

SDD: SYSTEM DESIGN DOCUMENT

C05

DressMe Now

Riferimento	
Versione	1.0
Data	24/11/2023
Destinatario	Prof.ssa F. Ferrucci
Presentato da	Luigi Allocca, Luigi Emanuele Sica
Approvato da	

Revision History

Data	Versione	Descrizione	Autori
25/11/2023	0.1	Creazione	Tutti i membri del team
26/11/2023	0.2	Aggiornamento	Tutti i membri del team
27/11/2023	0.3	Aggiornamento	Tutti i membri del team
27/11/2023	0.4	Aggiornamento	Alfredo Napoli, Carlo di Vico
27/11/2023	0.5	Aggiornamento	Alfredo Baratta, Aniello Tremulo, Guerino Pio Altieri
28/11/2023	0.6	Aggiornamento	Carlo di Vico, Christian Iervasi, Aniello Tremulo
30/11/2023	0.7	Aggiornamento	Tutti i membri del team
2/12/2023	0.8	Aggiornamento	Pietro Lagioia
17/12/2023	0.9	Revisione	Carlo di Vico
17/12/2023	1.0	Prima stesura completa	Tutti i membri del team
22/12/2023	1.1	Aggiornamento	Alfredo Baratta, Alfredo Napoli
19/01/2024	1.2	Revisione	Carlo di Vico

Sommario

1. Obiettivo del sistema
2. Design Goals
 - 2.1. Design Trade-off
3. Organizzazione del Documento
4. Architettura del Sistema corrente
5. Architettura del Sistema Proposto
 - 5.1. Decomposizione in sottosistemi
 - 5.1.1. Deployment Diagram
 - 5.2. Mapping hardware/software
 - 5.3. Gestione dati persistenti
 - 5.4. Controllo degli accessi di sicurezza
 - 5.5. Controllo flusso globale del sistema
 - 5.6. Condizioni Limite
6. Servizi dei sottosistemi
7. Class Diagram

Obiettivi del sistema (Tutti i membri del team)

Il nostro sito web lavora nell'industria della moda e-commerce, con l'obiettivo di portare avanti l'innovazione nel settore dell'abbigliamento sostenibile. Abbiamo l'ambizione di migliorare l'esperienza di shopping degli utenti, introducendo un elemento innovativo: un avatar 2D personalizzabile. Questo avatar non è solo un'aggiunta estetica, ma un'importante funzione che permette agli utenti di provare i prodotti virtualmente prima di effettuare un acquisto. Questo approccio non solo potenzialmente riduce il numero di resi, ma contribuisce anche a ridurre l'impatto ambientale complessivo della moda.

Stiamo quindi lavorando per creare un'esperienza di shopping personalizzata e coinvolgente. Il nostro avatar 2D personalizzabile non solo permette agli utenti di provare i nostri prodotti virtualmente, ma offre anche un modo per esprimere la loro personalità e il loro stile. Questo approccio coinvolgente può aiutare a creare un legame più forte con i nostri clienti e a promuovere i nostri prodotti.

Il nostro sito si distingue per la sua dedizione all'eco-sostenibilità. Abbiamo l'obiettivo di minimizzare l'impatto ambientale dei nostri prodotti, sia nel ciclo di vita del prodotto che nel processo di produzione. Per fare ciò, stiamo lavorando per garantire che i nostri prodotti siano realizzati con materiali sostenibili e che il nostro processo di produzione sia il più efficiente possibile.

Inoltre, utilizziamo tecnologie innovative basate sull'intelligenza artificiale per rendere più efficiente la ricerca di prodotti in base a determinate specifiche fornite dall'utente. In questo modo cerchiamo di rendere più efficiente tale meccanismo e la proposta dei singoli prodotti, in modo da soddisfare a pieno le esigenze del cliente.

Design Goals (Tutti i membri del team)

Priorità	ID Design Goals	Descrizione Design Goals	Categoria	RNF di Origine
Alta	DG_1 Usabilità	Il sistema deve garantire una semplice ed intuitiva visualizzazione delle pagine e delle funzionalità proposte.	End user	RNF_1 RNF_3
Alta	DG_2 Usabilità	Il sistema deve essere responsive in modo da potersi adattare a qualsiasi dispositivo ed essere utilizzato in diverse situazioni.	End user	RNF_2
Alta	DG_3 Usabilità	Il sistema deve fornire chiari messaggi di errore qualora si dovesse verificare qualche imprevisto (es. input utente non valido oppure errori di sistema).	Dependability	RNF_4
Alta	DG_4 Manutenibilità	Il sistema deve essere scritto in modo chiaro, utilizzando design pattern specifici e deve garantire la modularità del codice stesso.	Maintenance	RNF_5 RNF_6
Alta	DG_5 Prestazioni	Il sistema deve garantire la sua corretta scalabilità.	Performance	RNF_7
Alta	DG_6 Prestazioni	Il sistema deve offrire tempi di risposta inferiore ai 3 secondi.	Performance	RNF_8
Alta	DG_7 Prestazioni	Il sistema deve sfruttare le best practices del linguaggio utilizzato (Python).	Performance	RNF_9
Alta	DG_8 Prestazioni	Il sistema deve essere realizzato in modo da non presentare code-	Performance	RNF_10

		smells.		
Alta	DG_9 Prestazioni	Il sistema deve garantire un accuracy molto precisa del modulo AI.	Performance	RNF_11
Alta	DG_10 Sicurezza	Il sistema deve garantire l'hash delle password utente.	Dependability	RNF_12
Alta	DG_11 Sicurezza	Il sistema deve garantire l'uso del certificato SSL per la trasmissione delle informazioni, in modo tale da proteggere i dati inviati dagli utenti.	Dependability	RNF_13
Alta	DG_12 Sicurezza	Il sistema deve prevedere un criterio per l'inserimento della password, in modo tale da aumentarne la complessità.	Dependability	RNF_14
Alta	DG_13 Sicurezza	Il sistema deve garantire il corretto filtraggio degli input inseriti dall'utente, in modo tale da ridurre il sorgere di potenziali problematiche.	Dependability	RNF_15
Alta	DG_14 Affidabilità	Il sistema deve essere sempre disponibile e accessibile agli utenti per la maggior parte del tempo e deve essere in grado di recuperarsi automaticamente da errori e guasti senza perdita di dati o servizi interrotti dai clienti senza procurare danni al sistema stesso né all'utente	Dependability	RNF_16 RNF_17
Alta	DG_15 Supportabilità	Il sistema deve garantire la completa compatibilità con una vasta gamma di browser web.	Dependability	RNF_18
Alta	DG_16	Il sistema deve garantire	Dependability	RNF_19

	Supportabilità	la completa compatibilità con diversi sistemi operativi quali Linux, Windows e macOS.		
--	----------------	--	--	--

Design Trade-off (Aniello Tremulo, Alfredo Napoli)

Utilizzo di memoria - Velocità del sistema

Per offrire agli utenti un'esperienza rapida e reattiva, abbiamo deciso di attribuire maggiore importanza alla velocità del sistema rispetto all'utilizzo della memoria.

Costi - Ridondanza

Come da obiettivi, il sistema deve essere ridondante: bisogna quindi avere un'architettura in grado di resistere ad errori o guasti e prevenire la perdita di dati per evitare danni sia all'utente sia al sistema stesso. È stato scelto di dare maggior importanza alla ridondanza rispetto ai costi.

Tempi di rilascio - Funzionalità

Se i tempi di rilascio sono stretti possono essere rilasciate meno funzionalità di quelle richieste. È stato scelto di dare maggior importanza ai tempi di rilascio rispetto alle funzionalità.

Tempi di rilascio - Qualità

Se i tempi di rilascio sono stretti possono essere rilasciate funzionalità di qualità inferiore ma pur sempre valide e implementate correttamente (es. Avatar implementato in 2D piuttosto che in 3D per evitare dilatazioni temporali eccessive).

Architettura del Sistema Proposto (Tutti i membri del team)

Il nostro sistema è una piattaforma di e-commerce nel settore della moda, focalizzata sull'abbigliamento sostenibile e sull'innovazione nell'esperienza di shopping. Al fine di supportare la nostra visione, stiamo adottando un'architettura Three-Tier, progettata per fornire una piattaforma modulare e scalabile.

- **Tier di Presentazione (Presentation):** Questo strato rappresenta l'interfaccia utente del nostro sito. L'elemento chiave è l'avatar 2D personalizzabile, che consente agli utenti di provare virtualmente i nostri prodotti. Questo strumento non solo mira a migliorare l'esperienza di shopping, ma anche a ridurre i resi e l'impatto ambientale.
- **Tier dell'Applicazione (Application):** Nel cuore del sistema si trova il tier dell'applicazione, dove avviene l'elaborazione dei dati. Qui, ad esempio, l'avatar 2D interagisce con la logica di business, permettendo agli utenti di personalizzare l'esperienza di shopping e di cercare prodotti in modo più efficiente attraverso l'uso di tecnologie AI.
- **Tier dei Dati (Storage):** Questo livello gestisce e archivia tutte le informazioni relative ai prodotti e alle preferenze degli utenti. Utilizzeremo un database relazionale (MySQL) per garantire l'efficienza nella gestione dei dati e per supportare le nostre funzionalità di ricerca avanzata.

Siamo orientati a questa struttura Three-Tier per vari motivi:

- **Sviluppo rapido:** La separazione modulare ci consente di sviluppare ogni tier in modo indipendente, consentendo un rapido progresso e l'adozione delle migliori pratiche per ciascuna area.
- **Scalabilità efficiente:** Ogni strato può essere scalato in modo indipendente, garantendo che il sistema cresca senza compromettere le prestazioni o la funzionalità.
- **Affidabilità ottimale:** L'isolamento dei tier riduce il rischio di interruzioni e migliora la disponibilità del sistema nel suo complesso.
- **Sicurezza avanzata:** L'architettura separata impedisce l'accesso diretto tra il livello di presentazione e quello dei dati, aumentando la sicurezza complessiva del sistema contro potenziali vulnerabilità.

Inoltre, la nostra piattaforma si distingue per il suo impegno verso l'eco-sostenibilità, con l'obiettivo di minimizzare l'impatto ambientale attraverso materiali sostenibili e un processo di produzione efficiente. L'uso delle tecnologie AI ci consente di personalizzare l'esperienza di ricerca e offrire prodotti che soddisfino al meglio le esigenze dei nostri clienti, promuovendo così la nostra filosofia di shopping sostenibile e coinvolgente.

Tuttavia, nonostante i vari vantaggi già elencati, la scelta dell'architettura Three-Tier comporta anche diversi svantaggi, tra cui abbiamo:

- **Complessità:** Anche se la separazione dei tier favorisce lo sviluppo modulare, può portare a un aumento della complessità del sistema. La gestione di più layer richiede una pianificazione accurata e una coordinazione stretta tra i membri del team, il che potrebbe portare ad una maggiore complessità nel tempo.
- **Overhead di comunicazione:** L'interazione tra i diversi tier richiede comunicazione tra di essi. Se non gestita correttamente, questa comunicazione può causare problemi (come rallentamenti nel sistema), specialmente se ci sono scambi di dati tra i livelli.
- **Manutenzione e debugging:** Avere più tier rende più complicata la manutenzione e il debugging del sistema. Identificare e risolvere i problemi che coinvolgono più livelli può richiedere maggiore dispendio per quanto riguarda tempo e risorse.
- **Costi:** L'architettura Three-Tier potrebbe richiedere più risorse in termini di hardware, sviluppo e manutenzione.
- **Requisiti di sicurezza aggiuntivi:** L'architettura Three-Tier richiede la gestione e la manutenzione di regole di sicurezza maggiori per garantire che ogni tier sia protetto adeguatamente.
- **Sfide nell'integrazione di nuove funzionalità:** Se una nuova funzionalità richiede l'integrazione e l'interazione tra più tier, potrebbe essere necessario apportare modifiche ai livelli del sistema, comportando un aumento del tempo e delle risorse necessarie ai fini implementative.
- **Limitazioni nell'adattabilità e flessibilità:** Sebbene l'architettura sia progettata per la scalabilità e l'isolamento dei componenti, potrebbe presentare limitazioni nell'adattabilità a cambiamenti significativi nell'infrastruttura o nelle esigenze di business.

Sottoinsiemi dell'Architettura Proposta (Tutti i membri del team)

Decomposizione dei Sottosistemi nell'Architettura Proposta:

La nostra architettura Three-Tier si traduce in una decomposizione dei sottosistemi che mira a separare la logica di Presentazione, di Business e di Storage. Questa struttura consente una gestione efficiente delle funzionalità specifiche, garantendo basso accoppiamento e alta coesione tra i componenti del sistema.

Tier di Presentazione:

- **InterfacciaUtente:** Questo sottosistema gestisce l'interfaccia diretta con l'utente, consentendo la visualizzazione e l'interazione con l'avatar 2D personalizzabile e altre funzionalità di navigazione e interazione dell'utente con i prodotti.
- **InterfacciaDirettore:** Questo sottosistema gestisce l'interfaccia diretta con l'amministratore, consentendo la visualizzazione, la modifica, ed in generale la gestione dei gestori, delle statistiche e del corretto funzionamento del sistema.
- **InterfacciaGestore:** Questo sottosistema gestisce l'interfaccia diretta con i gestori, consentendo la visualizzazione, la modifica, ed in generale la gestione degli ordini e dei prodotti.

Tier dell'Applicazione:

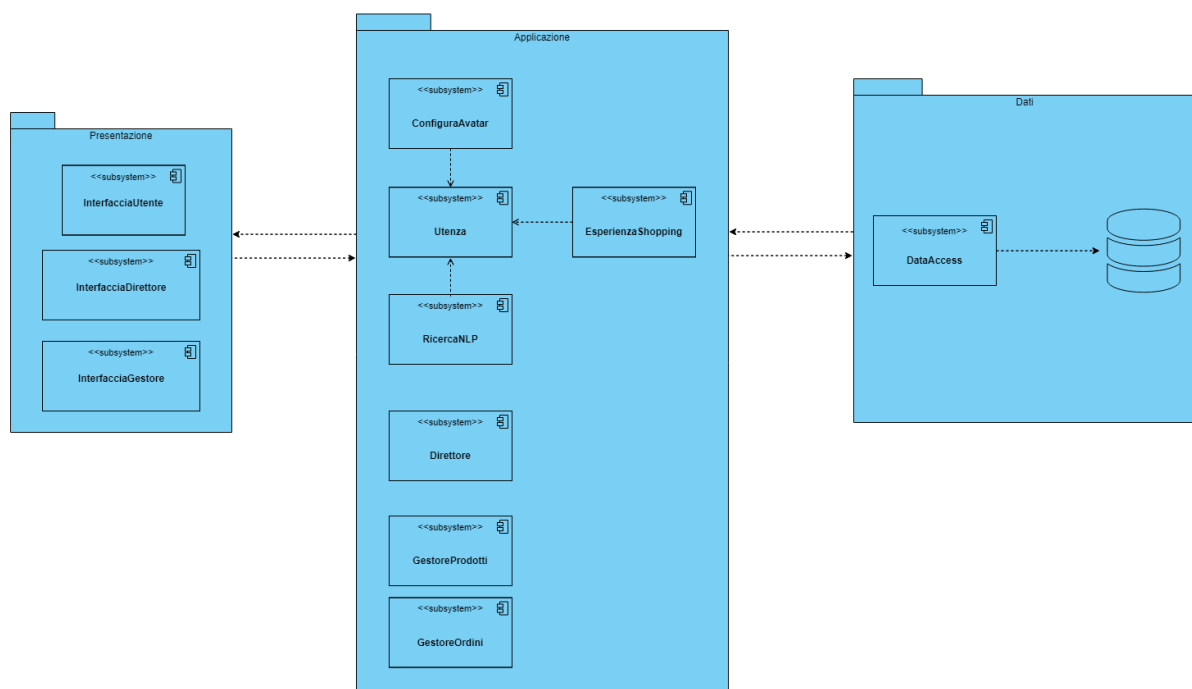
- **Utenza:** È il sottosistema responsabile della gestione delle operazioni relative ai singoli utenti ed è caratterizzato da funzionalità quali ad esempio il login, il logout e la registrazione degli utenti. Gestisce l'autenticazione e l'autorizzazione degli utenti registrati.
- **Direttore:** È il sottosistema responsabile della gestione delle operazioni di gestione degli utenti, dei loro privilegi e della visualizzazione dei dati statistici del sistema.
- **EsperienzaShopping:** È il sottosistema responsabile della gestione delle operazioni generali relative ai prodotti ed è caratterizzato da funzionalità quali, ad esempio, l'acquisto, la visualizzazione e la restituzione.
- **ConfiguraAvatar:** È il sottosistema responsabile della gestione dell'avatar di ogni singolo utente ed è caratterizzato da funzionalità quali ad esempio la presentazione e la personalizzazione dei prodotti attraverso l'avatar 2D. Si occupa delle operazioni che consentono agli utenti di configurare il proprio avatar.
- **RicercaNLP:** È il sottosistema responsabile della logica di ricerca dei prodotti utilizzando un modulo di intelligenza artificiale per comprendere lo stile dell'utente e mostrare prodotti più adatti alla sua ricerca.
- **GestoreOrdini:** È il sottosistema responsabile della gestione delle operazioni di gestione degli ordini e presenta funzionalità quali ad esempio la modifica e la cancellazione di determinati ordini.

- **GestoreProdotti:** È il sottosistema responsabile della gestione delle operazioni di gestione dei prodotti e presenta funzionalità quali ad esempio l'aggiunta, la modifica e l'eliminazione di prodotti.

Tier dei Dati:

- **DataAccess:** Questo sottosistema funge da ponte tra l'applicazione e il database relazionale (MySQL). Si occupa dell'accesso ai dati e della loro memorizzazione persistente, garantendo un'efficace gestione dei dati relativi ai prodotti, alle preferenze degli utenti e alle informazioni di sistema.

Questa struttura di sottosistemi, all'interno della nostra architettura proposta, mira a garantire una gestione chiara e distinta delle funzionalità del sistema, consentendo uno sviluppo modulare e scalabile in linea con gli obiettivi dell'e-commerce dedicato alla moda sostenibile e all'innovazione dell'esperienza di shopping.



Deployment Diagram (Tutti i membri del team)

Il deployment diagram mette in risalto la disposizione fisica dei componenti del sistema e le relazioni tra di essi durante l'esecuzione del software.

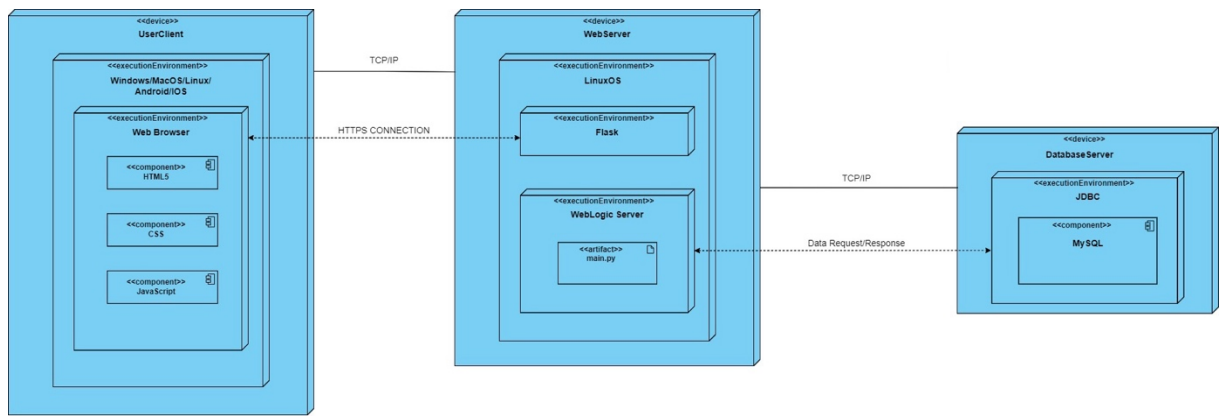
Componenti del Deployment:

- **User Client:** Rappresenta l'utente finale che interagisce con il sistema attraverso un Web Browser. Il Client richiede funzionalità attraverso l'interfaccia web, che richiede la capacità di interpretare HTML, CSS e JavaScript per fruire delle diverse funzionalità offerte dal software.
- **Web Server:** Il Web Server funge da punto di accesso per il Client. Gestisce le richieste provenienti dall'interfaccia web dell'utente e inoltra tali richieste alla componente Web Logic per l'elaborazione e l'esecuzione.
- **Web Logic:** Questa componente riceve le richieste provenienti dal Web Server e soddisfa le funzionalità richieste dal Client. Inoltre, gestisce la connessione al database, interagendo con esso per l'accesso e la manipolazione dei dati richiesti dalle operazioni dell'utente.
- **Database:** Il database è il componente che conserva e gestisce i dati del sistema. La connessione al database è fornita dalla componente Web Logic, che interagisce con il database per l'accesso, la memorizzazione e l'aggiornamento dei dati richiesti dall'applicazione.

Interazioni del Deployment:

- Il Client si collega al Web Server attraverso un Web Browser, inoltrando richieste per accedere alle funzionalità dell'applicazione.
- Il Web Server riceve le richieste dal Client e le inoltra alla Web Logic per l'elaborazione.
- La Web Logic soddisfa le richieste del Client, processa le operazioni richieste e gestisce la connessione al database per l'accesso e la gestione dei dati.
- Il Database archivia e gestisce i dati richiesti dall'applicazione, fornendo un ambiente sicuro e affidabile per la memorizzazione dei dati persistenti.

Questo deployment diagram rappresenta la disposizione fisica dei componenti e le relazioni di connessione durante il funzionamento del sistema, evidenziando le interazioni tra Client, Web Server, Web Logic e il Database.



Mapping Hardware e Software (Tutti i membri del team)

Il nostro sistema si basa su una struttura hardware composta da client e server, ciascuno con requisiti specifici per il corretto funzionamento del sistema.

Client:

- **Specifiche:** Qualsiasi dispositivo con accesso a Internet e un web browser installato. Non richiede requisiti hardware particolarmente avanzati e può essere un computer desktop, un laptop, uno smartphone o qualsiasi altro dispositivo connesso a Internet.

Server:

- Una macchina con connessione a Internet e capacità di memorizzazione elevata per gestire una grande quantità di dati. Può essere un server dedicato, cloud server o una macchina con specifiche hardware robuste per sostenere il carico di lavoro.

Comunicazione:

- **Protocollo:** Lo scambio dati avviene tramite il protocollo HTTP, consentendo al client di inviare richieste al server e ricevere risposte da quest'ultimo. Questo protocollo è ampiamente utilizzato per le comunicazioni web, garantendo la trasmissione affidabile dei dati attraverso Internet.

Database Management System (DBMS):

- Il sistema richiede un DBMS per la gestione dei dati persistenti, consentendo la comunicazione efficiente tra più client. Nel contesto delle richieste di accesso e memorizzazione dei dati, il DBMS svolge un ruolo cruciale nella struttura del sistema, garantendo la coerenza e l'accesso affidabile ai dati.

Questo setup hardware/software offre una struttura flessibile e scalabile per il sistema, consentendo un'interazione efficiente tra client e server attraverso Internet, garantendo al contempo la sicurezza e l'affidabilità delle operazioni di comunicazione e memorizzazione dei dati.

Gestione Dati Persistenti (Alfredo Napoli, Carlo Di Vico)

Il nostro sistema è progettato per gestire dati persistenti attraverso l'utilizzo di un database relazionale, nello specifico MySQL. Tale scelta è basata su specifiche esigenze e considerazioni che sottolineano alcuni vantaggi offerti dai database relazionali rispetto a quelli non relazionali.

Caratteristiche e Vantaggi del Database Relazionale:

- **Struttura Organizzata:** MySQL offre una struttura ben organizzata con tabelle interconnesse, consentendo relazioni complesse tra i dati. L'uso di "join" permette di recuperare informazioni da tabelle diverse in base alle relazioni definite.
- **Consistenza e Integrità dei Dati:** Grazie alle regole di integrità referenziale e alle transazioni, MySQL garantisce la coerenza e l'integrità dei dati, riducendo il rischio di informazioni inconsistenti o frammentate.
- **Affidabilità e Sicurezza:** MySQL fornisce meccanismi di sicurezza avanzati per la gestione degli accessi e dei privilegi, proteggendo i dati da accessi non autorizzati. Inoltre, offre soluzioni di backup e recovery per garantire l'integrità dei dati in caso di guasti o perdita.
- **Query Ottimizzate:** Grazie alla possibilità di eseguire query complesse e ottimizzate, MySQL offre un'efficienza notevole nella manipolazione dei dati, garantendo tempi di risposta adeguati anche con volumi di dati considerevoli.

Visualizzazione dello Schema e Formato di Memorizzazione:

- **Struttura Tabellare:** Lo schema del database relazionale in MySQL è rappresentato attraverso tabelle collegate tra loro. Ogni tabella contiene righe di dati correlate, seguendo un formato tabellare che semplifica la comprensione della struttura dei dati.
- **Linguaggio SQL (Structured Query Language):** MySQL utilizza il linguaggio SQL per la gestione dei dati, facilitando operazioni di lettura, scrittura, aggiornamento e cancellazione dei dati, offrendo al contempo un elevato grado di flessibilità e precisione nella gestione delle informazioni.

La scelta di utilizzare MySQL, un robusto sistema di gestione di database relazionali, permette al nostro sistema di beneficiare delle caratteristiche distintive di questa tecnologia, garantendo affidabilità, flessibilità e sicurezza nella gestione dei dati persistenti del nostro sistema.

Indirizzo						
id_indirizzo	id_utente	provincia	cap	via	tipo	città

Utente								
id_utente	nome	cognome	email	password	data_nascita	telefono	sex	is_deleted

Configurazione avatar											
id_avatar	color_pelle	colorocchi	colorcapelli	lunghezza_capelli	altezza	peso	barba	sex	età	dimensioni_corpo	id_utente

Personale			
Id_personale	email	password	Tipo_personale

Transazione					
id_utente	id_ordine	id_transazione	data	totale	stato

Ordine					
id_utente	id_ordine	stato	data	reso	Note reso

Prodotto in ordine						
id_ordine	id_prodotto	Id_taglia	quantita	reso	Stato reso	Note reso

Prodotto									
id_prodotto	nome	categoria	marca	descrizione	vestibilità	prezzo	colore	materiale	Is_deleted

Immagine			
id_immagine	id_prodotto	immagine	tipo

Taglia	
id_taglia	Nometaglia

Taglia_prodotto		
id_taglia	Id_prodotto	quantità

Carrello	
id	id_utente

ProdottoInCarrello			
Id_carrello	id_prodotto	quantità	Id_taglia

Controllo degli Accessi e Sicurezza (Alfredo Baratta, Aniello

Tremulo, Guerino Pio Altieri)

All'interno del nostro sistema vi è un unico attore che si serve delle diverse funzionalità che il sistema mette a disposizione. Attraverso il meccanismo dell'Access Control List, abbiamo stilato una lista di coppie [Attore, Operazione], in modo che sia più chiaro quali sono le funzionalità di cui l'utente del sito "DressMeNow" si può servire.

Lista di controllo degli accessi per il sottosistema Utenza:

Attore	Operazione
Utente	Login()
Utente	Logout()
Utente	Registrazione()
Utente	CancellaAccount()

Lista di controllo degli accessi per il sottosistema GestioneProdotti:

Attore	Operazione
Utente	AggiungiProdottoCarrello()
Utente	VisualizzaProdotto()
Utente	RimuoviProdottoCarrello()
Utente	SvuotaCarrello()
Utente	CercaProdotto()

Lista di controllo degli accessi per il sottosistema ConfiguraAvatar:

Attore	Operazione
Utente	ConfiguraAvatar()

Lista di controllo degli accessi per il sottosistema GestoreOrdini:

Attore	Operazione
Personale	AggiungiOrdine()
Personale	ModificaOrdine()
Personale	CreaReso()
Personale	ModificaReso()

Lista di controllo degli accessi per il sottosistema GestoreProdotti:

Attore	Operazione
Personale	AggiungiProdotto()
Personale	EliminaProdotto()
Personale	ModificaProdotto()
Personale	AggiungiImmagineProdotto()
Personale	RimuoviImmagineProdotto()
Personale	AggiungiTaglia()
Personale	RimuoviTaglia()
Personale	ModificaTaglia()
Personale	AggiungiImmaginePerAvatar()
Personale	RimuoviImmaginePerAvatar()

Lista di controllo degli accessi per il sottosistema Direttore:

Attore	Operazione
Personale	AggiungiGestoreOrdine()
Personale	RimuoviGestoreOrdine()
Personale	AggiungiGestoreProdotto()
Personale	RimuoviGestoreProdotto()

Controllo flusso globale del sistema (Carlo Di Vico, Christian

Iervasi, Aniello Tremulo)

Il nostro sistema è progettato per una costante interazione con l'utente e per questo adotta un approccio di controllo del flusso basato su eventi, noto come "event-driven". Questo tipo di controllo del flusso si focalizza sullo scatenarsi e sulla gestione degli eventi che avvengono all'interno del sistema.

Caratteristiche dell'Approccio Event-Driven:

- **Interazione Continua:** Il sistema è progettato per reagire e gestire eventi in risposta alle azioni dell'utente o a variazioni interne. Ogni azione o interazione dell'utente è considerata un evento che avvia una serie di processi all'interno del sistema.
- **Reattività agli Eventi:** Gli eventi, che possono derivare dalle azioni dell'utente o da stimoli interni al sistema, fungono da inneschi per l'avvio di processi specifici o di flussi di lavoro predefiniti.
- **Gestione Dinamica del Flusso:** Il controllo del flusso è determinato dalle interazioni degli utenti e dai vari eventi scatenati all'interno del sistema. Ciò permette una gestione flessibile e dinamica delle azioni e dei processi che avvengono in base alle circostanze e alle richieste dell'utente.
- **Scalabilità e Reattività:** Questo approccio è altamente scalabile e reattivo, in quanto consente al sistema di adattarsi facilmente a varie situazioni e richieste degli utenti, permettendo una gestione efficiente anche di un alto volume di eventi simultanei.

Condizioni limite (Aniello Tremulo, Carlo Di Vico, Christian Iervasi)

Per il primo avvio del sistema è fondamentale che tutto venga gestito da un web server collegato ad un database. Una volta che il web server viene avviato sarà possibile utilizzare DressMeNow cercando prodotti, visualizzandoli, vederne le caratteristiche o le immagini o anche effettuando l'accesso per svolgere operazioni come acquisto di prodotti, provare prodotti sull'avatar, accedere alla ricerca tramite NLP ecc.

Avvio del sistema:

Identificativo Avvio_Sistema	Avvio del server	Data	28/11/2023
		Versione	1.0.00
		Autore	Tutti i membri del team
Descrizione	Verranno fornite le funzionalità per l'avvio del server da parte dell'amministratore del sistema (direttore)		
Attore Principale	Direttore - È interessato ad avviare il server		
Attori Secondari	N/A		
Entry condition	Il direttore è interessato ad avviare il server AND il sistema deve permetterglielo		
Exit condition On success	L'amministratore avvia il server correttamente		
Exit condition On failure	Il server non viene avviato		
Rilevanza/ User priority	Alta		
Frequenza stimata	1 volta/anno		
Extension Point	N/A		
Generalization of	N/A		
Flusso di eventi principale/ MAIN SCENARIO			
1	Direttore	Stabilisce la connessione col database e avvia uno script di configurazione del DB	
2	Sistema	Inizializza il DB	
3	Direttore	Deploya il file del sistema sul server	
4	Sistema	Se il server viene spento normalmente, avvia il sistema	

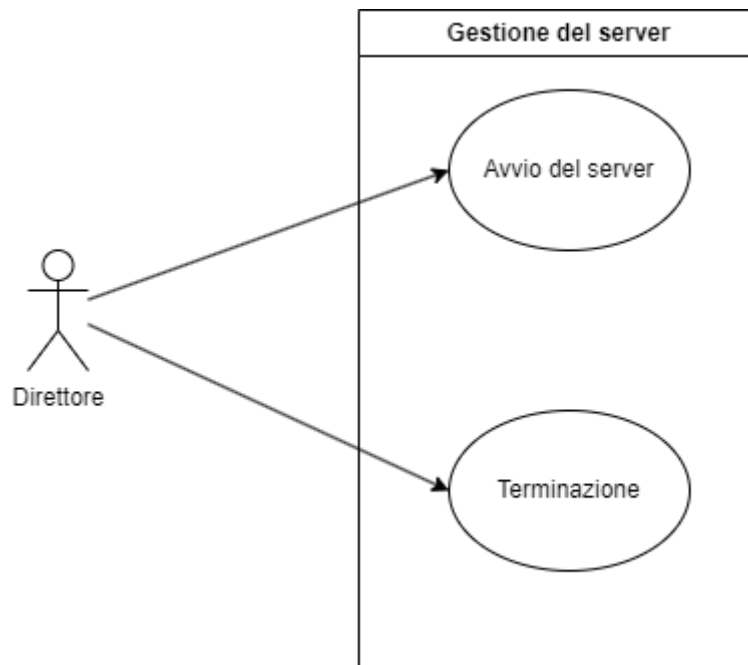
I Flusso di eventi alternativo: Il server avuto un arresto improvviso precedentemente		
4.1	Sistema	Se i server era stato arrestato a causa di un errore, viene notificato al direttore che c'è stato un errore.
I Flusso di eventi di Errore: Avvio non riuscito		
4.1	Sistema	Viene mostrato a schermo il messaggio di errore

Terminazione:

La terminazione del sistema avviene attraverso la funzionalità di logout. Per consentire che il server termini nel modo corretto, viene disposto un meccanismo secondo il quale, alla chiusura del server, nessun altro client ha possibilità di collegarsi ad esso.

Identificativo Termina_Sistema	Terminazione	Data	28/11/2023
		Versione	1.0.00
		Autore	Tutti i membri del team
Descrizione	Verranno fornite le funzionalità di terminazione del server da parte dell'amministratore del sistema (direttore)		
Attore Principale	Direttore - È interessato a terminare il server		
Attori Secondari	N/A		
Entry condition	L'amministratore è interessato a terminare il server AND il sistema deve permetterglielo		
Exit condition On success	L'amministratore termina il server correttamente AND il sistema farà in modo da non permettere agli utenti di collegarsi al sito mentre il server è termianto		
Exit condition On failure	Il server non termina		
Rilevanza/ User priority	Alta		
Frequenza stimata	1 volta/anno		
Extension Point	N/A		
Generalization of	N/A		
Flusso di eventi principale/ MAIN SCENARIO			
1	Direttore	Avvia uno script per terminare la connessione al database	

2	Sistema	Esegue la terminazione del server
I Flusso di eventi di Errore: Terminazione non riuscita		
2.1	Sistema	Non riesce a terminare il server e mostra un messaggio di errore

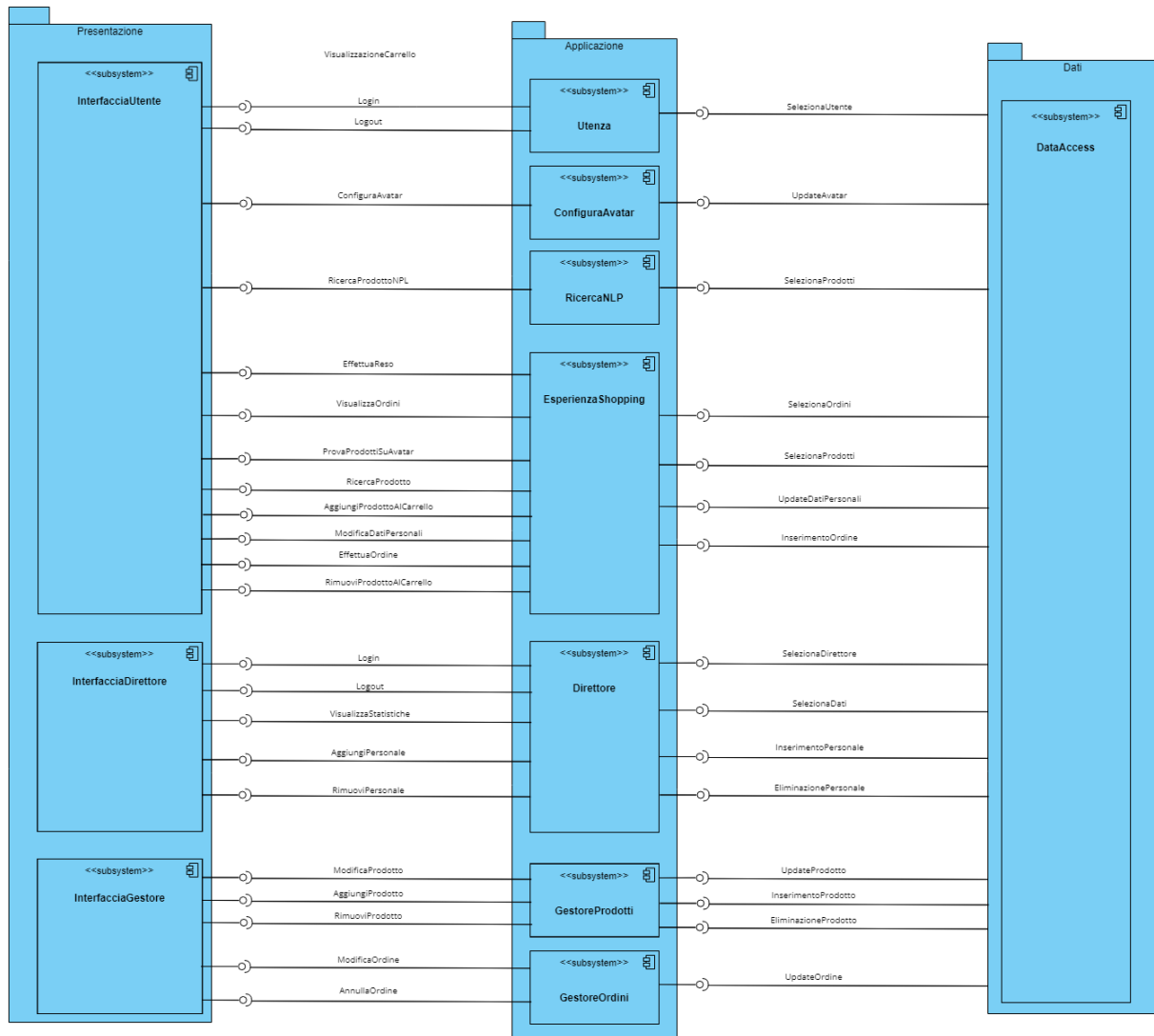


Fallimento:

Le possibili cose che potrebbero portare al fallimento del nostro sistema sono:

- Un sovraccarico del database potrebbe portare a una perdita di dati. Siccome nel nostro caso i dati presenti nel database possono essere dati sensibili (come nel caso dei dati personali dell'utente), il nostro sistema provvederà ad effettuare regolari backup, in modo da ripristinare i dati che potrebbero essersi persi.
- Un grave problema con una parte del sistema per il quale non abbiamo trovato alcuna soluzione.
- Terminazione inaspettata del server. Non prenderemo contromisure, l'unica operazione che ci faremo è la riattivazione dell'intero il sistema.

Servizi di Sottosistemi (Tutti i membri del team)



Class Diagram (Tutti i membri del team)

