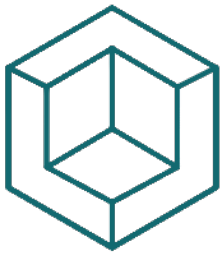
System Design Document



MUSEUM MANAGEMENT
SYSTEM

Riferimento	NC15_sdd_v1
Versione	1.0
Data	28/11/2024
Destinatario	Prof. Carmine Gravino
Presentato da	Team NC15
Approvato da	--

Sommario

Team Members

Nome	Cognome	Acronimo	Contatto
Salvatore	Nocera	SN	s.nocera7@studenti.unisa.it
Angelo	Barone	AB	a.barone69@studenti.unisa.it
Vincenzo	Bucciero	VB	v.bucciero2@studenti.unisa.it

Revision History

Data	Versione	Autori	Descrizione
28/11/2024	0.1	SN	Stesura capitolo introduttivo
28/11/2024	0.2	AB	Stesura Design Goals e Trade Off
29/11/2024	0.5	AB	Individuazione dell'Architettura Software Proposta con relativi diagrammi UML
7/12/2024	0.6	VB	Gestione dei dati persistenti
6/01/2025	0.7	AB	Stesura Sezione Accesso e Sicurezza
3/02/2025	1.0	SN – AB - VB	Revisione

1. Introduzione

1.1. Scopo del Sistema

Il progetto mira a creare un sistema software innovativo che rivoluzioni la gestione di un museo, combinando efficienza interna e un'esperienza di visita moderna e interattiva.

L'obiettivo è semplificare il lavoro quotidiano del personale, riducendo le attività manuali, e allo stesso tempo migliorare l'accessibilità e il coinvolgimento dei visitatori attraverso una piattaforma integrata.

Che cosa si vuole ottenere?

Da un lato, il sistema vuole ottimizzare la gestione interna del museo. Questo significa fornire al personale strumenti avanzati per organizzare e monitorare le collezioni, pianificare le attività e gestire risorse e orari in modo più efficace.

Dall'altro, punta a migliorare l'esperienza dei visitatori, offrendo loro la possibilità di accedere facilmente a informazioni dettagliate su mostre e opere, oltre a permettere di acquistare i biglietti online, evitando lunghe attese e gestendo meglio i flussi di pubblico.

Come funzionerà il sistema?

La piattaforma avrà due grandi aree di intervento:

1. Gestione interna:

   Il personale del museo potrà archiviare e catalogare digitalmente le opere d'arte, monitorare le risorse disponibili e organizzare turni e attività. Tutto sarà centralizzato e automatizzato, riducendo il lavoro manuale e migliorando la visibilità sulle operazioni quotidiane.

2. Esperienza visitatore:

   I visitatori avranno accesso a una piattaforma digitale con contenuti interattivi per esplorare le mostre. Inoltre, potranno acquistare i biglietti online in pochi clic, risparmiando tempo e migliorando l'organizzazione delle visite.

Cosa sarà incluso e cosa no?

Il sistema si concentrerà su aspetti essenziali della gestione museale, come il catalogo delle collezioni, la gestione del personale, la biglietteria online e le informazioni per i visitatori.

Tuttavia, non si occuperà di aspetti esterni al museo, come la logistica legata al trasporto delle opere, né della gestione di materiali non digitali.

Perché è importante?

L'idea nasce dalla necessità di modernizzare il modo in cui i musei operano, rendendo più semplice ed efficiente la gestione interna e più ricca e accessibile l'esperienza culturale per i visitatori.

In questo modo, il museo sarà non solo più organizzato, ma anche più attrattivo e fruibile per un pubblico sempre più ampio.

1.2. Obiettivi di Progettazione

Id	Descrizione	Categoria	Priorità	RNF di origine
DG_USA_1	Rendere l'interfaccia comprensibile e utilizzabile da almeno l'80% degli utenti in un test di usabilità.	Usabilità	6	RNF_USA_1
DG_USA_2	Fornire feedback visivo o testuale entro 1 secondo dall'azione utente per l'80% delle operazioni.	Usabilità	12	RNF_USA_3
DG_AF_1	Garantire un tasso di successo operativo del sistema pari al 99% durante un periodo di utilizzo simulato di 100 ore.	Affidabilità	8	RNF_AF_1
DG_AF_2	Generare notifiche chiare e risoluzioni guidate per il 100% degli errori critici entro 5 secondi dall'identificazione del problema, monitorato nei test.	Affidabilità	12	RNF_AF_2
DG_AF_3	Implementare una separazione delle operazioni per ruolo, verificata con un tasso di errore umano inferiore al 5% in test pratici con utenti rappresentativi.	Affidabilità	8	RNF_AF_3

DG_PR_1	Garantire che i tempi di risposta dell'interfaccia non superino i 2 secondi per il 95% delle operazioni in condizioni di carico normale.	Prestazioni	10	RNF_PR_1
DG_PR_2	Assicurare capacità di memorizzazione sufficiente per archiviare almeno 1 milione di record con un utilizzo medio del 75% della capacità disponibile.	Prestazioni	5	RNF_PR_2
DG_SEC_1	Implementare un sistema di autenticazione che blocchi almeno il 95% dei tentativi di accesso non autorizzato nei test di penetrazione.	Sicurezza	9	RNF_SEC_1
DG_MAN_1	Utilizzare standard di sviluppo che permettano una riduzione del tempo medio di manutenzione correttiva a meno di 2 ore.	Manutenibilità	4	RNF_MAN_1
DG_SCA_1	Garantire il supporto di almeno 100 utenti simultanei.	Scalabilità	15	RNF_SCA_1
DG_IM_1	Adottare un'architettura web-based per il sistema.	Implementazione	15	RNF_IM_1
DG_INT_1	Garantire la persistenza e coerenza dei dati con un tasso di errore inferiore allo 0,1% in test di integrità dei dati.	Interfacce	12	RNF_INT_1
DG_INT_2	Usare MySQL come database su server locale.	Interfacce	7	RNF_INT_3

DG_OP_1	Implementare strumenti di gestione per gli amministratori.	Operazioni	2	RNF_OP_1
DG_OP_2	Fornire supporto tecnico remoto con un tempo medio di risposta inferiore a 2 ore	Operazioni	1	RNF_OP_2

1.3. Trade-Off

Trade-Off	Descrizione
Tempo di risposta vs Sicurezza	L'implementazione di autenticazioni sicure (es. 2FA o crittografia avanzata) può aumentare i tempi di risposta durante l'accesso degli utenti.
Costi di sviluppo vs Usabilità	Garantire interfacce intuitive e funzionalità user-friendly richiede maggiori investimenti in design, test di usabilità e iterazioni, aumentando il budget di sviluppo.
Costi di sviluppo vs Portabilità	Sviluppare per compatibilità multipiattaforma (Web, Android, iOS) richiede framework avanzati, aumentando i costi iniziali, ma migliorando la scalabilità e la flessibilità.
Costi di sviluppo vs Scalabilità	Implementare una struttura scalabile per supportare più utenti simultanei richiede un'infrastruttura inizialmente più costosa, che potrebbe essere sovradimensionata in fase iniziale.
Sicurezza vs Usabilità	Misure di sicurezza avanzate (es. timeout delle sessioni, complessità delle password) possono ridurre l'immediatezza d'uso e la facilità di accesso per gli utenti meno esperti.
Manutenibilità vs Prestazioni	Scrivere codice estremamente modulare e conforme agli standard per migliorare la manutenibilità può introdurre una lieve perdita di efficienza nelle operazioni eseguite dal sistema.

1.4. Definizioni, Acronimi e Abbreviazioni

Definizioni: Glossario (?)

Acronimi:

- **MMS**: Museum Management System
- **RAD**: Requirement Analysis Document;
- **SDD**: System Design Document;
- **DG**: Design Goal;
- **RNF**: Requisito Non Funzionale;
- **USA**: Usabilità;
- **AF**: Affidabilità;
- **PR**: Prestazioni;
- **SEC**: Sicurezza
- **MAN**: Manutenibilità
- **SCA**: Scalabilità
- **IM**: Implementazione
- **INT**: Interfacce
- **OP**: Operazioni
- **PAK**: Packaging
- **LAW**: Legali
- **VIS**: Visitatore
- **US**: Utente Registrato
- **PER**: Personale del museo
- **AD**: Admin;
- **MVC**: Model View Controller;
- **HTTP**: HyperText Transfer Protocol;
- **DBMS**: DataBase Management System;
- **JSP**: Java Server Page
- **SC**: Scenario
- **SU**: StartUp
- **SD**: Shutdown
- **FA**: Failure
- **UC**: Use Case

1.5. Riferimenti

- Documento di Statement of Work relativo a questo progetto.
- Documento di Requirement Analysis Document relativo a questo progetto.

1.6. Organizzazione del Documento

Il presente documento è organizzato in quattro/cinque sezioni:

- **Introduzione**: Viene descritto in generale lo scopo del sistema, gli obiettivi di design che il sistema propone di raggiungere;
- **Architettura Sistema Corrente**: Descrive, nel caso esista, lo stato attuale e le funzionalità offerte dal sistema corrente;
- **Architettura Sistema Proposto**: Viene presentata l'architettura del sistema proposto, in cui sarà documentata la decomposizione in sottosistemi, il mapping hardware/software, la gestione dei dati persistenti, il controllo degli accessi e sicurezza, il controllo del flusso globale del sistema, le condizioni limite;
- **Servizi dei Sottosistemi**: Vengono presentati i servizi dei sottosistemi;
- **Glossario**: Raccolta di vocaboli meno comuni utilizzati nella stesura del documento.

2. Architettura software corrente

2.1. Panoramica

Al momento non esiste un software che integri tutte le funzionalità offerte da Museum Management System in un'unica piattaforma. Le soluzioni alternative risultano frammentate e poco organiche, rendendo impossibile un confronto diretto e completo con l'architettura del sistema proposto.

3. Architettura software proposta

3.1. Panoramica

La piattaforma MMS è un'applicazione software progettata per interagire con gli utenti attraverso un'interfaccia web e gestire la persistenza dei dati utilizzando un database relazionale.

Il sistema adotta un'architettura **Three-Tier**, suddividendo l'applicazione in tre livelli principali:

- **Interface Layer**: include gli oggetti Boundary, responsabili della visualizzazione dei dati e della gestione dell'interazione con l'utente.
- **Application Logic Layer**: comprende gli oggetti Control, incaricati di ricevere i comandi dall'utente, applicare la logica del sistema e coordinare il flusso di dati tra la View e il Model.
- **Storage Layer**: gestisce la logica di business relativa alla memorizzazione, al recupero e alla manipolazione dei dati persistenti, fornendo metodi sicuri ed efficienti per l'accesso al database.

Questa scelta ci permette una migliore separazione dei concetti e organizzazione del codice, che comporta la semplificazione dei processi di **modifica** e **manutenzione** del codice. Inoltre, avremo importanti vantaggi in termini di:

- **Scalabilità & Riusabilità**
- **Flessibilità & Testabilità**

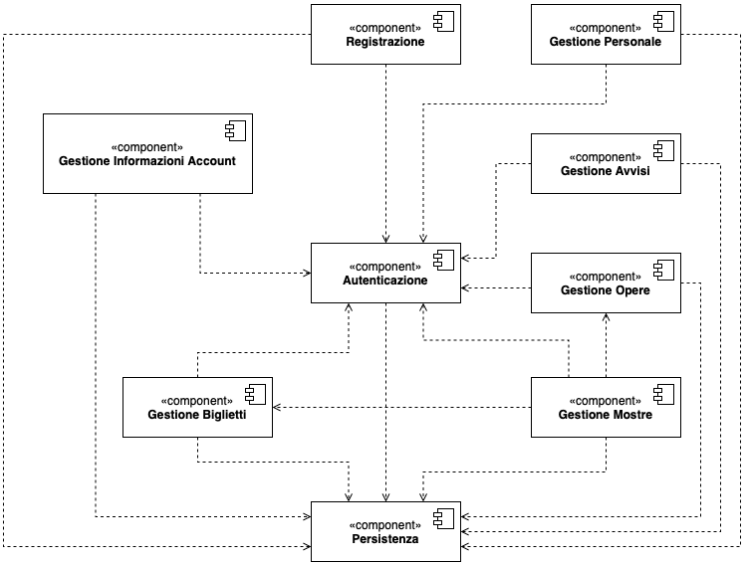
3.2. Decomposizione in sottosistemi

I sottosistemi individuati sono:

- **Sottosistema "Autenticazione"**: si occupa dell'autenticazione di utenti e personale già registrati.
  - Login
  - Logout
- **Sottosistema "Registrazione"**: si occupa della registrazione di nuovi utenti.
  - Creazione nuovo account
- **Sottosistema "Gestione Opere":** supporta il personale del museo nella gestione delle opere
  - Visualizzazione opere.
  - Aggiunta di opere.
  - Modifica opere.
  - Eliminazione opere.
  - Valutazione delle opere.
- **Sottosistema "Gestione Mostre":** consente la pianificazione di eventi e mostre.
  - Visualizzazione degli eventi programmati.
  - Creazione, modifica ed eliminazione di mostre.
- **Sottosistema "Gestione Account"**
  - Visualizzazione area utente.
  - Modifica dati account.
  - Eliminazione account.
- **Sottosistema "Gestione Personale"**: gestisce il personale del museo.
  - Creazione di account del personale.
  - Assegnazione di ruoli e turni al personale.
- **Sottosistema "Gestione Avvisi"**: gestisce la pubblicazione e visualizzazione di avvisi:
  - Creazione, modifica ed eliminazione di avvisi.
  - Visualizzazione avvisi.
- **Sottosistema "Gestione Biglietti"**: gestisce le operazioni di acquisto biglietti per gli utenti.
  - Creazione di Biglietti.
  - Acquisto di Biglietti.
  - Visualizzazione Biglietti.
- **Sottosistema di Persistenza dei Dati:** garantisce la gestione sicura ed efficiente dei dati del sistema.
  - Gestione della persistenza e coerenza dei dati in un database relazionale (MySQL).
  - Accesso ai dati da parte degli altri sottosistemi.

3.2.1. Diagramma di Decomposizione

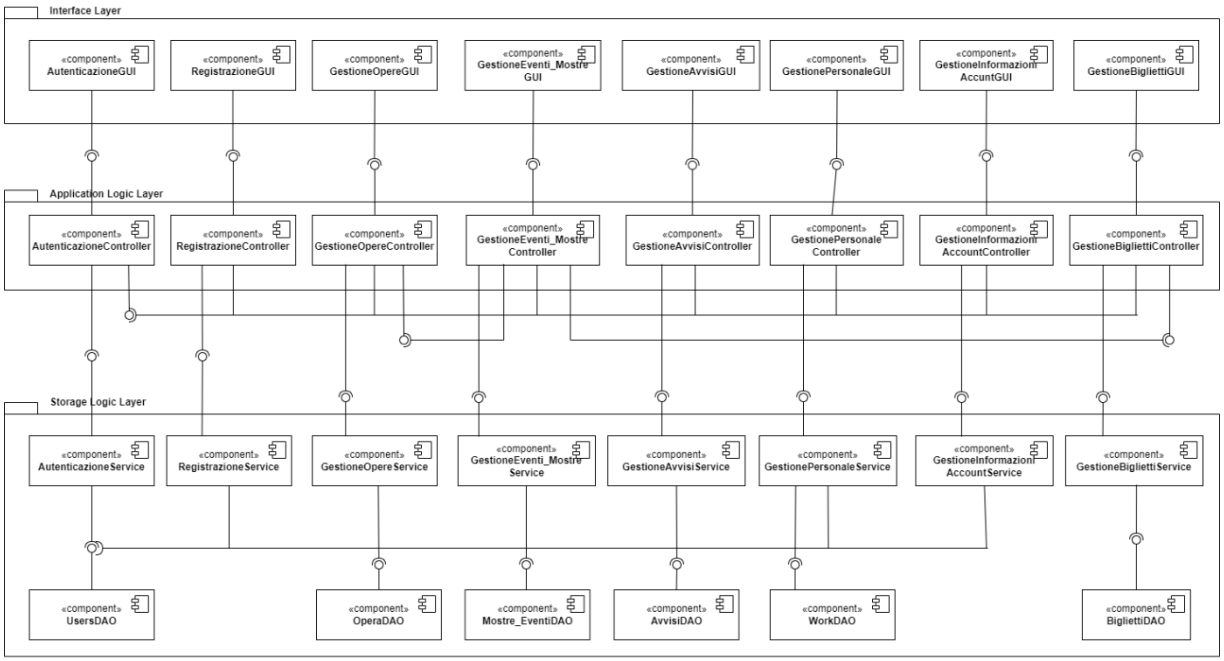
Sono mostrate di seguito le dipendenze tra i sottosistemi attraverso un component diagram UML:



3.2.2. Diagramma Architetturale

Di seguito una vista dettagliata di ciascun sottosistema evidenziando le componenti principale:

- **Interface Layer**: contiene le varie view che mostreranno al cliente le pagine web.
- **Application Logic Layer**: contiene la logica che controlla il sistema
- **Storage Logic Layer**: contiene la logica di business e i DAO che forniscono accesso ai dati.

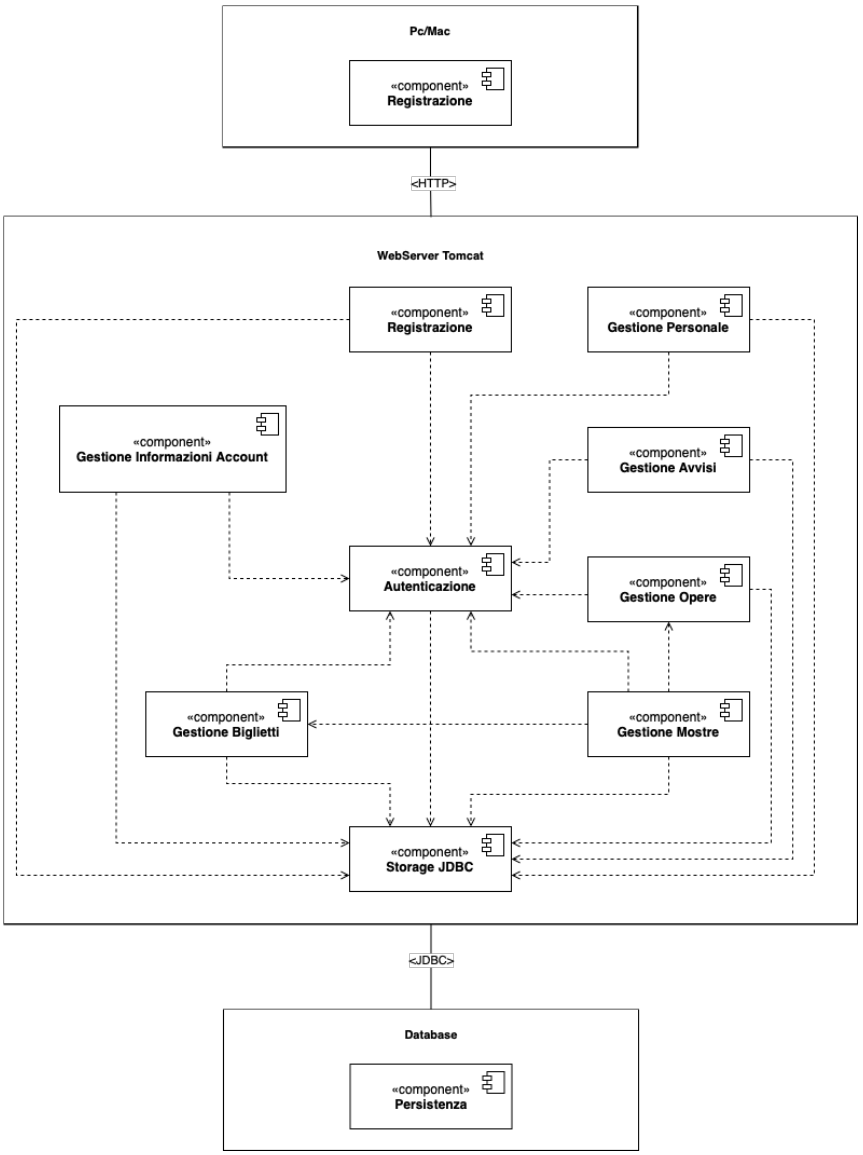


3.3. Hardware/software mapping

L'applicazione web in sviluppo sarà ospitata su una piattaforma hardware composta da un server centrale, progettato per gestire le richieste dei client provenienti da qualsiasi dispositivo dotato di browser web e connessione Internet.

Il sistema proposto utilizza un web server dedicato e adotta un'architettura centralizzata, in cui tutte le operazioni sono gestite su un singolo nodo. Questa configurazione semplifica la gestione e il monitoraggio del sistema, concentrando i servizi in un unico punto di controllo.

Di seguito è presentato un diagramma UML che illustra la mappatura tra hardware e software.



3.4. Gestione dati persistenti

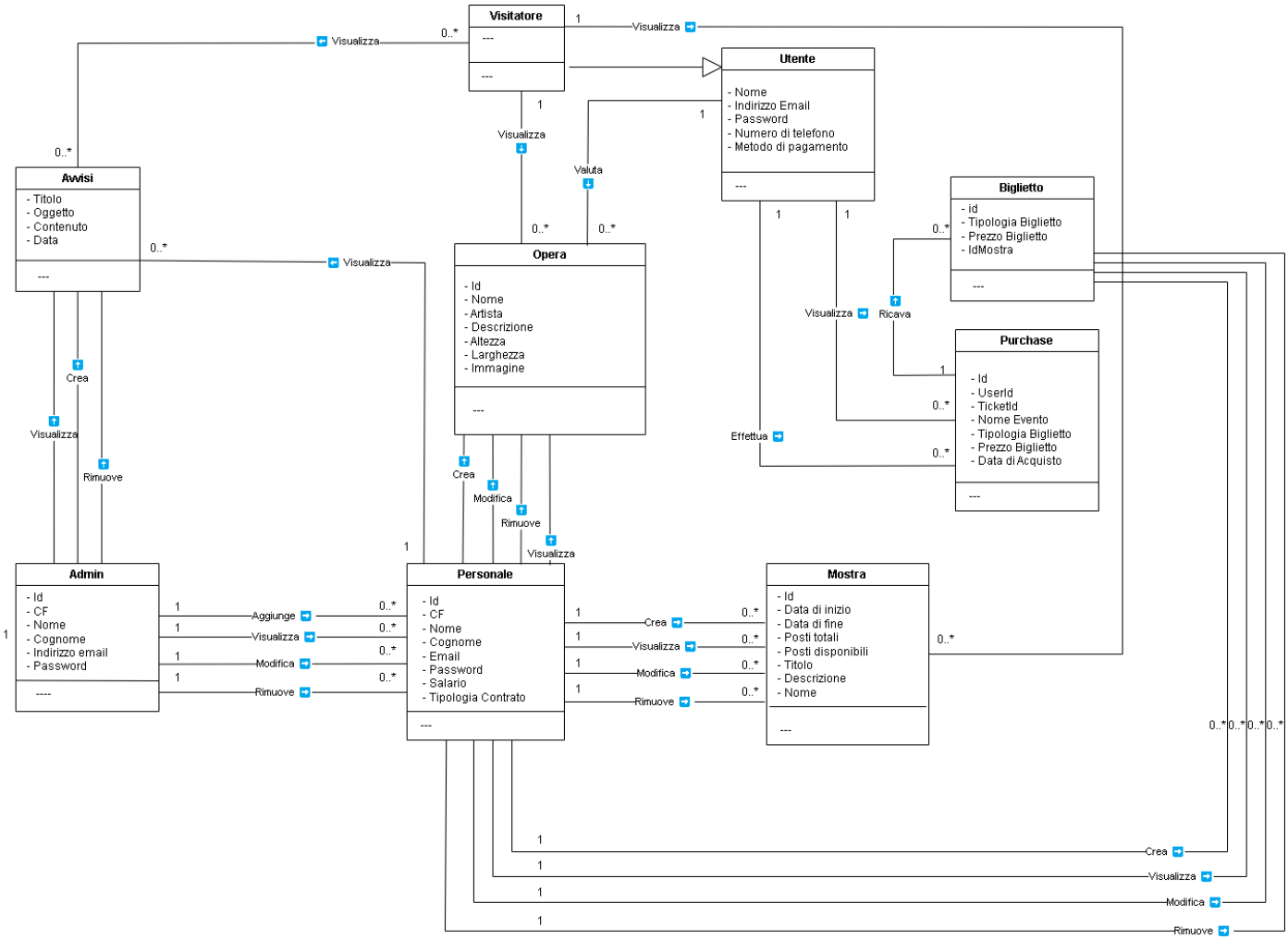
Scelta di utilizzo di un database relazionale. Per garantire un accesso concorrente e per poter garantire consistenza e astrazione nella gestione dei dati persistenti tramite utilizzo di DBMS.

L'utilizzo di DBMS permette:

- 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
- Privatezza dei dati, garantita dal fatto che un DBMS permette un accesso protetto ai dati (esempio: tramite utilizzo di Prepared Statement). Utenti diversi possono avere accesso a diverse porzioni della base di dati e possono essere abilitati a diverse operazioni su di esse.
- 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.

3.4.1. CD_SDD: Entity Class Diagram ristrutturato

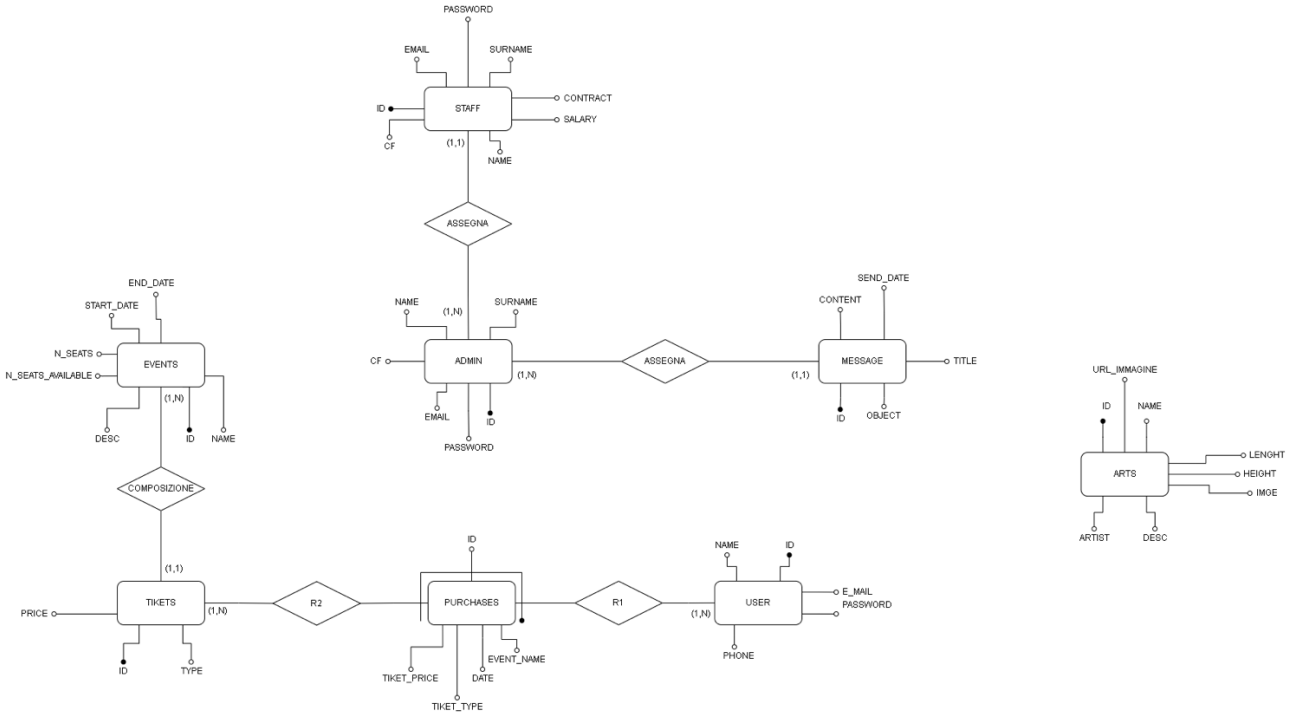
L'entity class diagram è stato ristrutturato a causa della non implementazione di requisiti funzionali di bassa priorità (Gestione dei turni) e la modifica nella gestione dell'acquisto e della visualizzazione dei biglietti per soddisfare i requisiti non funzionali.



3.4.2. ER: Schema ER del database

Nell' implementazione del database seguendo i vincoli specificati dal cliente e per il rispetto dei requisiti non funzionali di ottimizzazione dello spazio non sarà necessario memorizzare in modo persistente i membri dello staff che svolgeranno operazioni riguardanti tikets e arts.

Inoltre, essendo presente un unico admin, e non contemplando il cambiamento di questo anche le operazioni effettuate dall'admin potranno non essere associate allo stesso, andando a risparmiare sia spazio per la memorizzazione delle associazioni, sia tempo per effettuare join per ricavarne dati.



3.4.3. Dizionario dei dati

Nome Entità	User		
Descrizione	Contiene dati relativi ad un utente registrato		
Nome Campo	Tipo	Vincolo chiave	Altri vincoli
User_id	int	Primary Kay	Auto increment
User_name	varchar(50)		Not null
User_password	varchar(255)		Not null
User_email	varchar(100)		Not null Unique

Nome Entità	Admin		
Descrizione	Contiene dati relativi all'admin		
Nome Campo	Tipo	Vincolo chiave	Altri vincoli
Admin_id	int	Primary Kay	Auto increment
Admin_cf	varchar(16)		Not null Unique
Admin_name	varchar(50)		Not null
Admin_surname	varchar(50)		Nut null
Admin_password	varchar(255)		Not null Unique
Admin_email	varchar(100)		Not null Unique

Nome Entità	Staff		
Descrizione	Contiene dati relativi allo staff		
Nome Campo	Tipo	Vincolo chiave	Altri vincoli
Staff_id	int	Primary Kay	Auto increment
Staff_cf	varchar(16)		Not null Unique
Staff_name	varchar(50)		Not null
Staff_surname	varchar(50)		Nut null
Staff_password	varchar(255)		Not null Unique
Staff_email	varchar(100)		Not null Unique
Staff_salary	Decimal(10,2)		Not null
Type_contract	Enum('Full-time' , 'Part-time' , 'Stage')		Not null

Nome Entità	Message		
Descrizione	Contiene dati relativi ad un messaggio rilasciato dall'admin		
Nome Campo	Tipo	Vincolo chiave	Altri vincoli
Message_id	int	Primary Kay	Auto increment
Message_title	varchar(50)		Not null
Message_object	varchar(50)		Not null
Message_content	varchar(50)		Not null
Send_date	date		Not null

Nome Entità	Arts		
Descrizione	Contiene dati relativi alle opere aggiunte dallo staff		
Nome Campo	Tipo	Vincolo chiave	Altri vincoli
Art_id	int	Primary Kay	Auto increment
Art_name	varchar(50)		Not null
Art_desc	varchar(200)		Not null
Art_artist	varchar(50)		Not null
Art_length	varchar(50)		Not null
Art_height	varchar(50)		Not null
Art_immage	varchar(100)		Not null

Nome Entità	Events		
Descrizione	Contiene dati relativi agli eventi aggiunti dallo staff		
Nome Campo	Tipo	Vincolo chiave	Altri vincoli
Event_id	int	Primary Kay	Auto increment
Start_date	Datetime		Not null
End_date	Datetime		Not null
N_seats	int		Not null
N_seats_available	int		Default 0
Event_desc	varchar(1000)		Not null
Event_name	varchar(20)		Not null

Nome Entità	Tickets		
Descrizione	Contiene dati relativi agi tikets aggiunti dallo staff		
Nome Campo	Tipo	Vincolo chiave	Altri vincoli
Ticket_id	int	Primary Kay	Auto increment
Ticket_type	Varchar(50)		Not null
Ticket_price	Decimal(10,2)		Not null
Event_id	int	Foreign key (Events)(event_id)	Not null

Nome Entità	Purchases		
Descrizione	Contiene dati relativi agli acquisti di biglietti da parte degli utenti		
Nome Campo	Tipo	Vincolo chiave	Altri vincoli
Purchase_id	int	Primary Kay	Auto increment
Purchase_date	Datetime		Not null
Event_name	Varchar(20)	Foreign key (event)(event_name)	Not null
Ticket_type	Varchar(50)	Foreign key (ticket)(ticket_type)	Not null
Ticket_price	Decimal(10,2)	Foreign key (ticket)(ticket_price)	Not null
User_id	int	Foreign key (user)(user_id)	Not null
User_name	Varchar(50)	Foreign key (user)(user_name)	Not nul

3.5. Controllo accessi e sicurezza

Per tenere traccia di quali attori possono accedere ai servizi offerti dal sistema di seguito viene mostrata la matrice degli accessi.

Attori / Oggetti	Visitatore	Utente	Personale	Admin
Registrazione	- Registrazione			
Autenticazione		- Login - Logout	- Login - Logout	- Login - Logout
Gestione Account		- Visualizzazione Area Utente - Modifica Dati Account - Eliminazione Account		- Visualizzazione Utenti - Eliminazione Utenti
Gestione Personale			- Visualizzazione Area Personale	- Visualizzazione Personale - Aggiunta Personale - Rimozione Personale - Modifica Personale - Modifica Turni
Gestione Opere		- Visualizzazione Opere - Valutazione Opere	- Visualizzazione Opere - Aggiunta Opere - Eliminazione Opere - Modifica Opere	
Gestione Avvisi	- Visualizzazione Avvisi	- Visualizzazione Avvisi	- Visualizzazione Avvisi	- Visualizzazione Avvisi - Creazione Avvisi - Rimozione Avvisi
Gestione Mostre	- Visualizzazione Mostre	- Visualizzazione Mostre	- Visualizzazione Mostre - Creazione Mostre - Eliminazione Mostre - Modifica Mostre	
Gestione Biglietti		- Visualizzazione Storico Biglietti - Acquisto Biglietti		

3.6. Controllo globale del software

Il sistema per la gestione museale è un sistema interattivo progettato per consentire agli utenti di accedere e utilizzare le funzionalità tramite un'interfaccia grafica user-friendly.

Il controllo globale è basato su un modello **event-driven** che coordina il flusso delle operazioni, questo approccio garantisce un'interazione fluida e asincrona, particolarmente efficace per una web-application come quella proposta.

3.7. Boundary conditions

Nella seguente sezione vengono descritti i comportamenti del sistema in situazioni estreme, inaspettate o al di fuori delle normali operazioni.

- ***Start-Up***

IDENTIFICATIVO	UC_SU	DATA	3/12/24
NOME	StartUpServer	VERSIONE	0.1
		AUTORE	AB
DESCRIZIONE	L'UC descrive la funzionalità di avvio del Server.		
ATTORE PRINCIPALE	ADMIN: Vuole avviare il server per rendere disponibile il sistema.		
ATTORI SECONDARI	NA		
ENTRY CONDITION	Il Server è pronto all'avvio.		
EXIT CONDITION ON SUCCESS	Il server è avviato e il sistema rende disponibili le funzionalità agli utenti.		
EXIT CONTITION ON FAILURE	Il server non è avviato.		
RILEVANZA/USER PRIORITY	Elevata		
FREQUENZA STIMATA	1/mese		
EXTENSION POINT	NA		

GENERALIZATION OF		NA
FLUSSO DI EVENTI PRINCIPALE		
1	Admin	Utilizza l'apposito comando per avviare il server
2	Sistema	Inizializza una connessione con il database.
3	Sistema	Verifica l'integrità dei dati persistenti salvati nel database.
4	Sistema	Avvia il server e lo notifica all'amministratore.
FLUSSO DI EVENTI DI ERRORE: Connessione assente		
2.a1	Sistema	Il sistema mostra un pop-up di errore che comunica all'amministratore che non è stato possibile avviare il Server a causa della mancanza di connessione.
FLUSSO DI EVENTI DI ERRORE: Dati non validi		
3.b1	Sistema	Il sistema mostra un pop-up di errore che segnala errori sull'integrità dei dati persistenti nel database e non procede all'avvio del server.
3.b2	Admin	Accede al database e corregge gli errori sui dati persistenti.
3.b3	Admin	Riesegue il primo step.

- ***Shut-Down***

IDENTIFICATIVO	UC_SD	DATA	3/12/24
NOME	Arresto del Server	VERSIONE	0.1
		AUTORE	AB
DESCRIZIONE	L'UC descrive la funzionalità di Arresto del Server.		
ATTORE PRINCIPALE	ADMIN: Vuole arrestare il server per spegnere il sistema.		

ATTORI SECONDARI		NA
ENTRY CONDITION		Il Server è avviato.
EXIT CONDITION ON SUCCESS		Il Server viene arrestato.
EXIT CONTITION ON FAILURE		Il server non viene arrestato.
RILEVANZA/USER PRIORITY		Elevata
FREQUENZA STIMATA		1/mese
EXTENSION POINT		NA
GENERALIZATION OF		NA
FLUSSO DI EVENTI PRINCIPALE		
1	Admin	Utilizza l'apposito comando per arrestare il Server
2	Sistema	Effettua il salvataggio dei dati persistenti.
3	Sistema	Chiude le connessioni attive ed arresta il server.
FLUSSO DI EVENTI DI ERRORE: Dati non salvati		
2.a1	Sistema	Il sistema mostra un pop-up di errore che comunica all'amministratore che il salvataggio dei dati persistenti non è andato a buon fine, quindi non procede all'arresto.

- ***Failure***

IDENTIFICATIVO	UC_FA	DATA	3/12/24
NOME	Fallimento del sistema	VERSIONE	0.1

		AUTORE	AB
DESCRIZIONE	L'UC descrive il comportamento del sistema al verificarsi di un fallimento del sistema.		
ATTORE PRINCIPALE	ADMIN		
ATTORI SECONDARI	NA		
ENTRY CONDITION	Il Sistema viene terminato inaspettatamente.		
EXIT CONDITION ON SUCCESS	Il Sistema viene riavviato correttamente.		
EXIT CONTITION ON FAILURE	Il Sistema non viene riavviato.		
FLUSSO DI EVENTI PRINCIPALE			
1	Admin	**Include US_SU**	

- ***Errore di accesso ai Dati Persistenti***

IDENTIFICATIVO	UC_FA	DATA	3/12/24
NOME	Errore di accesso ai dati persistenti.	VERSIONE	0.1
		AUTORE	AB
DESCRIZIONE	L'UC descrive il comportamento del sistema al verificarsi di un errore di accesso ai dati persistenti.		
ATTORE PRINCIPALE	ADMIN: Vuole risolvere la condizione di fallimento.		
ATTORI SECONDARI	NA		
ENTRY CONDITION	Il Server riscontra un errore nell'accesso ai dati persistenti o essi risultano corrotti.		
EXIT CONDITION ON SUCCESS	Il Server riprende il corretto funzionamento.		
EXIT CONTITION ON FAILURE	Il Server non riprende il corretto funzionamento.		
FLUSSO DI EVENTI PRINCIPALE			

1	Sistema	Notifica l'amministratore dell'impossibilità di accedere ai dati persistenti.
2	Admin	Arresta forzatamente tramite l'apposito comando.
3	Admin	Risolve l'errore sui dati persistenti
4	Admin	Riavvia il Sistema (**include UC_SU**)

4. Servizi dei sottosistemi

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

4.1 Autenticazione

SERVIZIO	DESCRIZIONE	INTERFACCIA
Login utente	Servizio per autenticare un visitatore sia esso Utente, Staff o Admin tramite credenziali.	AutenticazioneService
Logout utente	Servizio per terminare una sessione attiva.	AutenticazioneService

4.2 Registrazione

SERVIZIO	DESCRIZIONE	INTERFACCIA
Registrazione utente	Servizio per creare un nuovo utente con informazioni personali, email e password.	RegistrazioneService

4.3 Gestione Opere

SERVIZIO	DESCRIZIONE	INTERFACCIA
Visualizzazione Opere	Servizio per recuperare e visualizzare l'elenco delle opere con dettagli.	GestioneOpereService
Aggiunta Opere	Servizio per aggiungere nuove opere al database con metadati.	GestioneOpereService
Modifica opere	Servizio per aggiornare le informazioni di un'opera esistente.	GestioneOpereService
Eliminazione opere	Servizio per rimuovere un'opera dal database.	GestioneOpereService
Valutazione delle opere	Servizio per registrare e calcolare le valutazioni delle opere.	GestioneOpereService

4.4 Gestione Mostre

SERVIZIO	DESCRIZIONE	INTERFACCIA
Visualizzazione Mostre programmate	Servizio per elencare tutte mostre attive e future con dettagli.	GestioneMostreService
Creazione Mostre	Servizio per pianificare una nuova mostra inserendo i dettagli.	GestioneMostreService
Modifica Mostre	Servizio per aggiornare i dettagli di una mostra esistente.	GestioneMostreService

Eliminazione Mostre	Servizio per rimuovere una mostra pianificata.	GestioneMostreService

4.5 Gestione Informazioni Account

SERVIZIO	DESCRIZIONE	INTERFACCIA
Visualizzazione Area Utente	Servizio per recuperare e mostrare i dati personali dell'utente.	GestioneAccountService
Modifica dati Account	Servizio per aggiornare le informazioni personali dell'utente.	GestioneAccountService
Eliminazione account	Servizio per cancellare l'account dell'utente e relativi dati personali.	GestioneAccountService

4.6 Gestione Personale

SERVIZIO	DESCRIZIONE	INTERFACCIA
Creazione di account del personale	Servizio per creare account per il personale museale con ruolo e credenziali.	GestionePersonaleService
Assegnazione di turni	Servizio per assegnare ruoli e pianificare turni di lavoro.	GestionePersonaleService
Modifica account	Servizio per modificare le informazioni inerenti a un membro del personale.	GestionePersonaleService
Eliminazione account	Servizio per cancellare l'account dell'operatore e relativi dati personali.	GestionePersonaleService

Visualizzazione Homepage personale	Servizio che fornisce tutte le informazioni necessarie a un membro dello staff.	GestionePersonaleService

4.7 Gestione Avvisi

SERVIZIO	DESCRIZIONE	INTERFACCIA
Creazione di avvisi	Servizio per creare nuovi avvisi indirizzati a visitatori o personale.	GestioneAvvisiService
Modifica avvisi	Servizio per aggiornare i dettagli di un avviso esistente.	GestioneAvvisiService
Eliminazione avvisi	Servizio per rimuovere avvisi scaduti o non più rilevanti.	GestioneAvvisiService
Visualizzazione avvisi	Servizio per elencare tutti gli avvisi attivi.	GestioneAvvisiService

4.8 Gestione Biglietti

SERVIZIO	DESCRIZIONE	INTERFACCIA
Creazione di biglietti	Servizio per generare biglietti di Ingresso.	GestioneBigliettiService
Acquisto di biglietti	Servizio per consentire agli utenti di acquistare biglietti.	GestioneBigliettiService
Visualizzazione biglietti	Servizio per elencare i biglietti acquistati dall'utente con dettagli.	GestioneBigliettiService

5. Design Patterns

In questa sezione vengono descritti i design pattern adottati ed i vantaggi offerti.

5.1 Facade

Il design pattern facade è un design pattern strutturale che offre un'interfaccia semplificata per accedere a un insieme di oggetti all'interno di un sottosistema complesso.

La sua implementazione prevede la definizione di un'interfaccia che funge da punto d'accesso unico per i servizi del sottosistema, mascherando dettagli implementativi.

Questo approccio permette di semplificare l'interazione con il sistema, riducendo la complessità per chi lo utilizza. I principali vantaggi offerti dell'utilizzo di tale design pattern includono:

- Semplificazione dell'accesso ai servizi: l'interfaccia di alto livello facilita l'interazione con il sistema, nascondendo i dettagli implementativi dei vari servizi.
- Riduzione dell'accoppiamento: l'astrazione favorisce una comunicazione più flessibile tra i componenti del sistema, riducendo l'accoppiamento.
- Facilitazione dei test: L'isolamento delle componenti agevola il testing.

5.2 Data Access Object (DAO)

Il pattern design Data Access Object (DAO) consente di separare la logica di business dalla gestione della persistenza dei dati, riducendo l'accoppiamento tra i diversi livelli dell'applicazione.

Tramite un'interfaccia DAO, viene astratto l'accesso ai dati, nascondendo tutti i dettagli relativi alla loro gestione. Questo approccio introduce un livello dedicato esclusivamente alla persistenza, migliorando la modularità del sistema. I principali vantaggi offerti sono:

- Separazione delle responsabilità: Il livello di persistenza è isolato, favorendo una gestione centralizzata dei dati ed evitando le dipendenze dalle specifiche del database
- Flessibilità e scalabilità: le componenti non dipendono dall'implementazione specifica del DAO, facilitando eventuali migrazioni verso nuovi database
- Manutenibilità: la gestione dei dati diventa più efficiente e semplifica operazioni di manutenzione
- Leggibilità del codice: l'architettura è più chiara e modulare grazie al basso accoppiamento tra livelli

La struttura ottenuta combinando questi design pattern contribuisce a rendere l'applicazione più stabile, manutenibile e facilmente estendibile, oltre che semplificare le fasi di testing