

System Design Document

EasyCoach

Cronologia delle revisioni

Data	Versione	Descrizione	Autori
22/11/2024	0.1	Scopo del sistema, Design goal e Trade off	NV, SF
23/11/2024	0.2	Revisione Design goal e Trade off	NV, FD, SF, RF
25/11/2024	0.3	Architettura del sistema attuale e architettura del sistema proposto	RF, FD
12/01/2025	0.4	Decomposizione in sottosistemi e Component Diagram	SF, NV
14/01/2025	0.5	Definizione dei Design pattern: Observer Pattern, Object Pool Pattern, DAO	NV, FR

		Pattern	
15/01/2025	0.6	Mapping Hardware e SoftWare, Deployment Diagram	FD, SF
17/01/2025	0.7	Gestione dei dati persistenti	FD
17/01/2025	0.8	Diagramma ER e dizionario dei dati	SF, FD
18/01/2025	0.9	Matrice degli accessi	NV
20/01/2025	0.10	Controllo globale del software e condizioni limite	SF, RF
21/01/2025	0.11	Servizi dei sottoinsiemi	FD, SF
22/01/2025	0.12	Glossario e indice	SF, NV
23/01/2025	1.0	Revisione finale	NV, SF, FD, RF

Membri del Team

Nome	Ruolo del progetto	Acronimo	Informazioni di Contratto
Nello Valentino	Membro del team	NV	n.valentino6@studenti.unisa.it
Simone Fausto	Membro del team	SF	s.fausto5@studenti.unisa.it
Francesco D'Arco	Membro del team	FD	f.darco6@studenti.unisa.it
Roberto Fiorenza	Membro del team	RF	r.fiorenza3@studenti.unisa.it

	1
Membri del Team	2
1. Introduzione	3
1.1 Scopo del sistema	3
1.2 Design Goals	3
1.3 Definizioni, acronimi e abbreviazioni	6
1.4 Riferimenti	7
2. Architettura del sistema corrente	7
3. Architettura del sistema proposto	8
3.1 Panoramica	8
3.2 Decomposizione in Sottosistemi	8
3.3 Mapping Hardware/Software	12
3.4 Gestione dei dati persistenti	13
3.5 Matrice degli accessi (Tabella di accesso globale)	19
3.6 Controllo globale del Software	21
3.7 Condizioni di limite	22
4. Servizi dei Sottosistemi	27
5. Glossario	30

1. Introduzione

1.1 Scopo del sistema

Il sistema che si vuole realizzare, mira a sviluppare una piattaforma che consenta a mentor e mentee di **interagire, connettersi, organizzare** sessioni e **collaborare** nella maniera più efficiente possibile.

In definitiva, la piattaforma si propone di rendere il mentoring **accessibile, sicuro e organizzato**, ottimizzando ogni aspetto dell'interazione tra mentor e mentee. Attraverso l'integrazione di funzionalità moderne e un design intuitivo, la piattaforma faciliterà il raggiungimento di obiettivi personali e professionali, creando un ambiente collaborativo e stimolante per entrambe le parti.

1.2 Design Goals

Nella presente sezione si andranno a presentare i **Design Goals**, ovvero le qualità sulle quali il sistema deve essere focalizzato. I design goal sono stati suddivisi nelle seguenti categorie:

- **Criteri di usabilità:** Garantire un'interfaccia intuitiva e facilmente utilizzabile
- **Criteri di affidabilità:** Progettare un sistema robusto che garantisca robustezza e protezione
- **Criteri di performance:** Ottimizzare il sistema per garantire tempi di risposta rapidi e basse latenze
- **Criteri di supportabilità:** Facilitare l'estensibilità e la manutenibilità del sistema

Ciascun design goal è descritto da:

- **Rank:** che ne specifica un valore di priorità compreso tra 1 e 16 (1 massima e 16 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:** ovvero il requisito non funzionale che lo ha generato

Rank	ID Design goal	Descrizione	Categoria	RNF di origine
8	DG_1	Progettare un sistema performante con tempi di risposta inferiori a 5 secondi	Performance	RNF_P_1
3	DG_2	Validare i dati inseriti dagli utenti lato server e client per garantirne consistenza e integrità	Affidabilità	RNF_A_2
6	DG_3	L'interfaccia sarà ottimizzata per adattarsi a diversi dispositivi (PC, tablet, smartphone) per grandezza da 480px a 1920px	Usabilità	RNF_U_01
3	DG_4	Assicurare la persistenza dei dati in modo tale che siano sempre recuperabili	Affidabilità	RNF_A_1
5	DG_5	Saranno adottati protocolli di crittografia per	Affidabilità	RNF_A_3

		proteggere le informazioni sensibili, come le password degli utenti		
6	DG_6	Progettare un sistema facilmente estensibile per l'aggiunta di nuove funzionalità	Supportabilità	RNF_S_1
7	DG_7	Fornire un Feedback visivo chiaro ed immediato	Usabilità	RNF_U_5
8	DG_8	Ogni pagina presenterà una barra di navigazione per facilitare l'accesso rapido alle diverse sezioni	Usabilità	RNF_U_4

Trade off

Trade off	Descrizione
Supportabilità vs Performance	La suddivisione del sistema in moduli distinti, ognuno con responsabilità ben definite, favorisce un basso accoppiamento e una chiara separazione tra le diverse funzionalità, migliorando la Supportabilità e facilitando eventuali modifiche o estensioni (DG_6). Tuttavia, questa struttura introduce un maggiore overhead nella comunicazione tra i moduli, aumentando il numero di iterazioni necessarie per completare un'operazione. Ciò potrebbe influire sui tempi di risposta , incidendo sulle prestazioni complessive

	del sistema (DG_1).
Persistenza Dati vs Performance	E' necessario che tutti i dati siano persistenti nel database (DG_4) in modo tale che le informazioni non vengano perse. Tuttavia, un elevato volume di letture e scritture, specialmente per operazioni critiche come le prenotazioni, può impattare le prestazioni e il tempo di risposta del sistema (DG_1). L'utilizzo di query indicizzate potrebbe ridurre questo problema.
Feedback esplicito vs Persistenza Dati	Fornire un feedback visivo immediato (DG_7), come notifiche in tempo reale, richiede l'elaborazione immediata delle informazioni e l'invio tempestivo agli utenti. Tuttavia ciò implica che anche queste informazioni non cruciali per il sistema debbano essere salvate all'interno del database (DG_4). Come soluzione si può adottare una cache temporanea che conservi le notifiche prima che queste siano inviate

1.3 Definizioni, acronimi e abbreviazioni

Di seguito sono elencate alcuni termini quali verranno riscontrati all'interno del documento:

- **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
- **Jitsi Meet:** sistema per la creazione e la gestione di videoconferenze
- **Dati persistenti:** dati non volatili, che vengono salvati all'interno di un database
- **SDD:** System Design Document
- **RAD:** Requirements Analysis Document
- **ODD:** Object Design Document

1.4 Riferimenti

Di seguito è elencata una lista dei documenti utili durante la lettura

- [Statement Of Work](#)
- [System Design Document](#)
- [Object Design Document](#)
- [Test Plan](#)
- [Manuale Utente](#)

2. Architettura del sistema corrente

Dopo un'attenta ricerca sul Web, abbiamo rilevato l'assenza di siti che offrono le stesse funzionalità di EasyCoach. Relativamente parlando, famosi siti come **Fiverr**, sono meno specifici e si concentrano sulla ricerca di un lavoro a tempo determinato che si scinde dalla visione di servizi la quale EasyCoach vuole fornire. La nostra piattaforma, mira ad un'interazione più amichevole, che si allontana dall'ambito lavorativo, per soddisfare delle esigenze, che siano temporanee o che possano fruttare nel tempo

3. Architettura del sistema proposto

3.1 Panoramica

Il sistema da noi proposto, sarà una piattaforma Web Based, che segue lo stile architetturale **Three-Tier** e l'utilizzo di **design pattern** appositi. Il motivo di tale scelta, si basa principalmente sull'esperienza lavorativa pregressa su questa architettura da parte degli sviluppatori, oltre che un'attenta suddivisione dei layer, che si occuperanno di aspetti e funzionalità differenti.

Il sistema adotta un'**architettura chiusa**, infatti ogni livello può comunicare solo con lo stato immediatamente inferiore. I vantaggi sono:

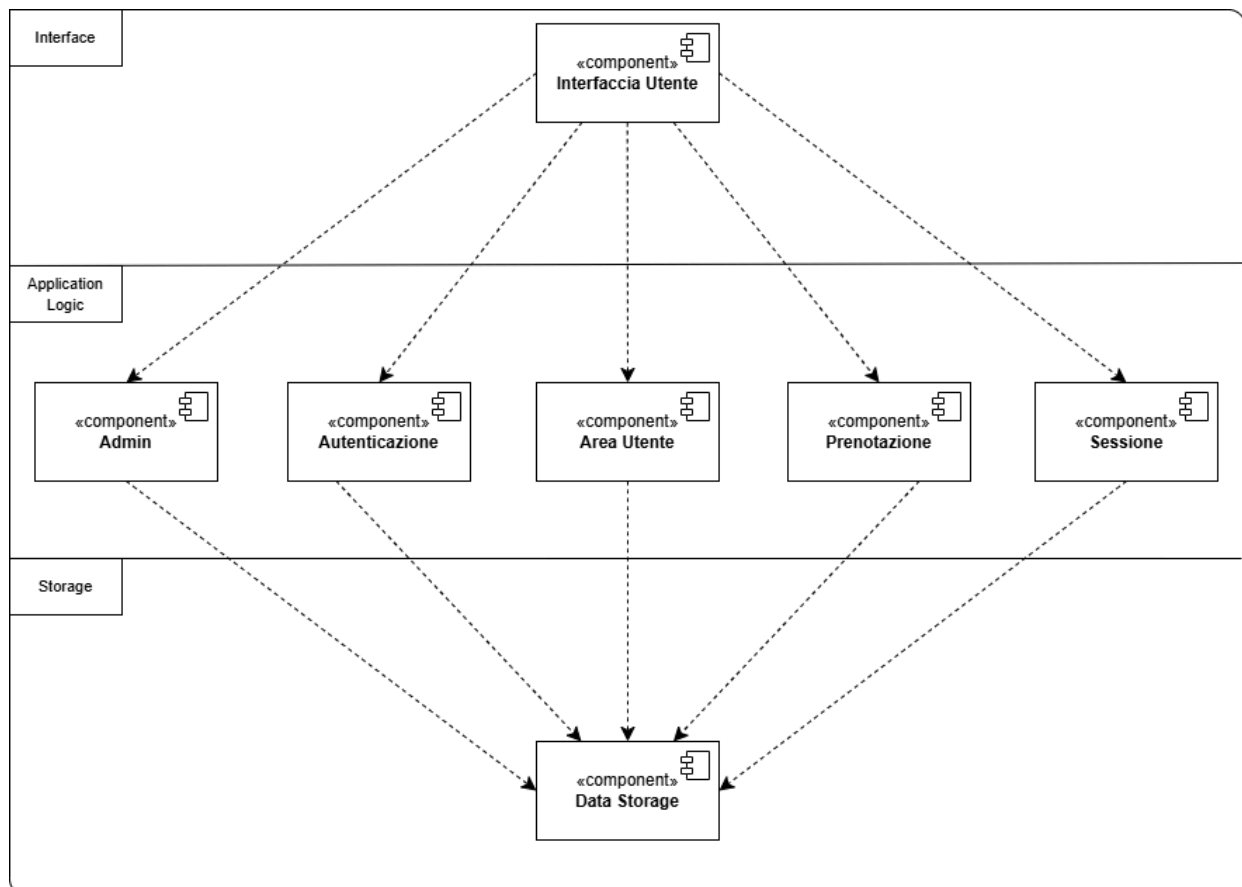
- **Manutenibilità:** le modifiche a uno strato non influenzano direttamente gli altri, rendendo più semplice la gestione del codice
- **Flessibilità:** la separazione dei sottosistemi consente di aggiornare o sostituire componenti senza compromettere l'intero sistema.
- **Modularità:** ogni livello è separato e si occupa di un aspetto specifico del sistema.

3.2 Decomposizione in Sottosistemi

Il sistema è suddiviso in più sottosistemi, ciascuno con responsabilità specifiche.

L'architettura **Three-Tier** di EasyCoach è strutturata come segue:

Component Diagram



Interface Layer

L'Interfaccia Utente è il sottosistema responsabile dell'**interazione con l'utente**. Fornisce le classi e le operazioni necessarie per acquisire input, visualizzare output e interagire con i servizi della piattaforma. Si occupa inoltre di:

- **Coordinare** la logica applicativa, delegando le operazioni principali ai sottosistemi del livello sottostante (Application Logic Layer).
- **Validare i dati in ingresso**, assicurando la conformità alle specifiche del sistema e restituendo eventuali messaggi di errore in caso di input errati o di dati omessi

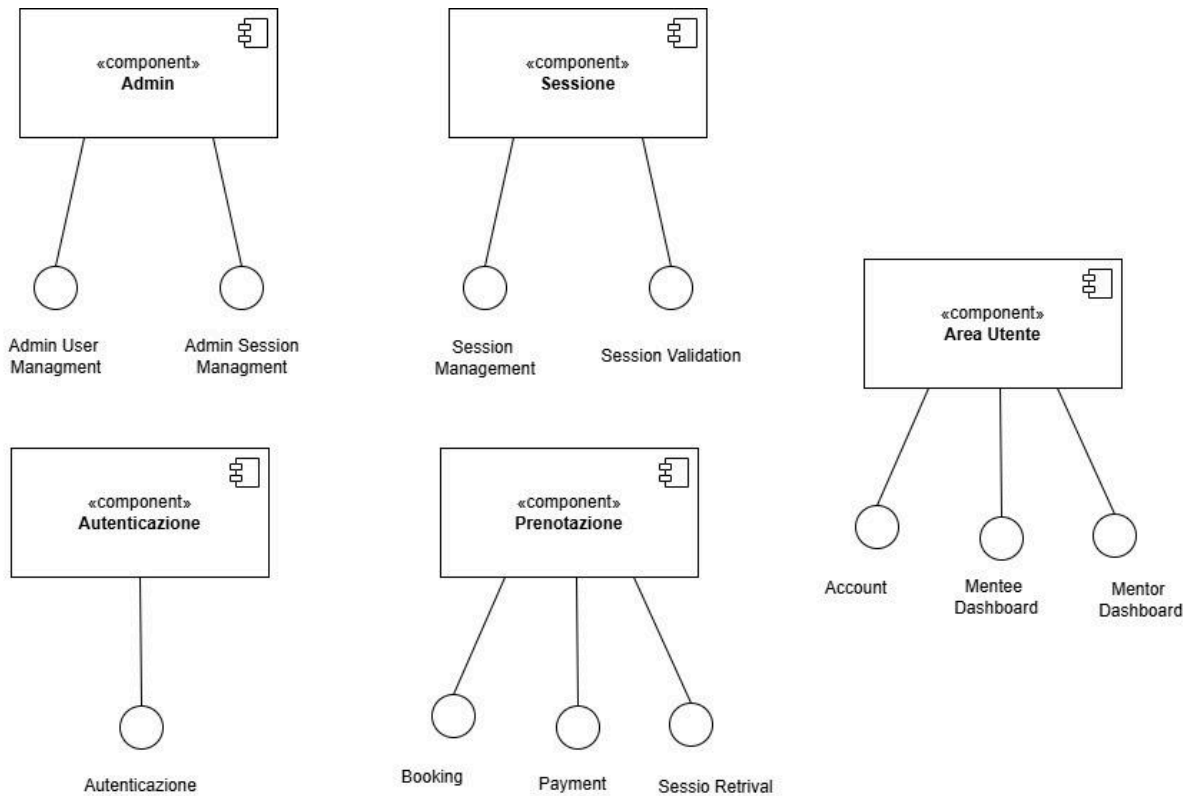
Application Logic Layer

La Logica Applicativa è suddivisa in cinque sottosistemi principali:

- **Admin:** si occupa di visualizzare tutti gli utenti e le sessioni presenti sulla piattaforma. Garantisce il mantenimento del sistema eliminando utenti o sessioni non conformi o non necessarie
- **Area utente:** contiene le dashboard degli utenti loggati (mentor e mentee) e le loro funzionalità. Include operazioni relative all'account come modifica password, elimina account, visualizza prenotazioni attive etc...
- **Autenticazione:** permette l'autenticazione di un ospite al sito, in modo tale che possa accedere alla piattaforma
- **Prenotazione:** gestisce la ricerca, la prenotazione e l'acquisto di una sessione.
- **Sessione:** consente la creazione, la modifica e la cancellazione di una sessione

Storage Layer

Rappresenta il sottosistema responsabile della memorizzazione e gestione dei dati persistenti, garantendo **integrità**, **sicurezza** e **accessibilità** delle informazioni. Questo livello gestisce tutte le operazioni di lettura e scrittura sui dati, assicurando che le informazioni relative agli utenti, alle sessioni, alle prenotazioni e ai pagamenti siano sempre disponibili e consistenti



3.2.2 Utilizzo dei Design Pattern nei sottosistemi

I design pattern che abbiamo deciso di utilizzare sono:

- Observer Pattern:** viene adottato con l'obiettivo di gestire la comunicazione in tempo reale tra gli utenti, in particolare per l'invio di notifiche tra mentor e mentee. La sua implementazione prevede la registrazione degli utenti come osservatori, i quali vengono automaticamente notificati quando si verifica un acquisto di una sessione. Questo meccanismo consente un aggiornamento immediato dell'interfaccia utente senza richiedere ricaricamenti della pagina o polling continuo al server.
- Object Pool Pattern:** utilizzato per ottimizzare la gestione delle connessioni al database, con l'obiettivo di ridurre il tempo di creazione e rilascio delle connessioni, migliorando le prestazioni del sistema. L'implementazione prevede un pool di connessioni riutilizzabili che vengono allocate alle richieste degli utenti e rilasciate quando non più necessarie, evitando così il sovraccarico dovuto alla creazione dinamica di nuove connessioni per ogni operazione. Questo approccio riduce la latenza e migliora l'efficienza nell'accesso ai dati.

- **DAO Pattern:** adottato con l'obiettivo di separare la logica di business dall'accesso ai dati, garantendo modularità e manutenibilità del codice. L'implementazione prevede la creazione di classi specifiche per la gestione delle operazioni CRUD sulle entità principali del database. Ogni classe DAO interagisce con il database fornendo un'interfaccia strutturata per il recupero e la manipolazione dei dati senza esporre direttamente la logica di persistenza al resto dell'applicazione.

3.3 Mapping Hardware/Software

EasyCoach è una piattaforma **web-based distribuita**, progettata per garantire scalabilità e accessibilità da qualsiasi dispositivo con connessione a Internet. L'architettura è suddivisa in tre componenti principali:

- **Client Machine:** esegue il **sottosistema Interface** tramite un browser web, gestendo l'interazione con l'utente.
- **Application Server:** esegue il **sottosistema Application Logic**, elaborando le richieste degli utenti, gestendo la logica di business e i controlli di sicurezza.
- **Database Server:** esegue il **sottosistema Storage**, garantendo la persistenza dei dati e la gestione delle transazioni.

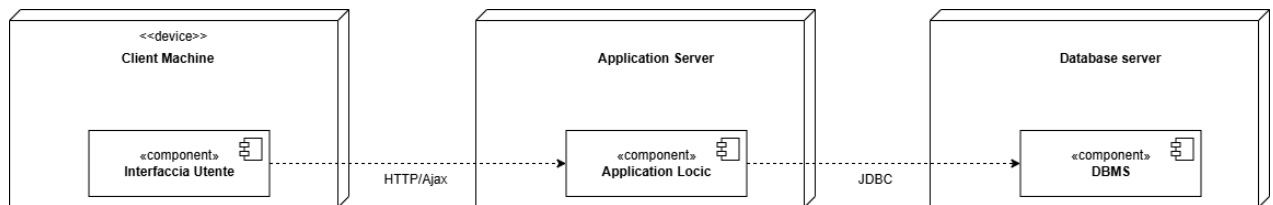
Il sistema sarà implementato utilizzando **Apache Tomcat 10.1.34** in combinazione con **Java JDK 21.0.5**. Tomcat fungerà da contenitore di Servlet, permettendo il deployment e l'esecuzione dell'applicazione web. La logica applicativa sarà sviluppata in **Java**, sfruttando la sua modularità e affidabilità.

Tecnologie Utilizzate:

- **Frontend:**
 - **HTML5, CSS3, Bootstrap 5:** creazione di un'interfaccia utente moderna, responsive e compatibile con diversi dispositivi.
 - **JavaScript:** gestione dell'interattività, validazione dinamica dei dati e miglioramento dell'esperienza utente.

- **AJAX** : utilizzato per richieste asincrone, riducendo i tempi di caricamento delle pagine senza necessità di ricaricare l'interfaccia.
- **Backend:**
 - **Jakarta Servlet API**: gestione delle richieste HTTP tra frontend e backend.
 - **JDBC (Java Database Connectivity)**: gestione della comunicazione con il database MySQL.

Deployment Diagram

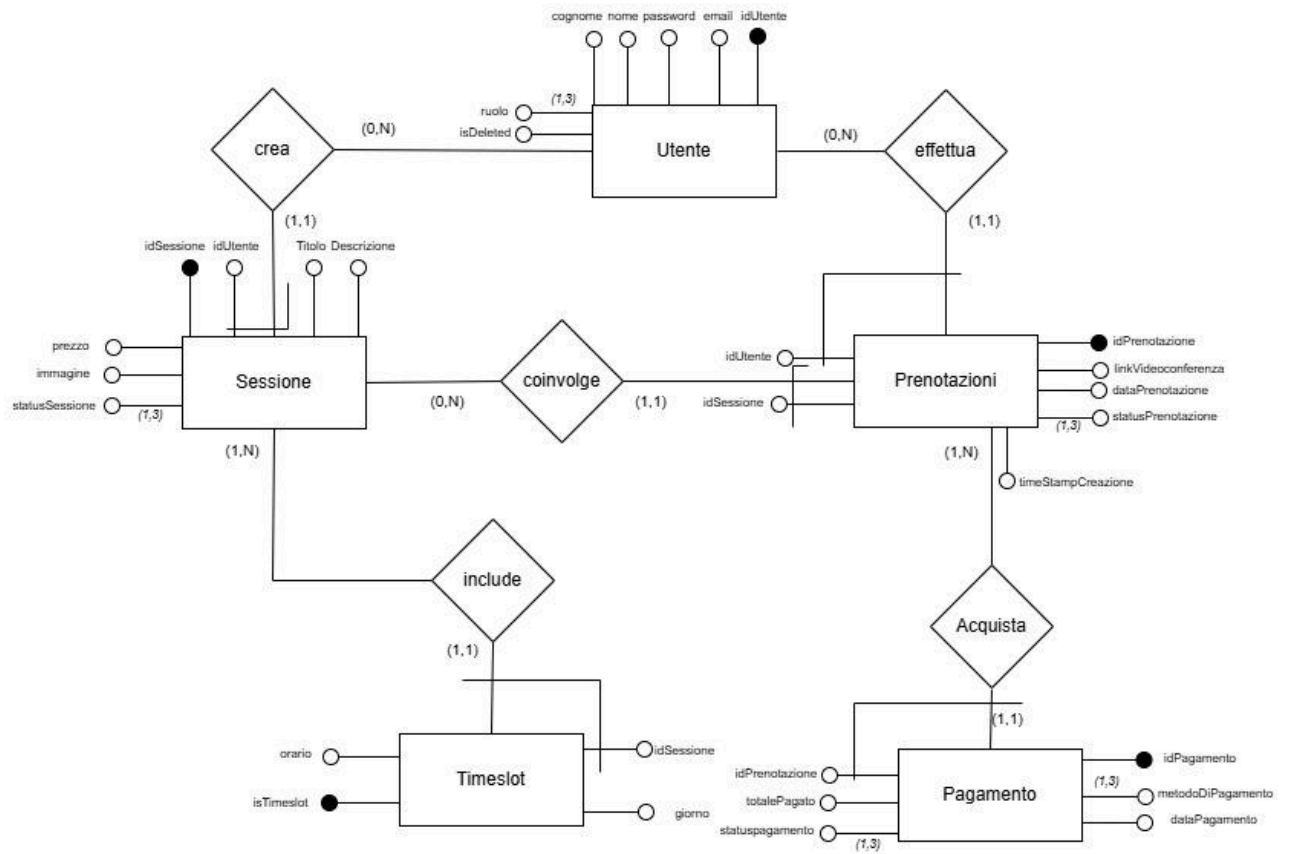


3.4 Gestione dei dati persistenti

Per la gestione dei dati persistenti del sistema EasyCoach, abbiamo scelto di utilizzare **MySQL** come database relazionale. L'adozione di un **DBMS relazionale** consente una gestione efficiente e strutturata delle informazioni, garantendo sicurezza, integrità e prestazioni ottimali.

- **Integrità dei dati:** MySQL consente l'implementazione di **vincoli di integrità referenziale**, garantendo la consistenza e correttezza delle informazioni, specialmente nella gestione delle prenotazioni e delle sessioni
- **Sicurezza e privacy:** il database supporta la crittografia per proteggere password e dati sensibili degli utenti. Le password sono memorizzate utilizzando **SHA-256**.
- **Performance e affidabilità:** supporta query ottimizzate per la ricerca, filtraggio e aggiornamento dei dati con un **basso tempo di risposta**. L'uso di **indexing** migliora le prestazioni nei casi di alto traffico.
- **Scalabilità:** MySQL permette di gestire una grande quantità di dati, con possibilità di replica e sharding per migliorare la distribuzione del carico in caso di espansione del sistema.

Diagramma ER



Dizionario dei dati

Di seguito si mostrano gli attributi per ogni entità individuata

Nome Entità	Timeslot		
Descrizione	Contiene i dati relativi ad un Timeslot		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
idTimeslot	int	PRIMARY KEY	NOT NULL
idSessione	int	FOREIGN KEY	NOT NULL
giorno	int		NOT NULL
orario	int		NOT NULL

Nome Entità	Pagamento		
Descrizione	Contiene i dati relativi ad un pagamento		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
idPagamento	int	PRIMARY KEY	NOT NULL AUTO_INCREMENT
idPrenotazione	int	FOREIGN KEY	NOT NULL
metodoPagamento	enum('Carta','Pay Pal','GooglePay')		DEFAULT 'CARTA'
totalePagato	decimal(10,2)		NOT NULL
statusPagamento	enum('In Attesa','Completato','Rifiutato')		DEFAULT 'IN ATTESA'

dataPagamento	datetime		CURRENT_TIMESTAMP
----------------------	----------	--	-------------------

Nome Entità	Prenotazione		
Descrizione	Contiene i dati relativi ad una prenotazione		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
idPrenotazione	int	PRIMARY KEY	NOT NULL, AUTO_INCREMENT
idUtente	int	FOREIGN KEY	NOT NULL
idTimeslot	int	FOREIGN KEY	NOT NULL
linkVideoconferenza	varchar(225)		DEFAULT NULL
statusPrenotazione	enum('In Attesa','Attiva','Conclusa','Annullata')		DEFAULT 'IN ATTESA'
dataPrenotazione	date		NOT NULL
idSessione	int	FOREIGN KEY	NOT NULL
timestampCreazione	timestamp		DEFAULT CURRENT_TIMESTAMP

Nome Entità	Sessione		
Descrizione	Contiene i dati relativi alla sessione		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
idSessione	int	PRIMARY KEY	NOT NULL, AUTO_INCREMENT
idUtente	int	FOREIGN KEY	NOT NULL
titolo	varchar(225)		NOT NULL
descrizione	varchar(1000)		NOT NULL
prezzo	double		NOT NULL
immagine	varchar(225)		NOT NULL
statusSessione	enum('Attiva','Non disponibile','Archiv iata')		DEFAULT 'ATTIVA'

Nome Entità	Utente		
Descrizione	Contiene i dati relativi all'utente		
Nome campo	Tipo	Vincolo di chiave	Altri vincoli
idUtente	int	PRIMARY KEY	NOT NULL, AUTO_INCREMENT
email	varchar(225)		NOT NULL
nome	varchar(225)		NOT NULL
cognome	varchar(225)		NOT NULL
password	varchar(225)		NOT NULL

ruolo	enum('Admin','Mentor','Mentee')		NOT NULL
isDeleted	tinyint(1)		NOT NULL DEFAULT '0'

3.5 Matrice degli accessi (Tabella di accesso globale)

Di seguito viene mostrata una **tabella di accesso globale**, dove ogni riga della matrice contiene una tupla. Se la tupla è presente per un determinato sottosistema e operazioni, l'accesso è consentito altrimenti no.

Attore	Sottosistema/Classe	Servizio	Descrizione
Ospite	Autenticazione	Registrazione	Possibilità di registrarsi al sistema
Amministratore	Admin	Gestione degli utenti	Può visualizzare o eliminare gli utenti presenti sulla piattaforma
	Autenticazione	Login	Possibilità di autenticarsi nel sistema
	Admin	Gestione delle sessioni	Può visualizzare o eliminare le sessioni presenti sulla piattaforma
Mentor	Area Utente	Accesso Area Utente	Può visualizzare la propria area utente
	Autenticazione	Login	Può accedere alla sua pagina
	Autenticazione	Logout	Può disconnettersi dalla

			piattaforma
	Sessione	Creazione Sessione	Può creare una nuova sessione
	Sessione	Modifica Sessione	Può modificare una sua sessione
	Sessione	Elimina Sessione	Può eliminare una sua sessione
	Area Utente	Visualizza Prenotazioni attive	Può visualizzare le proprie prenotazioni attive
	Area Utente	Visualizza Sessioni	Può visualizzare le proprie sessioni
	Prenotazione	Ricerca	Può cercare una sessione nel sistema
	Prenotazione	Visualizza Sessione Cercata	Può visualizzare i dettagli di una sessione ricercata
Mentee	Area Utente	Accesso Area Utente	Può visualizzare la propria area utente
	Autenticazione	Login	Può accedere alla sua pagina
	Autenticazione	Logout	Può disconnettersi dalla piattaforma
	Prenotazione	Ricerca	Può cercare una sessione nel sistema
	Prenotazione	Visualizza Sessione Cercata	Può visualizzare i dettagli di una sessione ricercata
	Prenotazione	Pagamento	Può acquistare una

			sessione
	Area Utente	Visualizza Prenotazioni	Visualizza le proprie prenotazioni attive e concluse

3.6 Controllo globale del Software

Il sistema EasyCoach adotta un flusso di controllo **event-driven** e segue un'architettura **three-tier** per garantire modularità, separazione delle responsabilità e una gestione ottimale delle risorse. Il controllo è centralizzato, con un chiaro punto di coordinamento che gestisce il flusso di richieste tra i vari livelli. Questa scelta progettuale consente di migliorare la manutenibilità, la scalabilità e la sicurezza complessiva del sistema.

EasyCoach segue un flusso di controllo strutturato in quattro fasi principali:

1. **Ricezione della richiesta:** Il Web Server attende richieste dal browser dell'utente e le inoltra alla logica applicativa.
2. **Elaborazione della richiesta:** Il sistema interpreta la richiesta, esegue la logica applicativa e interagisce con il livello di accesso ai dati per recuperare o aggiornare le informazioni necessarie.
3. **Generazione della risposta:** Dopo aver elaborato la richiesta, i risultati vengono restituiti al Web Server.
4. **Risposta al client:** Il Web Server invia la risposta al browser dell'utente, completando il ciclo.

3.7 Condizioni di limite

Il paragrafo seguente rappresen gli **Boundary Use Case**, ovvero le condizioni limite del sistema che includono lo startup, shutdown, fallimento del sistema ed errore all'accesso ai dati persistenti

Avvio Del Sistema

Identificativo UCBC_1	Avvio del Sistema	Data	20/01/2025
		Versione	1
		Autori	SF
Descrizione	Lo Uc permette l’avvio del sistema		
Attore principale	Amministratore		
Attore secondari	NA		
Entry Condition	L’amministratore accede alla macchina su cui risiede il sistema		
Exit Condition On success	Il sistema viene avviato correttamente		
Exit Condition On failed	Il sistema non viene avviato		
Flusso di eventi principale			
1	Amministratore	Esegue sulla macchina il comando che avvia il sistema	
2	Sistema	Il sistema controlla la disponibilità delle risorse principali: Avvio del server Apache Tomcat, Connessione al database MySQL, File di configurazione, Servizi esterni	
3	Sistema	Il sistema avvia i Job predefiniti dello Scheduler, il quale procede	

		<p>modificare lo stato delle Prenotazioni nel seguente modo:</p> <ul style="list-style-type: none"> ● CONCLUSA: se la data e l'ora della prenotazione sono precedenti a quella attuale ● ANNULLATA: se una prenotazione si trova nello stato di IN_ATTESA
I Flusso di Eventi Alternativo: I dati persistenti nel database sono danneggiati o non coerenti		
2.a1	Sistema	invia una notifica all'amministratore con il messaggio di errore relativo all'integrità dei dati, specificando quali tabelle o dati sono coinvolti
2.a2	Amministratore	accede al sistema di gestione del database
2.a3	Amministratore	esegue operazioni di ripristino del database
2.a4	Amministratore	avvia nuovamente il sistema per verificare che i dati siano ora correttamente recuperati e che l'integrità sia ristabilita
I Flusso di Eventi Alternativo: Lo scheduler non avvia i Job correttamente		
2.b1	Sistema	Rileva un errore nell'avvio dei job dello Scheduler e invia una notifica all'amministratore
2.b2	Amministratore	Consulta i log di errore per identificare la causa del problema ed esegue un tentativo di riavvio manuale dello Scheduler
2.b3	Sistema	Riprova ad avviare il processo e verifica se l'operazione ha avuto successo

Spegnimento Del Sistema

Identificativo UCBC_2	Spegnimento del sistema	Data	20/01/2025
		Versione	1
		Autori	RF
Descrizione	Lo UC permette lo spegnimento del sistema		
Attore principale	Amministratore		
Attore 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 failed	Il sistema non viene spento		
Flusso di eventi principale			
1	Amministratore	L’amministratore invia un segnale di spegnimento al sistema	
2	Sistema	Chiude le connessioni attive e completa le transazioni in corso, garantendo l’integrità dei dati. Arresta la JVM e si disconnette dal database MYSQL	
I Flusso di Eventi Alternativo: Ci sono connessioni ancora aperte			
2.b1	Sistema	Recide le connessioni verso l’esterno	
2.b2	Sistema	Notifica l’Amministratore dell’avvenuto spegnimento del sistema e il numero di connessioni recise	

Fallimento Del Sistema

Identificativo UCBC_1	Fallimento del sistema	Data	20/01/2025
		Versione	1
		Autori	SF
Descrizione	L'UC definisce il comportamento del Sistema in caso di fallimento		
Attore principale	Amministratore		
Attore secondari	NA		
Entry Condition	Il Sistema viene terminato inaspettatamente		
Exit Condition On success	Il Sistema viene riavviato correttamente		
Exit Condition On failed	Il Sistema non viene riavviato		
Flusso di eventi principale			
1	Amministratore	Include UCBC_1	

4. Servizi dei Sottosistemi

Considerando i sottosistemi del livello Application Logic: Admin, Autenticazione, Area Utente, Sessione, Prenotazione si elencano i servizi che offrono

Sottosistema Admin

Servizio	Descrizione	Interfaccia
Visualizza Utente	Permette di visualizzare gli utenti presenti sulla piattaforma	AdminUserManagementService
Elimina Utente	Permette di eliminare gli utenti presenti sulla piattaforma	AdminUserManagementService
Visualizza Sessioni	Permette di visualizzare le sessioni presenti sulla piattaforma	AdminSessionManagementService
Elimina Sessioni	Permette di eliminare le sessioni presenti sulla piattaforma	AdminSessionManagementService

Sottosistema Autenticazione

Servizio	Descrizione	Interfaccia
Registrazione	Permette all'ospite di registrarsi alla piattaforma	AutenticazioneService
Login	Permette l'accesso all'utente sulla piattaforma	AutenticazioneService
Validazione	Permette di validare i dati di input	AutenticazioneService
Controllo Email	Permette di verificare se una email è già registrata	AutenticazioneService

Sottosistema Area Utente

Servizio	Descrizione	Interfaccia
Modifica Password	Permette di modificare la password di un utente	AccountService
Elimina Account	Permette ad un utente di eliminare il proprio account	AccountService
Controllo Prenotazioni attive	Permette di verificare se un utente ha prenotazioni attive	AccountService
Hashing Password	Permette di effettuare l'hashing SHA-256 della password	AccountService
HomePage	Reinderizza l'utente alla HomePage, caricando dati importanti come sessioni e mentor	HomeService

Dashboard Mentee	Permette di visualizzare le prenotazioni attive e concluse del Mentee	MenteeDashboardService
Dashboard Mentor	Permette di visualizzare le sessioni e le prenotazioni attive del Mentor	MentorDashboardService

Sottosistema Sessione

Servizio	Descrizione	Interfaccia
Crea Sessione	Permette di creare una sessione	SessionManagementService
Validazione campi sessione	Permette di controllare i campi della creazione di una sessione	SessionValidationService
Modifica Sessione	Permette di modificare una sessione presente nel sistema	SessionManagementService
Elimina Sessione	Permette di eliminare una sessione presente nel sistema	SessionManagementService

Sottosistema Prenotazione

Servizio	Descrizione	Interfaccia
Cerca sessioni	Permette di cercare una sessione	SessionRetrievalService
Visualizza sessione	Permette di visualizzare i dettagli di una sessione	SessionRetrievalService
Controllo Disponibilità	Permette di verificare la disponibilità di una sessione	BookingService
Crea Prenotazione	Permette di creare una nuova prenotazione nello stato di attesa	BookingService
Conferma Prenotazione	Permette di confermare una prenotazione	BookingService
Pagamento	Permette di effettuare il pagamento di una prenotazione	PaymentService
Validazione Carta	Permette di controllare i dati inerenti alla carta	PaymentService
Notifica	Permette di notificare mentor e mentee dell'acquisto della sessione	PaymentService

5. Glossario

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

Sigla/Termine	Definizione
Design Goal	Qualità sulle quali il sistema deve essere focalizzato
Dati persistenti	Dati non volatili, salvati all'interno di un database
Three-Tier Architecture	Architettura a tre livelli (Interfaccia, Logica Applicativa, Storage)
DG	Obiettivi di progettazione del sistema
Admin	Sottosistema per la gestione amministrativa della piattaforma
Application Logic Layer	Livello che contiene la logica di business del sistema
Storage Layer	Livello responsabile della gestione dei dati persistenti
Observer Pattern	Pattern utilizzato per la gestione delle notifiche in tempo reale
Object Pool Pattern	Pattern per l'ottimizzazione delle connessioni al database
DAO Pattern	Pattern per la separazione della logica di business dall'accesso ai dati
Jakarta Servlet API	Tecnologia utilizzata per la gestione delle richieste HTTP
JDBC (Java Database Connectivity)	Tecnologia per la comunicazione con il database MySQL
SHA-256	Algoritmo di hashing usato per la protezione delle password
Bootstrap 5	Framework per la creazione di interfacce responsive
AJAX	Tecnologia per richieste asincrone e miglioramento della UX

Apache Tomcat	Server per l'esecuzione di applicazioni Java web-based
Sessione	Un'unità di mentoring creata da un Mentor e prenotabile da un Mentee
Prenotazione	Processo di acquisto e conferma di una sessione da parte di un Mentee
Hashing Password	Processo di crittografia delle password per la sicurezza degli utenti