBIGLIAMENTO (ID)
RICHIESTAPROMOZIONE (IDutente) VIR UTENTE (ID)
RICHIESTAPROMOZIONE (IDadmin) VIR ADMIN (ID)
POST (IDutente) VIR UTENTE (ID)
ECOLOGISTA (IDutente) VIR UTENTE (ID)
EVENTO (IDutente) VIR UTENTE (ID)

3.5 Controllo degli accessi e sicurezza

Matrice degli accessi per ogni attore.

Oggetti \ Attori	Utente	Ecologista	Admin
Gestione Utenti	RegistrazioneUtente VisualizzaDatiUtente ModificaDatiUtente	VisualizzaDatiUtente ModificaDatiUtente	
Gestione Guardaroba	VisualizzaGuardaroba InserisciCapo VisualizzaDettagliVestiti EliminaCapo	VisualizzaGuardaroba InserisciCapo VisualizzaDettagliVestiti EliminaCapo	
Gestione Suggerimenti IA	SuggerisciCapiAbbigliamento VisualizzaCronologiaSuggerimenti ForniscFeedback FiltraCronologiaSuggerimenti VisualizzaDettaglioSuggerimento	SuggerisciCapiAbbigliamento VisualizzaCronologiaSuggerimenti ForniscFeedback FiltraCronologiaSuggerimenti VisualizzaDettaglioSuggerimento	
Gestione Meteo	VisualizzaMeteo	VisualizzaMeteo	
Gestione Città	SalvaNeiPreferiti RimuoviDaiPreferiti VisualizzaLocalitaPreferite	SalvaNeiPreferiti RimuoviDaiPreferiti VisualizzaLocalitaPreferite	
Gestione Ambiente	VisualizzaEventi VisualizzaDettaglioEventi VisualizzaPost PassaAdEcologista	VisualizzaEventi VisualizzaDettaglioEventi VisualizzaPost PubblicaPost PianificaEventi	ValutaRichiesta

3.6 Controllo del flusso globale

Il nostro sistema avrà un controllo del flusso globale di tipo **Event Driven**. WeatherStyle ha bisogno che l'Utente dia luogo ad un evento per poter avviare il suo flusso di esecuzioni. Ciò è possibile tramite delle interfacce grafiche che il sistema mette a disposizione dell'Utente e nelle quali quest'ultimo genera un evento tramite un insieme di azioni che può eseguire. Tale evento viene poi gestito dall'handler ad esso dedicato e verrà tramandato al suo sottosistema di riferimento, che gestisce la logica di application e storage qualora servisse, e infine provvederà a soddisfare la richiesta.

3.7 Condizioni Limite

Identificativo UC_BC1		*Avvio del sistema*	*Data*	07/12/2022
			Vers.	0.1
			Autore	Tutto il team
Descrizione		Lo Use Case dà la possibilità di avviare il sistema.		
Attore Principale		Admin		
Entry Condition		L'Admin accede al server Tomcat. AND L'Admin accede al server MySQL.		
Exit condition On success		Il sistema viene avviato correttamente.		
Exit condition On failure		Il sistema non viene avviato.		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO				
1	**Admin**	Esegue il comando di avvio di una connessione al database su server MySQL.		
2	**Sistema**	Verifica la sanità dei dati persistenti.		
3	**Admin**	Esegue il comando di avvio dell'applicativo su server Tomcat.		
4	**Sistema**	Rende disponibili le sue funzionalità e i suoi servizi agli utenti.		
Scenario/Flusso di eventi Alternativo: *i dati persistenti sono danneggiati*				
2.a1	**Sistema**	Mostra all'Admin un messaggio che informa dei problemi ai dati persistenti.		
2.a2	**Admin**	Corregge i dati presenti nel database.		
2.a3	**Admin**	Riesegue il passaggio 1.		

Identificativo UC_BC2	*Spegnimento del sistema*	*Data*	07/12/2022
		Vers.	0.1
		Autore	Tutto il team
Descrizione	Lo Use Case dà la possibilità di spegnere completamente il sistema.		
Attore Principale	Admin		
Entry Condition	L'Admin accede al server Tomcat . AND L'admin accede al server MySQL. AND il sistema era stato avviato precedentemente.		
Exit condition On success	Il sistema viene spento correttamente.		
Exit condition On failure	Il sistema non viene spento.		

FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	**Admin**	Esegue il comando di stop dell'applicativo dal server Tomcat.
2	**Sistema**	Controlla che non ci siano connessioni aperte verso l'esterno, in caso affermativo termina l'esecuzione dell'applicativo.
3	**Admin**	Esegue il comando di disconnessione al database dal server MySQL.
4	**Sistema**	Chiude la connessione al database.
Scenario/Flusso di eventi Alternativo: *sono presenti connessioni ancora aperte verso l'esterno*		
2.a1	**Sistema**	Mostra all'Admin un messaggio che informa della presenza di connessioni ancora aperte verso l'esterno.
2.a2	**Sistema**	Attende una quantità di tempo in cui non riceve altre richieste di connessione e porta a termine richieste già in corso.
2.a3	**Sistema**	Notifica l'Admin della corretta terminazione dell'applicativo.

4. Servizi dei sottosistemi

Gestione Meteo

Servizio	Descrizione	Interfaccia
Ottieni Previsioni meteo di una città	Servizio che permette di ottenere le previsioni meteo di una città.	GestioneMeteoService
Salva Previsioni Meteo	Servizio che permette il salvataggio di previsioni meteo nel DB.	GestioneMeteoDAO
Ottieni una previsione meteo salvata nel DB	Servizio che permette di ottenere una previsione meteo salvata nel DB.	GestioneMeteoDAO

Gestione Città

Servizio	Descrizione	Interfaccia
Ottieni elenco città con un certo prefisso	Servizio che permette di ottenere l'elenco di città con un certo prefisso.	GestioneCittaService
Ottieni le coordinate di una città	Servizio che permette di ottenere le coordinate di una certa città.	GestioneCittàService
Salva una città nel DB	Servizio che permette di salvare una città nel DB.	GestioneCittàDAO
Salva una città preferita di un utente	Servizio che permette di salvare una città preferita di un utente.	GestioneCittàDAO

Gestione Guardaroba

Servizio	Descrizione	Interfaccia
Salva un capo d'abbigliamento nel DB	Servizio che permette di salvare un capo d'abbigliamento nel DB.	GestioneGuardarobaDAO
Ottieni un capo d'abbigliamento salvato nel DB	Servizio che permette di ottenere un capo d'abbigliamento nel DB.	GestioneGuardarobaDAO
Elimina un capo d'abbigliamento salvato nel DB	Servizio che permette di rimuovere un capo d'abbigliamento dal DB.	GestioneGuardarobaDAO

Gestione Suggerimento IA

Servizio	Descrizione	Interfaccia
Ottieni capi d'abbigliamento suggeriti	Servizio che permette di ottenere i capi d'abbigliamento suggeriti in base alle previsioni meteo.	GestioneSuggerimentoIALogic
Ottieni cronologia suggerimenti	Servizio che permette di ottenere la cronologia dei suggerimenti proposti.	GestioneSuggerimentoIADAO
Salva feedback suggerimento nel DB	Servizio che permette di salvare un feedback per il suggerimento nel DB.	GestioneSuggerimentoIADAO
Salva un outfit nel DB	Servizio che permette di salvare un outfit nel DB.	GestioneSuggerimentoIADAO
Salva un suggerimento nel DB	Servizio che permette di salvare un suggerimento nel DB.	GestioneSuggerimentoIADAO

Gestione Ambiente

Servizio	Descrizione	Interfaccia
Ottieni eventi salvati nel DB	Servizio che permette di ottenere gli eventi salvati nel DB.	GestioneAmbienteDAO
Ottieni post salvati nel DB	Servizio che permette di ottenere i post salvati nel DB.	GestioneAmbienteDAO
Salva post nel DB	Servizio che permette di salvare un post nel DB.	GestioneAmbienteDAO
Salva evento nel DB	Servizio che permette di salvare un evento nel DB.	GestioneAmbienteDAO
Salva richiesta promozione nel DB	Servizio che permette di salvare una richiesta di promozione nel DB.	GestioneAmbienteDAO
Ottieni richieste di promozione dal DB	Servizio che permette di ottenere le richieste di promozione dal DB da parte di un Admin.	GestioneAmbienteDAO

Gestione Utenti

Servizio	Descrizione	Interfaccia
Salva Utente nel DB	Servizio che permette di salvare un utente nel DB.	GestioneUtenteDAO
Ottieni Utente dal DB	Servizio che permette di ottenere un utente dal DB.	GestioneUtenteDAO
Ottieni Admin dal DB	Servizio che permette di ottenere un admin dal DB.	GestioneUtenteDAO
Salva ecologista nel DB	Servizio che permette di salvare un ecologista nel DB.	GestioneUtenteDAO