



PT-LINKER

SDD System Design Document

PT-LINKER

Riferimento	C17_SDD_ver.1.0
Versione	1.0
Data	03/12/2024
Destinatario	Prof.sso Carmine Gravino
Presentato da	C17 Team Triade
Approvato da	



Revision History

Data	Versione	Descrizione	Autori
03/12/2024	0.1	Prima stesura	AF/CC
04/02/2025	1	Revisione	CC
05/02/2025	1.1	Revisione + Correzioni ODD	AF/DS



Team members

Nome	Ruolo nel progetto	Acronimo	Informazioni di contatto
Andrea Filipuzzi	Team Member	AF	a.filipuzzi@studenti.unisa.it
Carmine Citro	Team Member	CC	c.citro27@studenti.unisa.it
Davide Santillo	Team Member	DS	d.santillo@studenti.unisa.it



Sommario

Revision History	2
Team members	3
1 Introduzione	5
1.1 Scopo del sistema	5
1.2 Obiettivi di Design (Design Goals).....	5
1.3 Organizzazione del Documento.....	7
2 Architettura dei sistemi già esistenti.....	7
3 Architettura del sistema proposto	8
3.1 Panoramica sulla sezione	8
3.2 Decomposizione in sottosistemi	8
3.3 Mapping Hardware/Software	16
3.4 Controllo degli Accessi e Sicurezza	16
3.5 Controllo globale del software	17
3.6 Condizioni Limite	17
4 Servizi dei sottosistemi	21
5 Cenni di ODD.....	26
5.1 design pattern	26
5.2 Obiettivo	26
5.3 implementazione	27

1 Introduzione

1.1 Scopo del sistema

PT-LINKER si propone di semplificare l'interazione tra professionisti, nel settore della nutrizione e del fitness, e clienti, al fine di avvicinare sempre più persone al mondo del fitness.

Attraverso il software i professionisti creano e personalizzano il piano di allenamento e programmi nutrizionali, monitorano i progressi dei clienti tramite report, e comunicare con loro attraverso una funzione di messaggistica integrata. I clienti visualizzano i programmi che gli sono stati assegnati, inseriscono le loro misurazioni corporee per tener traccia dei progressi nel tempo.

Il sistema, gestito da un Amministratore, permette l'iscrizione da parte di personal trainer, nutrizionisti e di conseguenza da parte di clienti.

1.2 Obiettivi di Design (Design Goals)

Nella presente sezione si andranno a presentare i Design Goals, ovvero le caratteristiche principale che devono essere perseguite durante lo sviluppo per garantire che il sistema soddisfi i requisiti funzionali e non funzionali.

I Design Goals sono stati suddivisi nelle seguenti categorie:

- Performance
- Dependability
- Maintenance
- End User

Ciascun Design Goals è descritto da:

- Rank
- ID Design Goal
- Descrizione
- Categoria
- RNF di origine

Rank	ID Design Goal	Descrizione	Categoria	RNF di origine
3	DG_1 Tempo di risposta	Il sistema deve garantire una risposta operativa di al massimo 15 secondi.	Performance	RNF_P_5
4	DG_2 Capacità di memoria	Il sistema deve garantire una quantità di memorizzazione superiore a 1tb per il corretto funzionamento di esso.	Performance	RNF_P_4
6	DG_3 Navigazione Utenti	Il sistema deve garantire il corretto funzionamento anche con un numero di utenti connessi superiore ai 500.	Performance	RNF_P_3
2	DG_4 Affidabilità dell'operazione	Il sistema deve garantire la corretta esecuzione di tutte le operazioni, senza possibili fallimenti.	Dependability	RNF_A_1
11	DG_5 Gestione dei fallimenti	Il sistema in caso di fallimento deve notificare l'utente dell'errore avvenuto.	Dependability	RNF_A_3
5	DG_6 Gestione dei Permessi	Il sistema deve garantire la distinzione dei permessi in base al ruolo dell'utente.	Dependability	RNF_A_4
1	DG_7 Sicurezza dei dati	Il sistema deve garantire una gestione sicura dei dati sensibili, assicurandone la visualizzazione dei dati personali solo agli utenti che hanno il diritto di accedervi.	Dependability	RNF_A_5 RNF_L_1
7	DG_8 Disponibilità	Il sistema dovrà essere disponibile dalle 3:00 a.m. alle 2:55 a.m.	Dependability	RNF_P_2

10	DG_9 Manutenibilità	Il sistema deve essere sviluppato in modo da garantire una facile e corretta manutenibilità di esso.	Maintenance	RNF_S_1
8	DG_10 Facilità d'uso	Il sistema deve essere usabile anche da un'utenza non esperta, senza l'utilizzo di documentazione.	End User	RNF_U_1 RNF_U_3
9	DG_11 Interfaccia intuitiva	L'interfaccia utente deve essere ottimizzata in modo da facilitare l'utilizzo per chi ne usufruisce.	ENF User	RNF_U_2

Trade-Off

Trade-Off	Descrizione
Tempi di risposta VS Navigazione Utente	Per garantire la connessione concorrente di un elevato numero di utenti, il sistema potrebbe subire un rallentamento nei tempi di risposta, influenzando negativamente l'esperienza di navigazione.
Manutenibilità VS Disponibilità	Per assicurare una manutenibilità efficace e regolare del sistema, potrebbero verificarsi periodi di indisponibilità o degrado del servizio, impattando la continuità operativa.

1.3 Organizzazione del Documento

Il presente documento di System Design è composto da quattro sezioni:

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

Architetture software già esistenti: Viene descritto i sistemi presenti ora sul mercato che hanno funzionalità simili al sistema proposto.

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

2 Architettura dei sistemi già esistenti

Oggi il mercato offre numerosi software pensati per migliorare l'interazione tra personal trainer, nutrizionisti e clienti. Questi strumenti sono progettati per semplificare la gestione delle attività professionali e offrire un'esperienza personalizzata ai clienti. In generale, queste piattaforme

combinano funzionalità di pianificazione, monitoraggio e comunicazione per consentire un supporto continuo e mirato.

Alcuni software, come *Trainerize* o *My PT Hub*, si concentrano sulla gestione integrata dei clienti, permettendo di organizzare allenamenti, elaborare piani alimentari e monitorare i progressi in un'unica soluzione. Sono particolarmente utili per personal trainer e nutrizionisti che operano in studi professionali o palestre.

3 Architettura del sistema proposto

3.1 Panoramica sulla sezione

Il sistema proposto è basato sullo stile architetturale Three Tier. Il motivo della presente scelta è che tale architettura è perfetta per lo sviluppo di web application come il nostro sistema, poiché la separazione della logica di presentazione da quella di elaborazione, migliora una serie di qualità, tra le quali:

- Leggibilità
- Manutenzione
- Riutilizzo

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

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

Per la gestione del database saranno usati:

- Java JDBC (Java Database Connectivity) per il collegamento al database.
- SQL per il database e DropBox per il salvataggio su cloud dei file.

3.2 Decomposizione in sottosistemi

I sottosistemi individuati sono:

- Registrazione
- Autenticazione
- Pagamento
- Recensione
- Scheda
- Dieta
- Chat
- Progressi
- Storage
- Database

Sono mostrate di seguito le dipendenze tra sottosistemi:

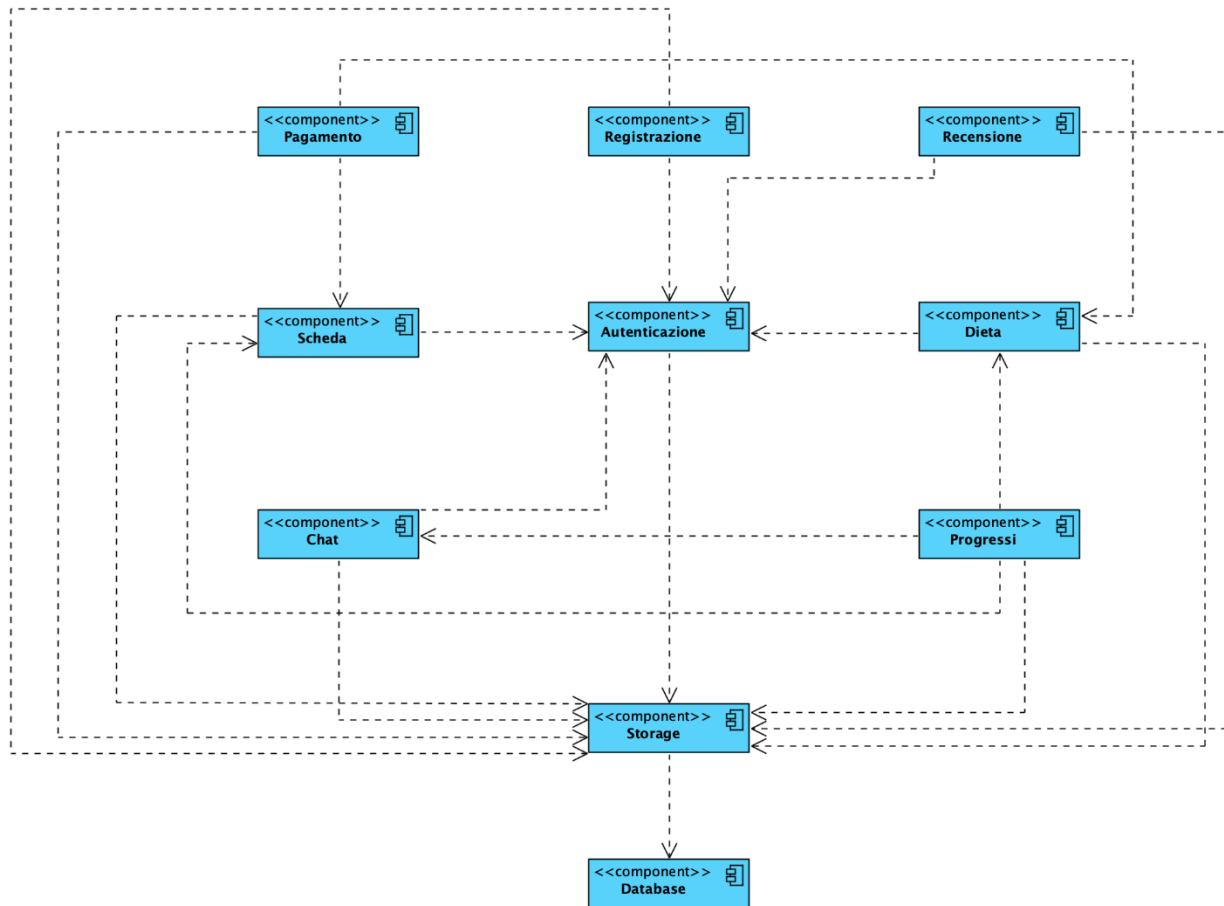
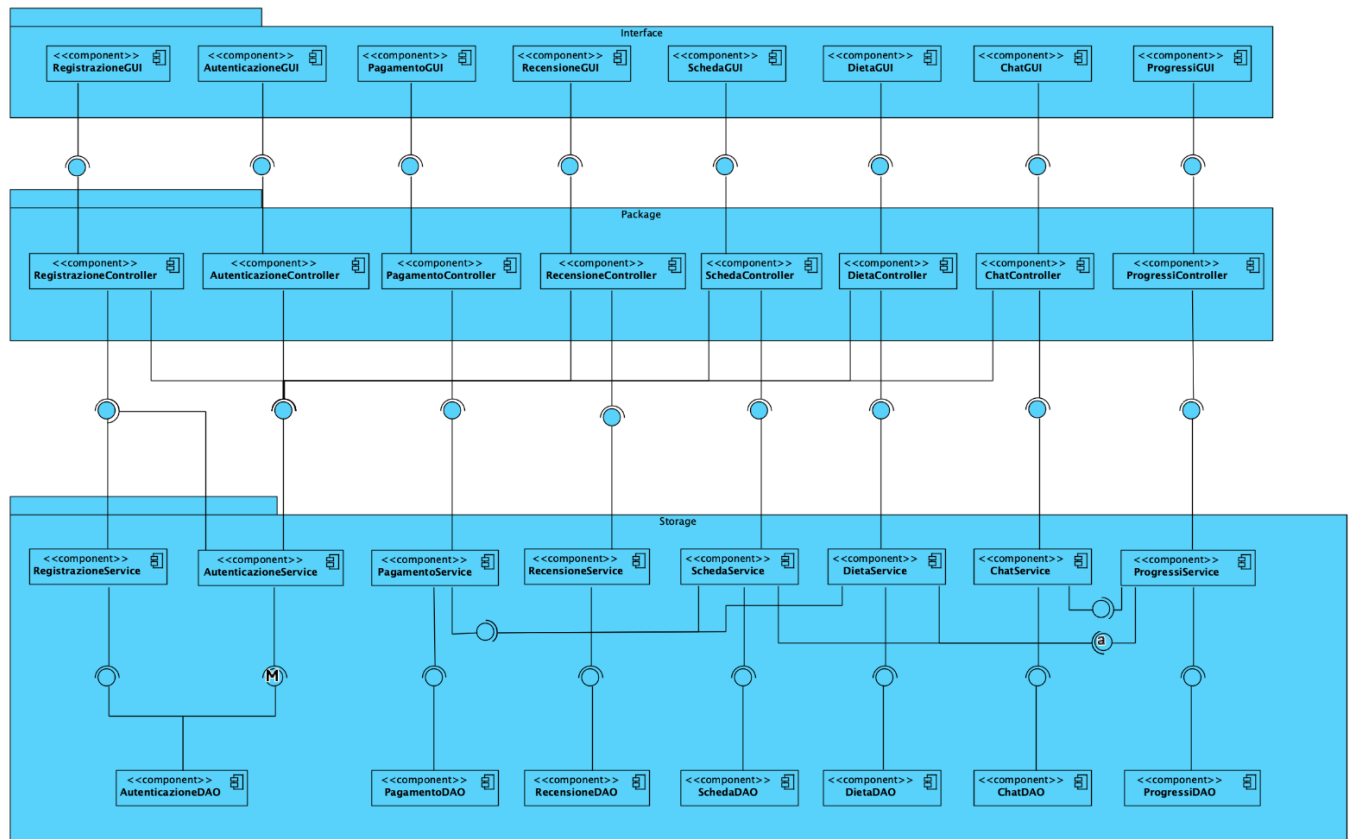
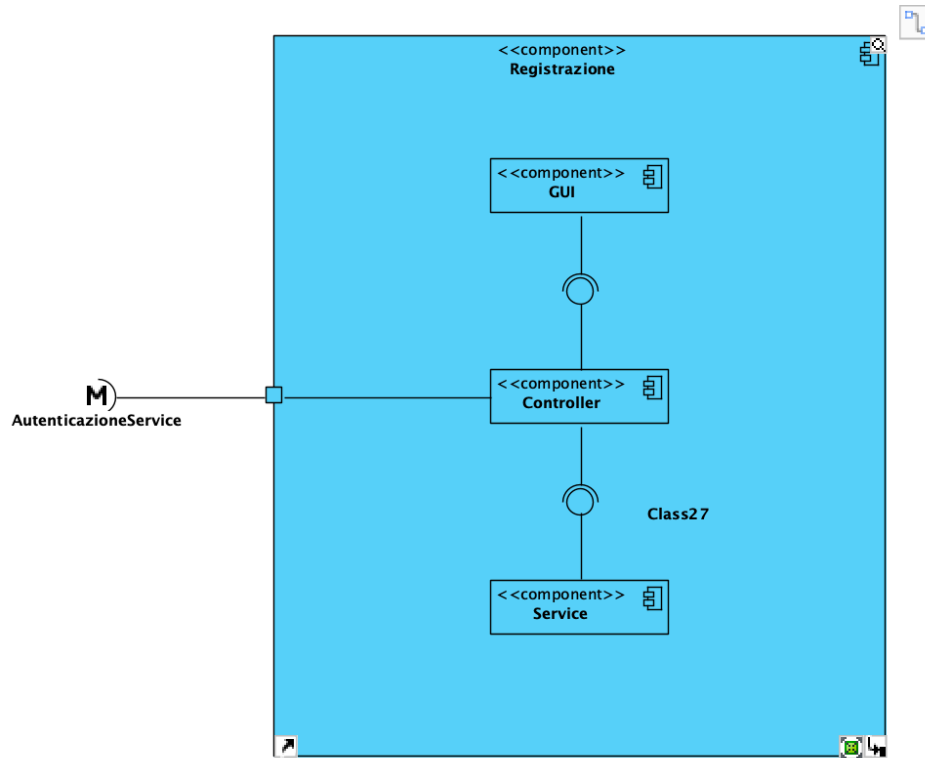


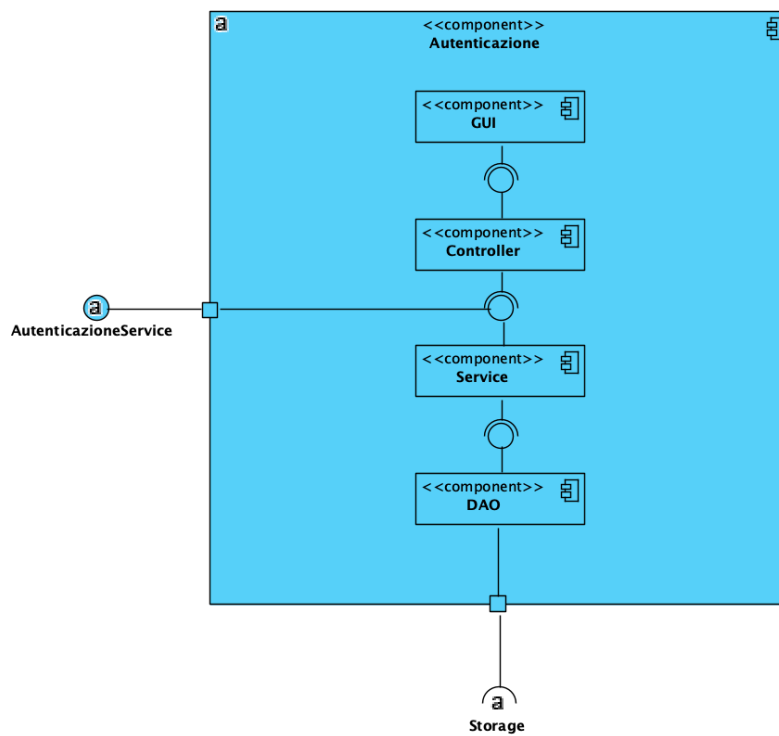
Diagramma Architeturale



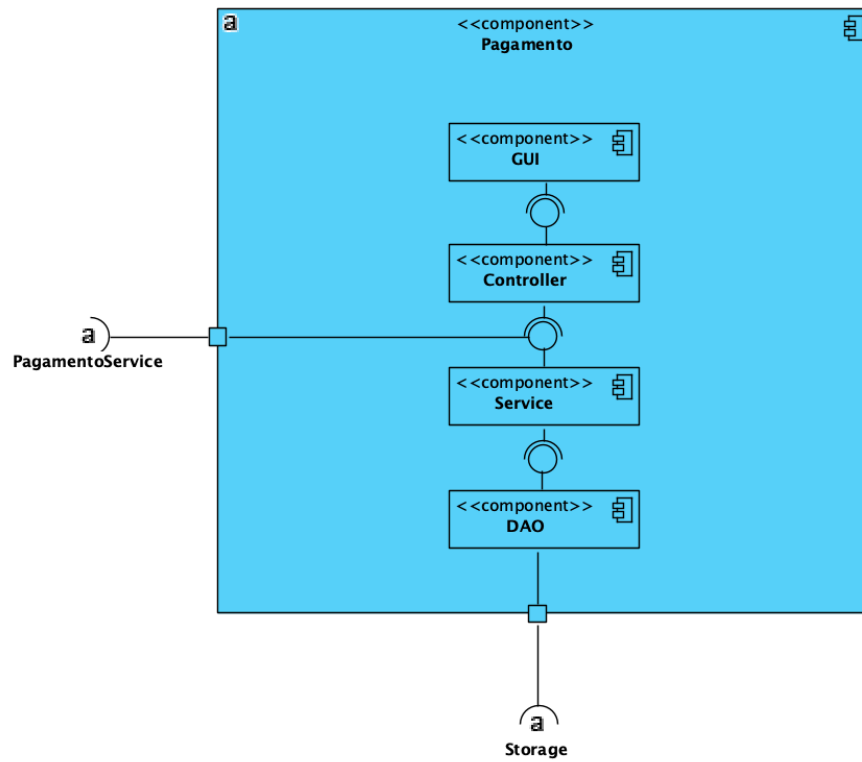
Sottosistema Registrazione



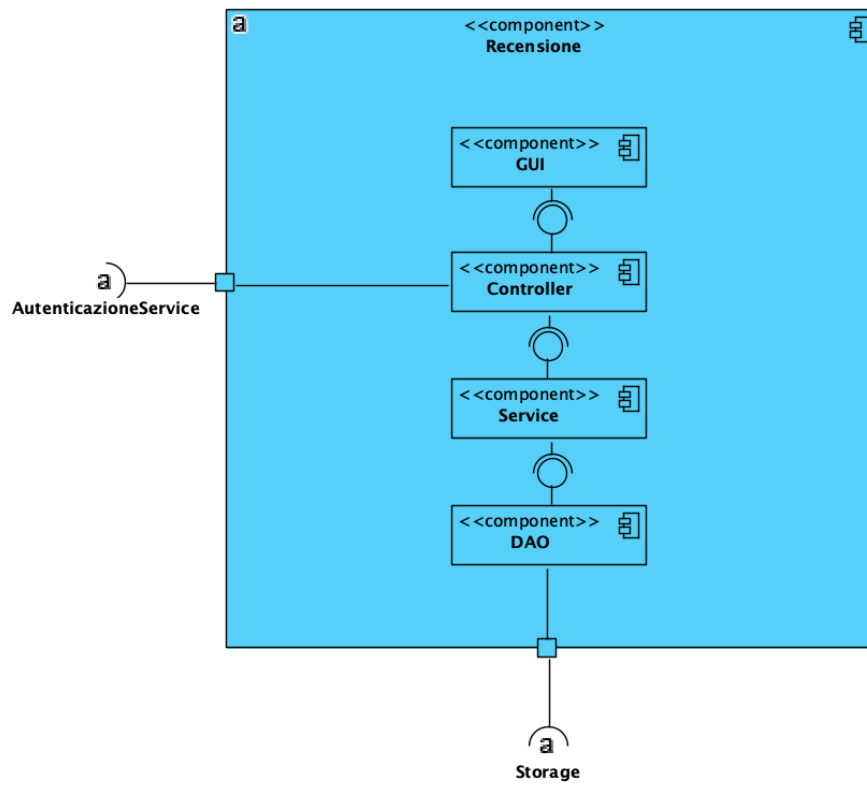
Sottosistema Autenticazione



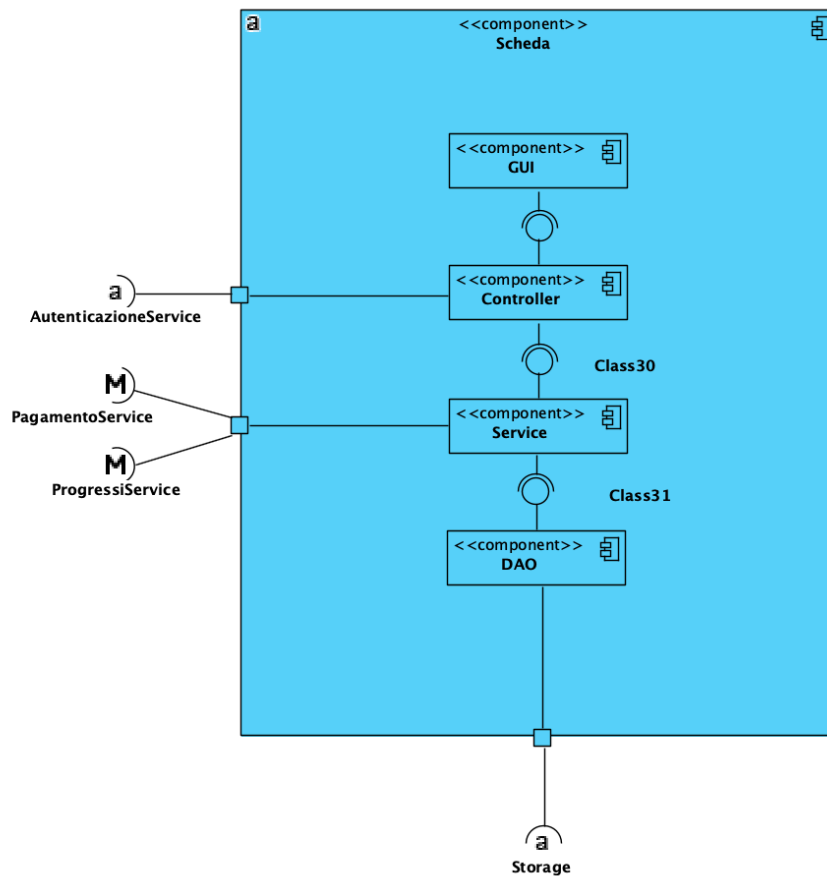
Sottosistema Pagamento



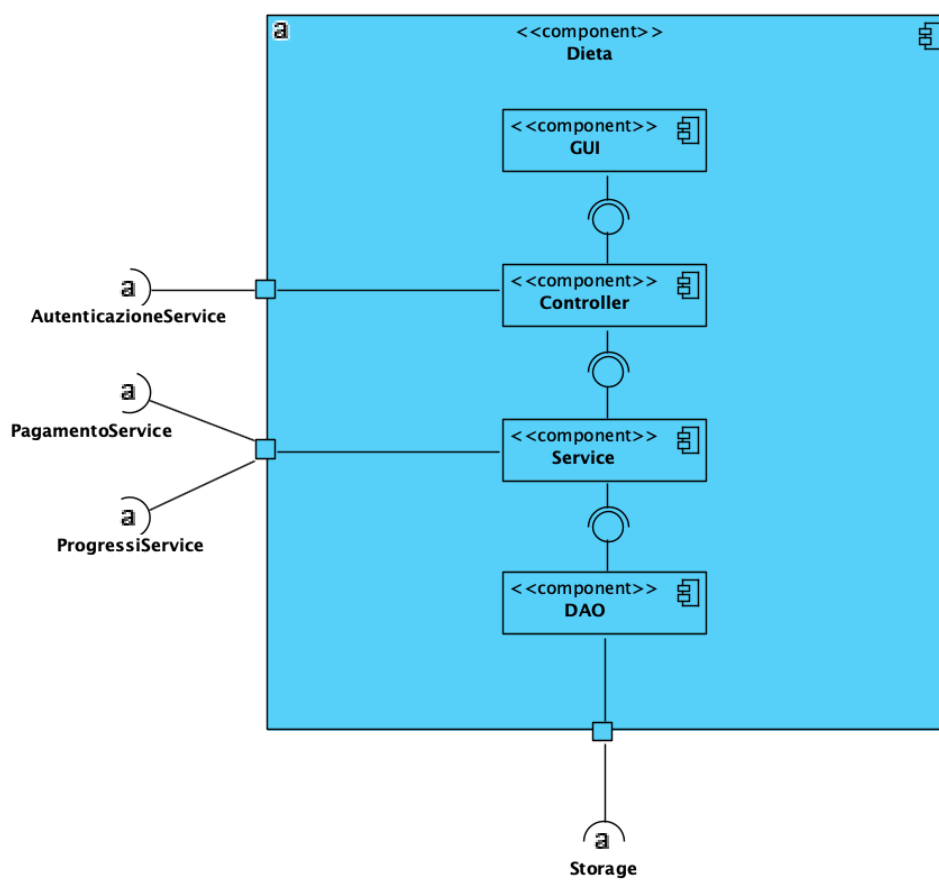
Sottosistema Recensione



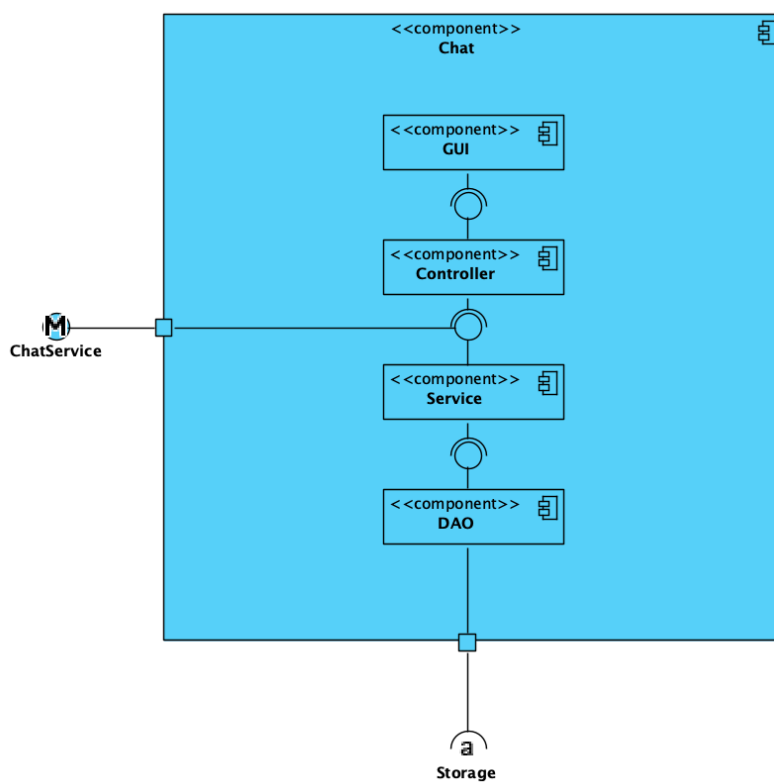
Sottosistema Scheda



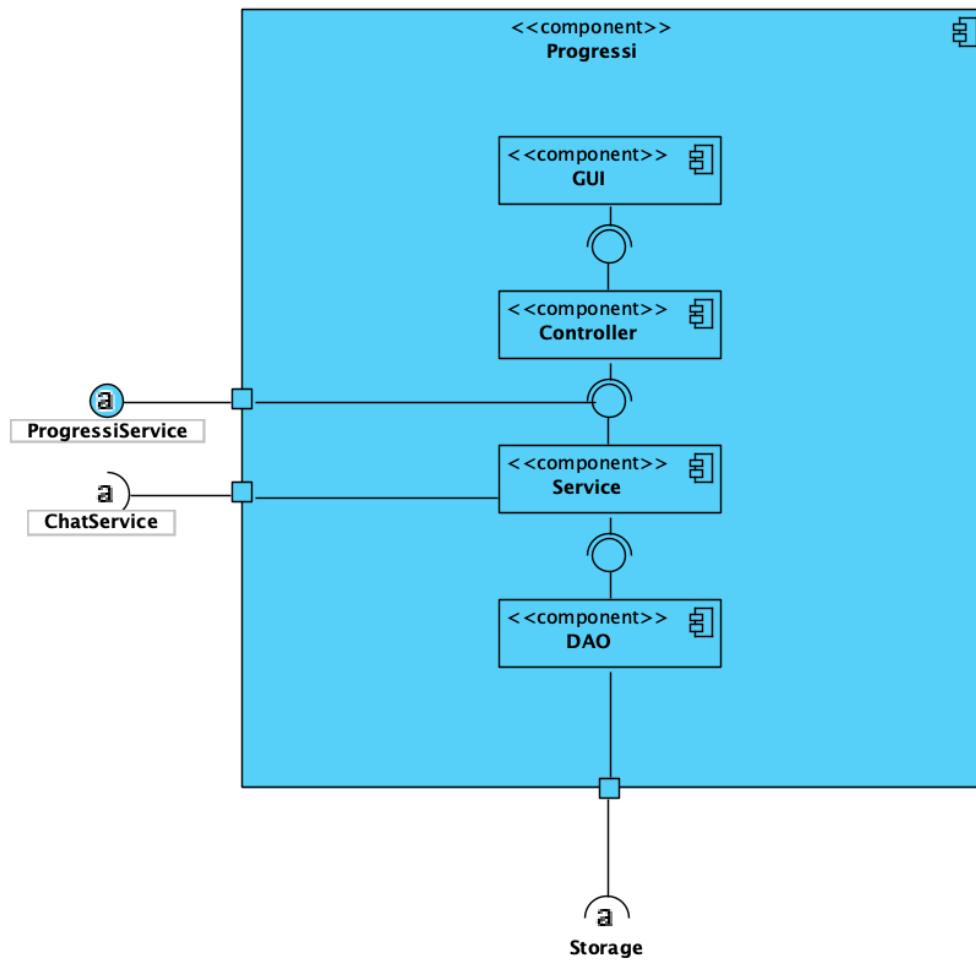
Sottosistema Dieta



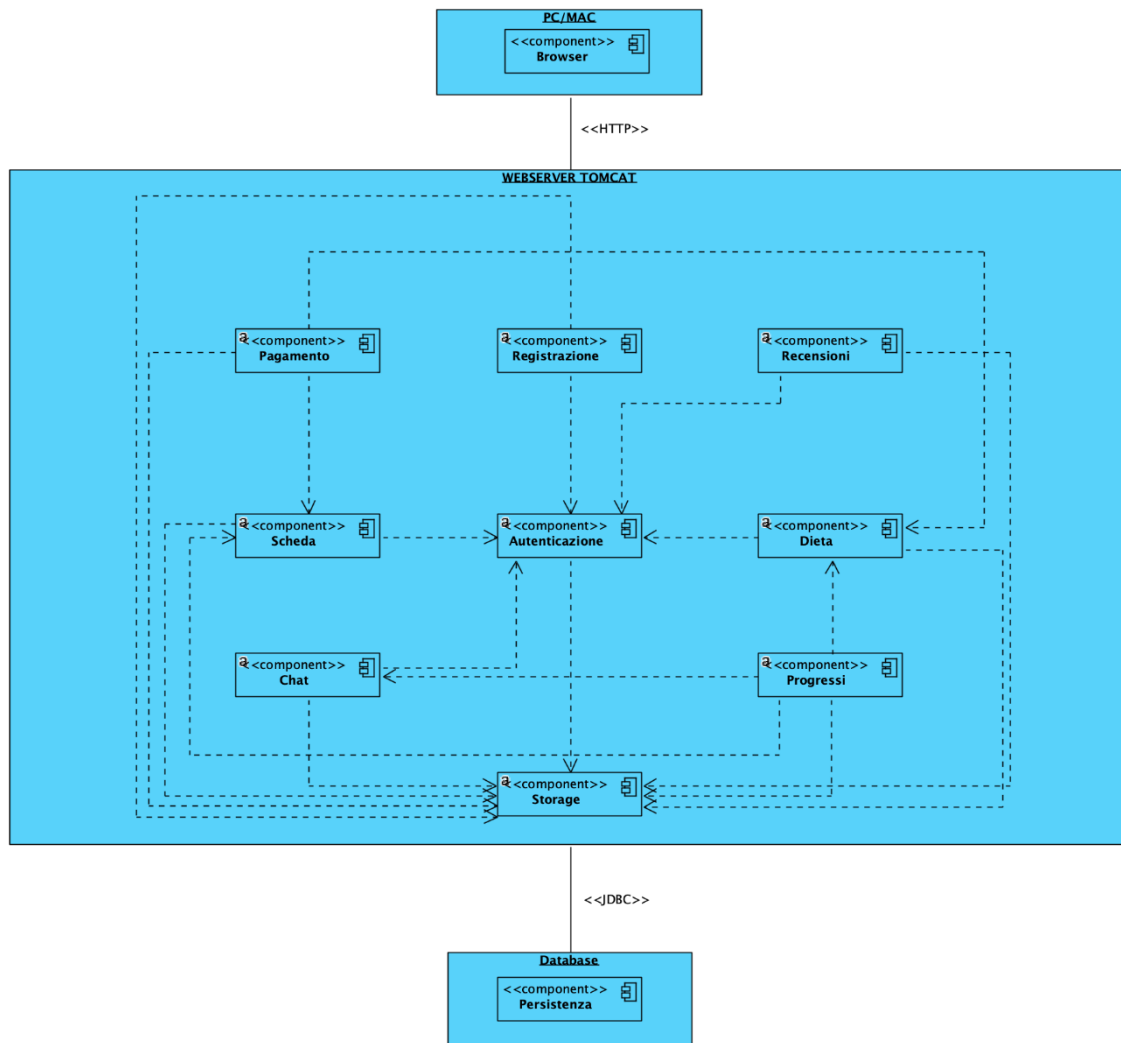
Sottosistema Chat



Sottosistema Progressi
SDD PT-LINKER 1.0



3.3 Mapping Hardware/Software



3.4 Controllo degli Accessi e Sicurezza

Attori	Cliente	Nutrizionista	Personal Trainer	Amministratore
Oggetti				
Registrazione	Registrazione Cliente	Registrazione Nutrizionista	Registrazione Personal Trainer	
Autenticazione	Login Logout Visualizzazione area utente Modifica dati Cancellazione Account Visualizzazione Storico Nutrizionista/Personal Trainer	Login Logout Visualizzazione area utente Modifica dati Cancellazione Account Visualizzazione Clienti Aggiungi Certificazioni	Login Logout Visualizzazione area utente Modifica dati Cancellazione Account Visualizzazione Clienti Aggiungi Certificazioni	Login Logout Cancellazione Account Verifica Certificazioni
Scheda	Cancellazione Scheda Visualizzazione Scheda		Gestione Scheda Modifica Scheda Cancellazione Scheda Visualizzazione Scheda	

Dieta	Visualizzazione Dieta	Creazione Dieta Modifica Dieta Cancellazione Dieta Visualizzazione Dieta		
Chat	Creazione Chat Personal Trainer Creazione Chat Nutrizionista			Cancellazione Chat Personal Trainer Cancellazione Chat Nutrizionista
Progressi	Aggiornamento Progressi Visualizzazione Progressi Modifica Progressi Visualizzazione feedback	Visualizzazione Progressi Creazione Feedback progressi Visualizzazione feedback	Visualizzazione Progressi Creazione Feedback progressi Visualizzazione feedback	
Recensione	Aggiungi Recensione Visualizza Recensione	Visualizza Recensione	Visualizza Recensione	Cancella recensione Visualizza Recensione
Pagamento	Effettua Pagamento Visualizzazione Pagamenti Aggiunta Pagamento	Visualizzazione Pagamenti Aggiunta Pagamento	Visualizzazione Pagamenti Aggiunta Pagamento	

3.5 Controllo globale del software

Il sistema PT_LINKER è un sistema interattivo per cui ogni funzionalità viene eseguita in seguito ad un comando impartito dall'utente tramite l'interfaccia grafica.

Quando un utente vuole accedere ed utilizzare una funzionalità del sistema può farlo tramite l'interfaccia grafica la quale selezionerà il controllo corrispondente. L'azione scatenerà un evento che verrà gestito dal suo handler. Quest'ultimo a sua volta indirizzerà il controllo del flusso di eventi al sottosistema che si occupa della logica di controllo e gestore del controllo che poi si rivolge ai servizi per la logica applicativa.

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

3.6 Condizioni Limite

Avvio del sistema

Identificativo	UCBC_1 – Avvio del Sistema	Data	02/12/2024
		Versione	1.0
		Autori	Carmine Citro
Descrizione	Il sistema viene avviato		
Attore principale	Amministratore		
Attori secondari	NA		
Entry condition	L'Amministratore accede al Server		
Exit condition	Il sistema viene avviato correttamente		

On success		
Exit condition		
On failure		
Flusso di eventi principale		
1	Amministratore	Digita da prompt dei comandi il comando per accedere al server.
2	Sistema	Verifica la sanità dei dati persistenti e rende disponibile i servizi.
I Flusso di Eventi Alternativo: I Dati Persistenti sono danneggiati		
2.a1	Sistema	Notifica l'Amministratore di problemi ai dati persistenti e non effettua l'avvio.
2.a2	Amministratore	Corregge i dati persistenti
2.a3	Amministratore	Esegue il Passaggio 1

Arresto del sistema

Identificativo	UCBC_2 – Arresto del sistema	Data	02/12/2024
		Versione	1.0
		Autori	Carmine Citro
Descrizione	Il sistema viene arrestato		
Attore principale	Amministratore		
Attori secondari	NA		
Entry condition	L’Amministratore accede al Server AND Il sistema è stato già avviato in precedenza		
Exit condition On success	Il sistema viene arrestato		
Exit condition On failure	Il sistema non viene arrestato		
Flusso di eventi principale			
1	Amministratore	Invia un segnale di arresto al Sistema dal Server.	
2	Sistema	Controlla che non ci siano connessioni ancora aperte, se non ci sono, arresta il sistema.	
I Flusso di eventi alternativo: Ci sono connessioni ancora aperte			
2.a1	Sistema	Notifica all’Amministratore che ci sono ancora connessioni aperte.	
2.a2	Sistema	Attende per rispondere a eventuali richieste al server.	

2.a3	Sistema	Controlla che non ci siano connessioni al server e termina l'esecuzione del sistema.
2.a4	Sistema	Notifica l'Amministratore dell'avvenuto arresto del sistema.
II Flusso di eventi alternativo: Ci sono connessioni ancora aperte		
2.a3.a1	Sistema	Recide le connessioni verso l'esterno.
2.a3.a2	Sistema	Notifica l'Amministratore dell'avvenuto arresto del sistema.

Fallimento del sistema

Identificativo	UCBC_3 – Fallimento del Sistema	Data	02/12/2024
		Versione	1.0
		Autori	Carmine Citro
Descrizione	Comportamento del sistema in caso di fallimento		
Attore principale	Amministratore		
Attori secondari	NA		
Entry condition	Il Sistema viene terminato inaspettatamente		
Exit condition On success	Il Sistema viene riavviato correttamene		
Exit condition On failure	Il Sistema non viene riavviato		
Flusso di eventi principale			
1	Amministratore	Include UCBC_1	

Errore di Accesso ai Dati Persistenti

Identificativo	UCBC_4 – Errore di Accesso ai Dati Persistenti	Data	02/12/2024
		Versione	1.0
		Autori	Carmine Citro
Descrizione	Comportamento del sistema in caso di errore all'accesso ai dati persistenti del sistema.		
Attore principale	Amministratore		
Attori secondari	NA		
Entry condition	Il Sistema non può accedere ai dati persistenti		
Exit condition	Il Sistema riprende il normale funzionamento		

On success		Il Sistema non riprende il normale funzionamento
Exit condition		
On failure		
Flusso di eventi principale		
1	Sistema	Notifica l'amministratore dell'impossibilità di accedere ai dati persistenti.
2	Sistema	Cessa di processare eventuali richieste dall'esterno e risponde a tutte le richieste con un messaggio di errore.
3	Amministratore	Include UCBC_2
4	Amministratore	Ripristina l'accessibilità o la sanità dei dati persistenti.
5	Amministratore	Include UCBC_1

4 Servizi dei sottosistemi

Sottosistema Registrazione

Servizio	Descrizione	Interfaccia
Registrazione Cliente	Questa funzionalità permette di registrarsi sulla piattaforma come cliente	RegistrazioneService
Registrazione Personal Trainer	Questa funzionalità permette di registrarsi sulla piattaforma come Personal Trainer	RegistrazioneService
Registrazione Nutrizionista	Questa funzionalità permette di registrarsi sulla piattaforma come Nutrizionista	RegistrazioneService

Sottosistema Autenticazione

Servizio	Descrizione	Interfaccia
Login	Questa funzionalità permette di effettuare l'accesso al sistema tramite le proprie credenziali per sfruttare le proprie funzionalità	AutenticazioneService
Logout	Questa funzionalità permette di disconnettersi dal sistema	AutenticazioneService

Visualizzazione Area Utente	Questa funzionalità permette di visualizzare i propri dati utente	AutenticazioneService
Modifica Dati	Questa funzionalità permette di modificare i propri dati	AutenticazioneService
Cancellazione Account	Questa funzionalità permette di effettuare la cancellazione del proprio profilo dal sistema	AutenticazioneService
Visualizzazione Storico	Questa funzionalità permette al cliente di avere uno storico di tutti i personal trainer/nutrizionisti con cui ha interagito	AutenticazioneService
Visualizzazione clienti	Questa funzionalità permette ai nutrizionisti/personal trainer di avere un elenco di tutti i loro clienti	AutenticazioneService
Aggiungi Certificazioni	Questa funzionalità permette ai professionisti di poter aggiungere nuove certificazioni	AutenticazioneService
Verifica Certificazioni	Questa funzionalità permette all'amministratore del sistema di verificare le certificazioni dei professionisti	AutenticazioneService

Sottosistema Scheda

Servizio	Descrizione	Interfaccia
Creazione Scheda	Questa funzionalità permette al personal trainer di creare una nuova scheda	SchedaService

Modifica Scheda	Questa funzionalità permette al personal trainer di modificare la scheda	SchedaService
Visualizzazione Scheda	Questa funzionalità permette all'utente/personal trainer di visualizzare la scheda	SchedaService
Cancellazione Scheda	Questa funzionalità permette all'utente/personal trainer di cancellare la scheda	SchedaService

Sottosistema Dieta

Servizio	Descrizione	Interfaccia
Creazione Dieta	Questa funzionalità permette al nutrizionista di creare una nuova dieta	DietaService
Modifica Dieta	Questa funzionalità permette al nutrizionista di modificare la dieta	DietaService
Visualizzazione Dieta	Questa funzionalità permette al nutrizionista/cliente di visualizzare la dieta	DietaService
Cancellazione Dieta	Questa funzionalità permette al nutrizionista/cliente di cancellare la dieta	DietaService

Sottosistema Chat

Servizio	Descrizione	Interfaccia
Creazione Chat	Questa funzionalità permette al cliente di creare una nuova chat con il personal trainer/nutrizionista	ChatService

Cancellazione Chat	Questa funzionalità permette all'amministratore di cancellare una chat	ChatService
---------------------------	--	-------------

Sottosistema Progressi

Servizio	Descrizione	Interfaccia
Aggiornamento Progressi	Questa funzionalità permette al cliente di caricare i propri progressi visibili ai professionisti	ProgressiService
Modifica Progressi	Questa funzionalità permette al cliente di modificare i progressi caricati	ProgressiService
Visualizzazione Progressi	Questa funzionalità permette ai professionisti e al cliente di visualizzare i progressi	ProgressiService
Creazione Feedback Progressi	Questa funzionalità permette ai professionisti di rilasciare dei feedback sui progressi dei propri clienti	ProgressiService
Visualizzazione Feedback	Questa funzionalità permette ai professionisti e ai clienti di visualizzare i feedback rilasciati	ProgressiService

Sottosistema Recensione

Servizio	Descrizione	Interfaccia
Aggiungi Recensione	Questa funzionalità permette al cliente di aggiungere delle recensioni sulle qualità dei professionisti con cui ha interagito	RecensioneService

Visualizza Recensione	Questa funzionalità permette agli utenti di visualizzare le recensioni pubblicate dai clienti	RecensioneService
Cancella Recensione	Questa funzionalità permette all'amministratore di cancellare una recensione	RecensioneService

Sottosistema Pagamento

Servizio	Descrizione	Interfaccia
Effettua Pagamento	Questa funzionalità permette al cliente di effettuare un pagamento	PagamentoService
Visualizzazione Pagamento	Questa funzionalità permette agli utenti di visualizzare i pagamenti effettuati/ricevuti	PagamentoService
Aggiunta Pagamento	Questa funzionalità permette ai professionisti di aggiungere un pagamento ad un cliente	PagamentoService

5 Cenni di ODD

5.1 design pattern

I design pattern che sono stati utilizzati sono il facadepattern e observer.

la classe **DietaFacade** viene utilizzata per gestire operazioni legate alla gestione delle diete in modo centralizzato, applicando il **Facade Pattern** per semplificare l'interazione con più classi sottostanti.

Il **Facade Pattern** è un **Design Pattern Strutturale** che fornisce un'interfaccia semplificata e unificata a un insieme complesso di classi.

Il suo obiettivo principale è **nascondere la complessità del sistema** e offrire un **punto di accesso unico** per interagire con più sottosistemi.

Come secondo design pattern è stato usato l'observer per notificare al cliente quando la sua dieta viene modificata.

L'**Observer Pattern** è un **Design Pattern Comportamentale** che permette di notificare **automaticamente** più oggetti (Observers) quando lo stato di un altro oggetto (Subject) cambia.

L'Osservatore rappresenta **un cliente che osserva il suo professionista**.

- Il **Professionista** è il **Subject** → Quando cambia qualcosa (es. dieta aggiornata), notifica i suoi clienti.
- Ogni **Cliente** è un **Observer** → Riceve aggiornamenti automatici sulle modifiche.

5.2 Obbiettivo

Gli obbiettivi che vogliamo raggiungere utilizzando il facade pattern sono:

Semplificare l'uso di un sistema complesso → L'utente finale utilizza solo il Facade, senza dover conoscere i dettagli interni.

Ridurre le dipendenze tra i componenti → Il client interagisce solo con il Facade, non direttamente con le classi sottostanti.

Migliorare la manutenibilità e l'estensibilità → Se il sottosistema cambia, basta modificare il Facade senza impattare il client.

Non limitare l'accesso diretto → Il client può ancora accedere direttamente ai sottosistemi se necessario.

Gli obiettivi che vogliamo raggiungere con l'observer sono di permettere al **Professionista (Subject)** di mantenere aggiornati i suoi **Clienti (Observers)** in modo **automatico e reattivo**.

5.3 implementazione

Nella nostra servlet, utilizziamo il **Facade Pattern** attraverso la classe DietaFacade. Questa classe **nasconde la complessità** della gestione delle diete e fornisce un'interfaccia semplificata per interagire con **database, file system e Dropbox**.

```
DietaFacade dietaFacade = new DietaFacade();
```

Qui creiamo un'istanza del Facade, che fungerà da punto di accesso unico per operazioni complesse.

Invece di gestire direttamente database, cartelle e file, la servlet **delega tutto** a DietaFacade.

```
String dieteFolderPath = getServletContext().getRealPath("/documenti/Diete");  
dietaFacade.ensureLocalFolderExists(dieteFolderPath);
```

Senza Facade: La servlet dovrebbe verificare e creare la cartella manualmente.

Con Facade: La servlet chiama **un solo metodo** e il Facade si occupa di tutto.

```
dietaFacade.downloadDietaFile(dropboxFilePath, localFilePath);
```

Senza Facade, la servlet dovrebbe:

- Creare una connessione con Dropbox.
- Scaricare il file.
- Gestire eventuali errori.

Con Facade, basta una **sola chiamata** → Il metodo downloadDietaFile si occupa di tutto.

DietaFacade agisce da **punto di accesso centralizzato** per tutte le operazioni complesse legate alla visualizzazione delle diete, semplificando il codice della servlet e rendendolo più leggibile e manutenibile.

L'Observer Pattern prevede due ruoli principali:

1Subject (Soggetto osservato)

- Tiene traccia degli **Osservatori (Observers)**.
- Notifica automaticamente gli Osservatori quando cambia il suo stato.

2Observers (Osservatori/Clienti)

- Si iscrivono al **Subject** per ricevere aggiornamenti.



- Reagiscono ai cambiamenti notificati dal **Subject**.

```
public interface Osservatore {  
void aggiorna(String messaggio);  
}
```

Ogni Cliente dovrà implementare il metodo `aggiorna()` per ricevere notifiche.

Ogni Cliente è un **Observer** e può ricevere notifiche dal Professionista.

Il **Professionista** gestisce un elenco di **Clienti (Observers)** e può inviare notifiche.