Laurea Magistrale in informatica-Università di Salerno
Corso di *Gestione dei Progetti Software*- Prof.ssa F.Ferrucci

FITDIARY

DECIDE, COMMIT, SUCCEED

System Design Document

Riferimento	2021_SDD_C07_FitDiary_Fasano-Spinelli_V1.1
Versione	1.1
Data	21/01/2022
Destinatario	Prof.ssa Filomena Ferrucci, Prof.re Fabio Palomba
Presentato da	C07 Team FitDiary: Daniele De Marco, Ilaria De Sio, Rebecca Di Matteo, Daniele Giaquinto, Davide La Gamba, Leonardo Monaco, Simone Spera, Antonio Trapanese.
Approvato da	

Revision History

Data	Versione	Descrizione	Autori
29/11/2021	0.1	SDD-Stesura Design Goals	Antonio Trapanese, Daniele De Marco, Daniele Giaquinto, Davide La Gamba, Rebecca Di Matteo, Simone Spera
30/11/2021	0.2	SDD-Architettura Sistema Corrente	Ilaria De Sio Leonardo Monaco
30/11/2021	0.3	SDD-Stesura Design Trade-Off	Tutto il team
01/12/2021	0.4	SDD-Stesura Boundary Conditions	Ilaria De Sio Davide La Gamba
01/12/2021	0.5	SDD-Mapping HW/SW	Rebecca Di Matteo Leonardo Monaco Simone Spera
01/12/2021	0.6	SDD-Decomposizione in Sottosistemi	Antonio Trapanese Daniele De Marco Daniele Giaquinto
02/12/2021	0.7	SDD-Gestione Dati Persistenti	Tutto il team
03/12/2021	0.8	SDD-Controllo Accesso e Sicurezza	Antonio Trapanese Daniele De Marco Simone Spera
03/12/2021	0.9	SDD-Controllo Flusso Globale Sistema	Rebecca Di Matteo Leonardo Monaco
03/12/2021	1.0	SDD-Servizi dei Sottosistemi	Tutto il team
21/01/2022	1.1	Gestione dati persistenti	Antonio Trapanese Daniele De Marco Davide La Gamba Rebecca Di Matteo

Team Members

Ruolo	Nome e Cognome	Acronimo	Email
PM	Salvatore Fasano	SF	s.fasano10@studenti.unisa.it
PM	Gianluca Spinelli	GS	g.spinelli18@studenti.unisa.it
TM	Daniele De Marco	DM	d.demarco11@studenti.unisa.it
TM	Ilaria De Sio	IS	i.desio7@studenti.unisa.it
TM	Rebecca Di Matteo	RM	r.dimatteo10@studenti.unisa.it
TM	Daniele Giaquinto	DG	d.giaquinto2@studenti.unisa.it
TM	Davide La Gamba	DL	d.lagamba@studenti.unisa.it
TM	Leonardo Monaco	LM	l.monaco11@studenti.unisa.it
TM	Simone Spera	SS	s.spera7@studenti.unisa.it
TM	Antonio Trapanese	AT	a.trapanese8@studenti.unisa.it

Sommario

1. Introduzione

1.1 Scopo del sistema

Il sistema che si vuole realizzare ha come obiettivo principale quello di facilitare l'interazione tra i preparatori e i propri clienti e migliorare la gestione dei loro dati. Attraverso una piattaforma online il preparatore potrà creare account per i propri clienti e gestire i loro protocolli alimentari e/o di allenamento.

Attualmente l'interazione tra i preparatori e i clienti è gestita attraverso l'uso di software di messaggistica General Purpose o social network, i principali problemi riscontrati nell'uso di questa metodologia sono:

- La difficoltà di mantenere e accedere allo storico dei miglioramenti del cliente: alla scadenza di ogni protocollo il preparatore deve richiedere i progressi al cliente tramite software di messaggistica General Purpose o social network e memorizzarli in qualche software esterno o su carta, altrimenti tali andrebbero persi nelle chat. Questo rende complessa la gestione dei dati.
- Per i clienti risulta macchinoso fruire dei vari protocolli forniti dal proprio preparatore, poiché' quest'ultimo dovrebbe salvarli sul file system del proprio dispositivo e/o stamparli su carta.

Il sistema proposto permette

al preparatore di facilitare la gestione dei propri clienti e dei loro protocolli.

1.2 Design Goals e Trade-Off

1.2.1 Design Goals

Rank	ID Design Goal	Descrizione	Categoria	RNF di origine
12	DG_1 Tempo di riposta	Il sistema dovrebbe, fornire una risposta ad una richiesta in meno di 1 secondo nel 95% dei casi.	Performance	RNF_P_2
10	DG_2 Throughput	Il sistema deve supportare 1500 task al secondo con una media di 3 task per utente.	Performance	RNF_P_1
5	DG_3 Disponibilità	Il sistema deve risultare accessibile 24/7, salvo interventi di manutenzione che possono intercorrere in fasce orarie predeterminate.	Dependability	RNF_A_1
2	DG_4 Robustezza	Il sistema deve garantire robustezza, gestendo nel modo previsto il 90% degli input errati da parte dell'utente.	Dependability	RNF_A_2
6	DG_5 Sicurezza	Il sistema deve fornire una connessione sicura e cifrata, al fine di salvaguardare i dati degli utenti da accessi non autorizzati.	Dependability	RNF_A_3
9	DG_6 Portabilità	Il sistema deve essere sviluppato al 90% da codice riusabile, garantendone in futuro la portabilità su nuove tecnologie.	Maintenance	RNF_S_1
4	DG_7 Estensibilità	Lo sviluppo del sistema prevede l'utilizzo di Design Patterns per garantire l'estensibilità.	Maintenance	RNF_S_1

13	DG_8 Estensibilità	L'architettura del sistema dovrebbe poter essere ampliata per gestire carichi crescenti di utenza.	Maintenance	RNF_S_2
7	DG_9 Usabilità	Il sistema deve garantire al 98% degli utenti l'esecuzione delle operazioni senza commettere errori nel flusso di navigazione.	End User	RNF_U_1
3	DG_10 Usabilità	Le interfacce devono garantire uniformità grafica e stilistica utilizzando una palette di colori predeterminata nel 99.9% delle schermate.	End User	RNF_U_1
8	DG_11 Utilità	Il numero di operazioni per aggiungere un protocollo deve essere al massimo 8.	End User	RNF_U_2
11	DG_12 Utilità	La grafica dovrebbe essere responsive sul 90% dei dispositivi in commercio.	End User	RNF_U_3
1	DG_13 Costi di sviluppo	Il costo previsto per lo sviluppo del sistema ammonta a 50 ore/uomo (8 software engineer + 2 project manager)	Cost	Documenti di management

1.2.2 Design Trade-Off

Trade-off	Descrizione
Tempi di risposta vs Sicurezza	Il sistema, col fine di garantire una maggiore sicurezza, potrebbe richiedere un tempo di risposta maggiore che, in ogni caso, sarà mantenuto al di sotto di un secondo nell'80% dei casi.
Tempi di risposta vs Robustezza	Il sistema, col fine di garantire una maggiore robustezza, effettuerà dei controlli sui vari input. Tali controlli potrebbero alterare il tempo di risposta a meno di 3 secondi nell'80% dei casi.
Costi di sviluppo vs Usabilità	Il sistema, col fine di rientrare nel budget prestabilito, ridurrà la percentuale di utenti che eseguono operazioni senza commettere errori nel flusso di navigazione al 90%.
Costi di sviluppo vs Portabilità	Il sistema, col fine di rientrare nel budget prestabilito, ridurrà la percentuale di codice riusabile al 75%.
Costi di sviluppo vs Utilizzo COTS	Il sistema, col fine di rientrare nel budget prestabilito, utilizzerà componenti COTS, per gestire alcune funzionalità (ad esempio il pagamento), a discapito di un completo sviluppo in-house.

1.3 Definizioni, Acronimi e Abbreviazioni

Definizioni

- **Design Goal**: Obiettivi di design progettati per il sistema proposto;
- **Design Trade-off**: Scelte e compromessi tra design goals dissonanti;
- **Deployment Diagram**: Diagramma UML di specifica per le relazioni tra le componenti realizzate (con relative tecnologie implementative) e le risorse Hardware e Software necessarie al corretto funzionamento del sistema;
- **Varchar, Integer & Date**: Tipologie di dati necessarie per la memorizzazione dei dati persistenti nella base dati;
- **Access Control List**: Metodologia di controllo degli accessi ai sottosistemi individuati in fase di design, dove per ogni sottosistema sono specificate coppie (Attore, Operazione) indicanti ognuna un'operazione disponibile ad un determinato attore del sistema;
- **Diagramma dei servizi**: Diagramma di specifica dei servizi offerti dai sottosistemi progettati.

Acronimi

- **Sottosistema**: Un sottoinsieme dei servizi del dominio applicativo, formato da servizi legati da una relazione funzionale.
- **Mapping Hardware/Software**: Studio della connessione tra parti fisiche e logiche di cui si compongono il sistema.
- **Backend**: La parte che si occupa di gestire il funzionamento del sistema a seguito delle interazioni da parte del cliente nel Frontend, include anche la gestione dei dati persistenti nel Database e del sistema.
- **Frontend**: La parte visibile all'utente di un programma e con cui egli può interagire.
- **Database**: Architettura esterna che si occupa della gestione e della memorizzazione dei dati persistenti.
- **RAD:** Requirement Analysis Document;
- **DG:** Design Goal;
- **UC:** Use Case.
- **SU:** Start Up.
- **SD:** Shut Down.

- **FA**: Failure.
- **PR:** Preparatore
- **CL:** Cliente
- **AD:** Admin
- **UG:** Utente Generico
- **SDD**: System Design Document
- **CTD**:

1.4 Riferimenti

Libro: -- Object-Oriented Software Engineering (Using UML, Patterns, and Java) Third Edition
Autori: -- Bernd Bruegge & Allen H. Dutoit
RAD -- 2021_RAD_C7_FitDiary_V1.0

1.5 Panoramica del documento

Il presente documento di System Design consta di quattro sezioni:

Introduzione: Viene descritto in generale lo scopo del sistema, gli obiettivi di design che il sistema propone di raggiungere.
Architettura software corrente: Viene descritto lo stato attuale dell'architettura del software già presente.
Architettura software proposta: Viene descritto come il sistema sarà definito e partizionato in sottosistemi, il loro mapping Hardware/Software, la gestione dei dati persistenti. Verranno poi presentate la struttura dei singoli sottosistemi e le boundary conditions riguardanti l'intero sistema.
Glossario: Contiene la lista dei termini usati nel documento con annessa spiegazione.

2. Architettura del Sistema Corrente

Al momento, non esiste alcun software che condensi l'interezza delle funzionalità di FitDiary in un unico servizio. Il mercato delle possibili alternative a questo software è pertanto incredibilmente frammentato e non esiste una reale architettura a cui è possibile confrontare in maniera ragionevole il sistema.

3. Architettura del Sistema Proposto

3.1 Panoramica

La piattaforma "FitDiary" è un'applicazione distribuita che interagisce con gli utenti mediante un'interfaccia web e gestisce la persistenza dei dati mediante un database relazionale.
Il sistema proposto è basato su uno stile architetturale Client-Server.
Il motivo della presente scelta è che tale architettura è perfetta per lo sviluppo di web application come il nostro sistema, poiché separa la logica di business dalla logica di presentazione, migliorando:

- Prestazioni
- Manutenzione
- Leggibilità
- Riuso
- Portabilità
- Scalabilità

Per la logica di business saranno usati:

- Framework SpringBoot (Java)
- PostgreSQL(Database)

Per la logica di presentazione saranno usati:

- HTML5
- CSS3
- JavaScript (ReactJS)

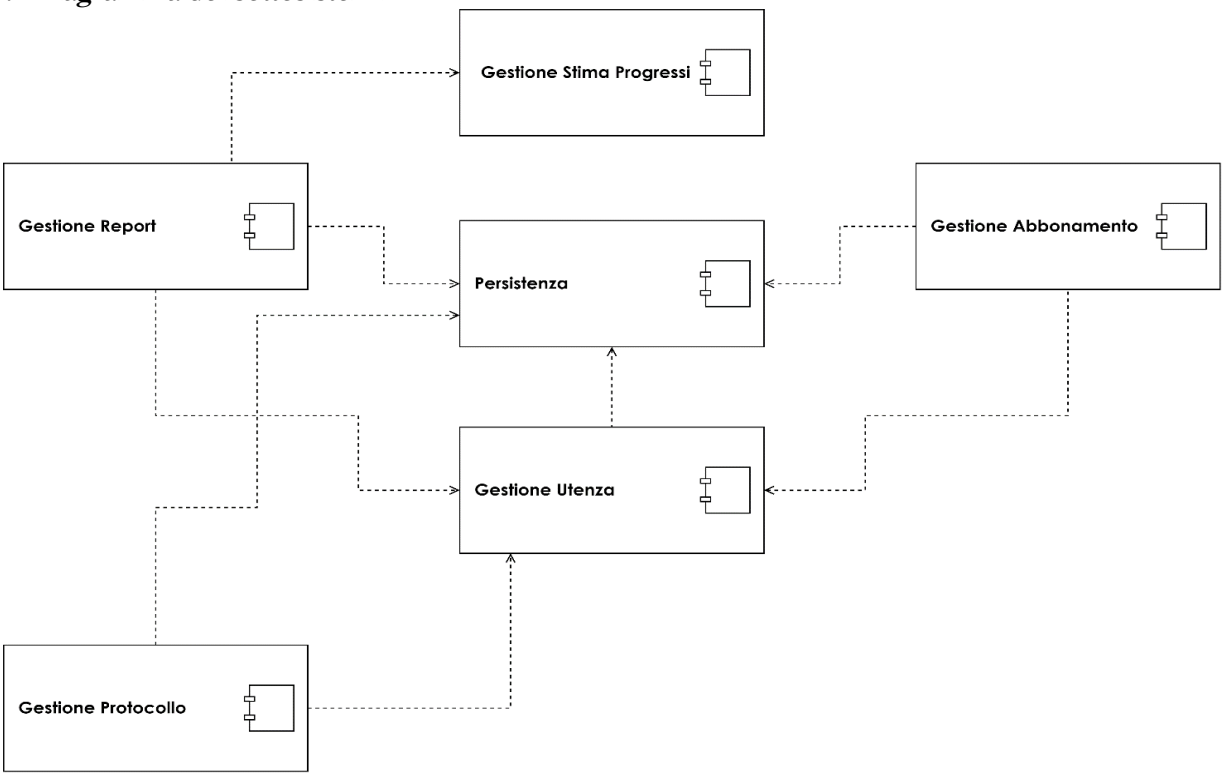
3.2 Decomposizione in sottosistemi

In modo da rispecchiare le funzionalità offerte dal sistema, e l'architettura proposta, il server addetto al frontend contiene le GUI del sistema mentre il server addetto al backend contiene tutte le funzionalità divise nei livelli di modello e controllo.

Il sistema viene quindi suddiviso nei seguenti sottosistemi:

- **Gestione utenza**: si occupa di gestire le funzioni di login, logout, modifica delle impostazioni utente, l'inserimento e la modifica dei dati personali, la registrazione del preparatore ed inoltre dell'eliminazione degli utenti da parte dell'amministratore.
- **Gestione protocollo**: è responsabile dell'elaborazione, dell'inserimento e della modifica dei protocolli.
- **Gestione abbonamento**: si occupa del canale di comunicazione con il sistema esterno "Stripe" per gestire i pagamenti e le notifiche di abbonamenti in scadenza.
- **Gestione report:** è responsabile della modifica e dell'inserimento dei report.
- **Gestione stima progressi**: è responsabile della stima dei progressi e del calcolo degli stessi tramite un modulo di intelligenza artificiale.
- **Persistenza**: Si occupa della gestione della persistenza tramite l'ausilio di un database.

3.2.1 Diagramma dei sottosistemi



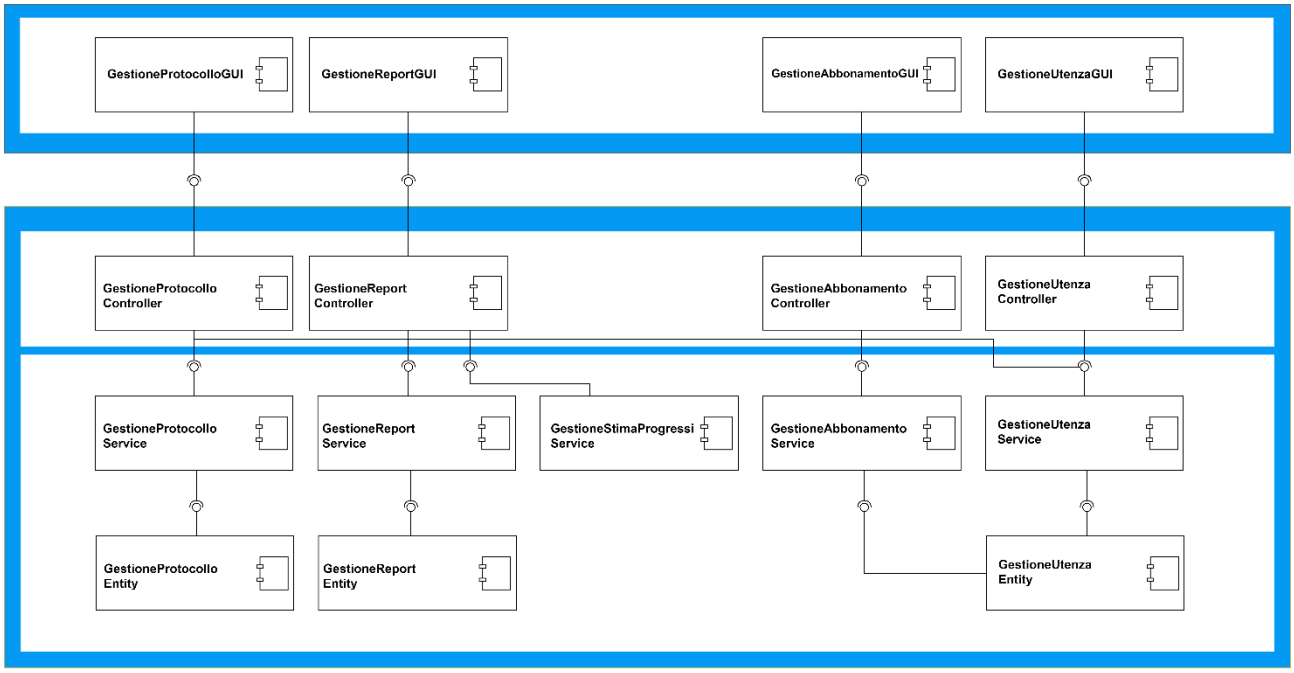
Alcuni sottosistemi saranno gestiti da componenti COTS (Commercial off the shelf), di seguito un elenco:

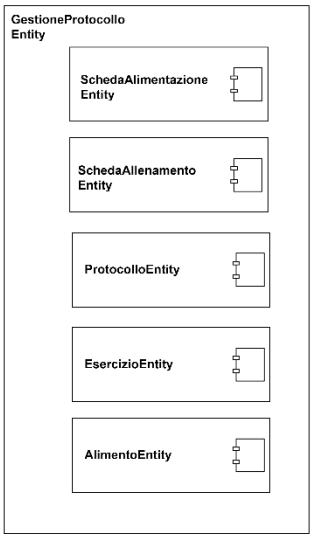
- Mediante Spring Data JPA, il sottosistema "Persistenza" gestirà le informazioni all'interno di un DBMS relazionale su sistema di Database As A Service .
- Stripe per la gestione dei pagamenti degli abbonamenti lato Preparatore. La comunicazione con questo COTS avverrà attraverso l'utilizzo delle librerie fornite dallo stesso Stripe, sia per la presentazione del componente di pagamento (chiamato "CardElement"), sia per la logica da associare all'acquisto dell'abbonamento da parte di un preparatore. Inoltre, mediante un meccanismo basato su WebHook, sarà lo stesso Stripe ad avvisare il nostro sistema per alcuni eventi come un pagamento terminato con successo oppure un mancato rinnovo.

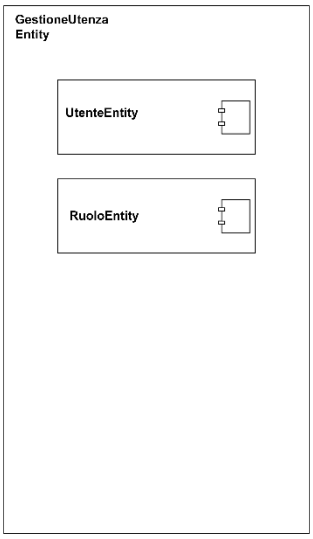
Di seguito una vista dettagliata di ciascun sottosistema evidenziando le componenti principali:

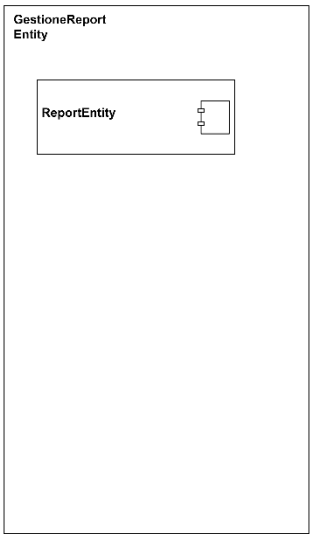
- **GUI**: Graphic User Interface, che contiene le varie view che saranno renderizzate per creare le pagine web da mostrare al cliente (Frontend).
- **Controller**: si occupa della logica per il controllo del sistema (Backend).
- **Service**: si occupa della logica di business (Backend).
- **Entity:** rappresenta le informazioni mappate all'interno del database (Backend).

3.2.2 Diagramma Architetturale







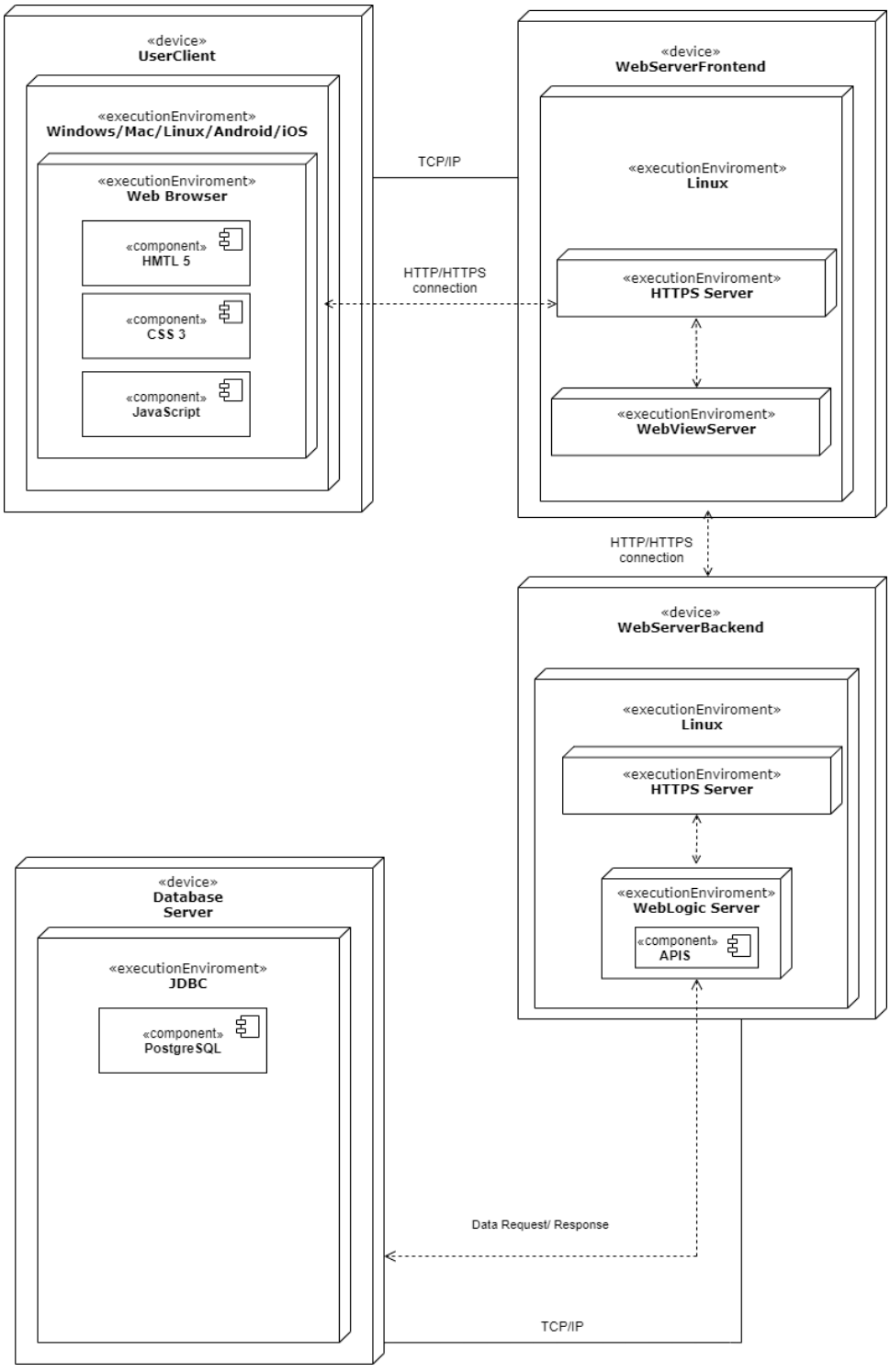


GestioneReportController è collegato a GestioneStimaProgressiService per consentire il calcolo della stima al momento dell'inserimento di un nuovo Report.

3.3 Mapping Hardware/Software

Il sistema utilizzerà una struttura hardware costituita da due server denominati ServerFrontend e ServerBackend. Il ServerBackend risponderà alle richieste del ServerFrontend mentre il ServerFrontend risponderà alle richieste inviate tramite interazione del client. Il client è rappresentato da una qualsiasi macchina con una connessione Internet e un web browser installato su di esso, in modo da potersi connettere al ServerFrontend per interagire col sistema. Il ServerFrontend contiene la Web Application utilizzabile dagli utenti. Invece, il ServerBackend è rappresentato da una qualsiasi macchina con una connessione Internet, dove è presente il DBMS PostgreSQL ed è capace di immagazzinare una certa quantità di dati. Esso, inoltre, gestisce la logica e i dati persistenti contenuti nel database. La comunicazione tra il client e il ServerBackend è basata sul protocollo HTTPS a cui il client invierà delle richieste e il server provvederà alle risposte.

3.3.1 Deployment Diagram

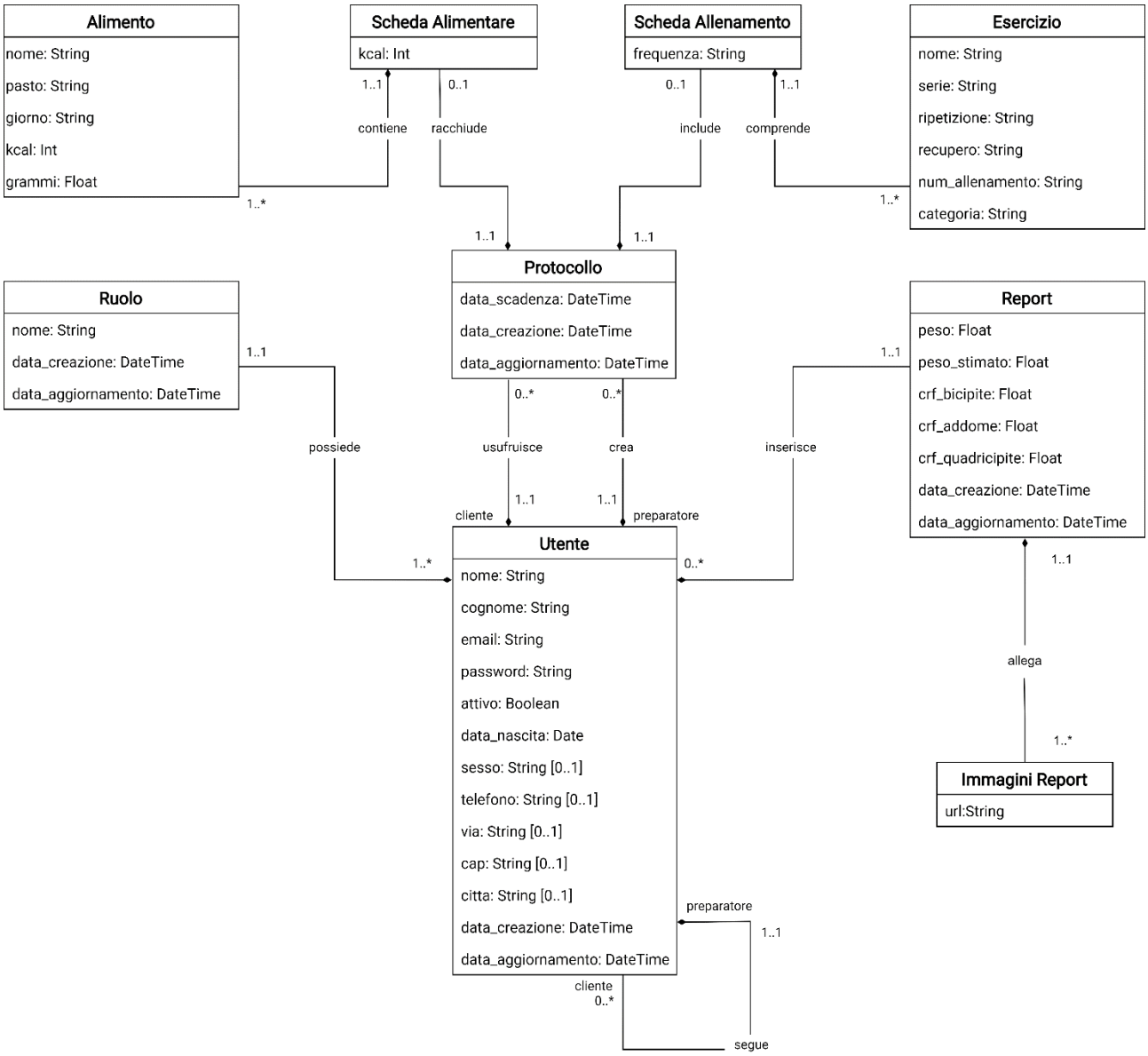


3.4 Gestione Dati Persistenti

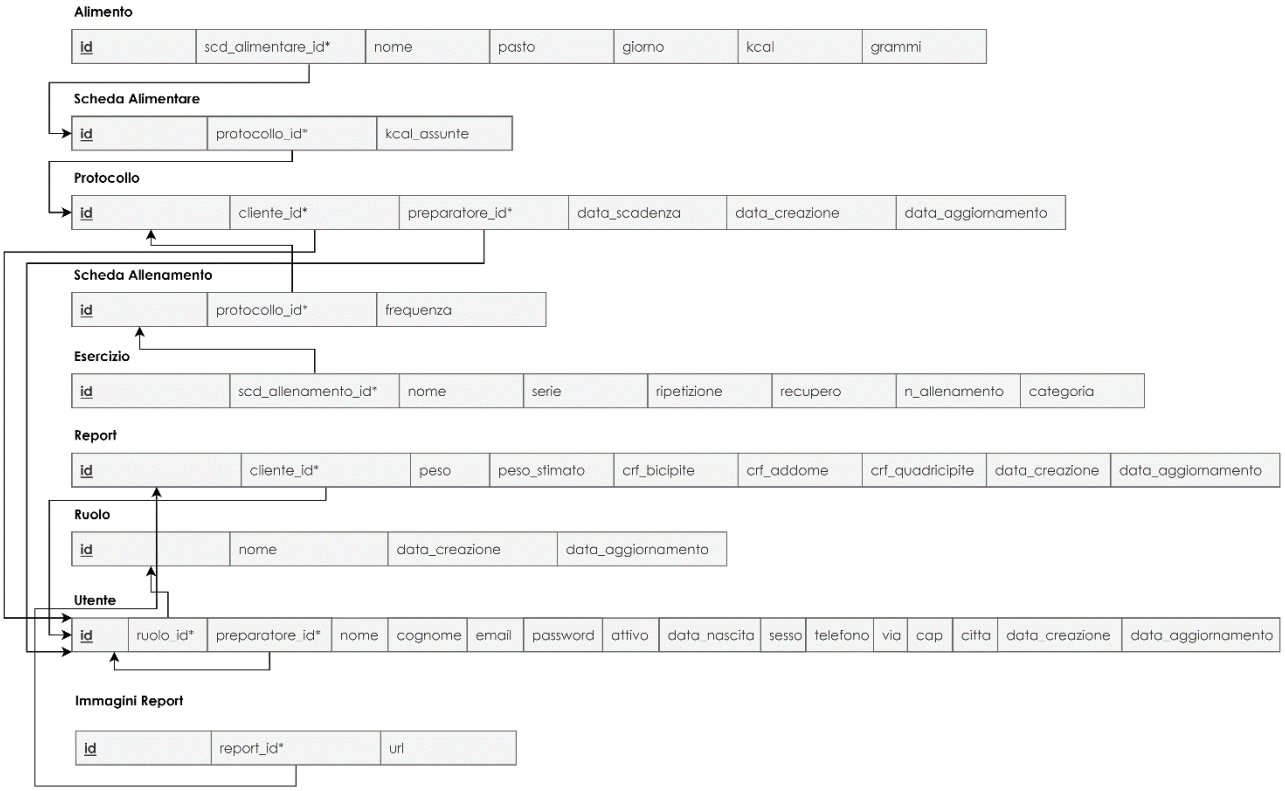
Per la gestione dei dati persistenti si è preferito usare un database relazionale, questa scelta è stata presa considerando l'accesso multiplo ai dati dei vari utenti.
La persistenza delle immagini avviene tramite file sia per la loro dimensione elevata sia per evitarne la perdita di qualità.
Durante la fase di system design, sono state prese delle decisioni:

- Inserire le Entity "Alimento" ed "Esercizio" per ridurre la complessità delle Entity "Scheda Allenamento" e "Scheda Alimentare" e dare maggiore riusabilità.
- Le Entity "Admin", "Preparatore" e "Cliente" sono state accorpate nell'Entity "Utente" per diminuire la complessità della persistenza dei dati.
- Inserire l'Entity "Ruolo" che servirà a memorizzare i possibili ruoli nel sistema, essi verranno poi associati agli utenti.
- Inserire l'attributo ridondante "kcal" nell'entità "Scheda Alimentare" essendo un valore richiesto frequentemente, per evitarne il ricalcolo.
- La cardinalità della relazione che lega un Cliente con un Preparatore è stata modificata in 1 a N poiché attualmente non è prevista la possibilità per un Cliente di essere seguito da più preparatori contemporaneamente ed inoltre, non sono stati identificati requisiti che richiedono uno storico clienti per ogni singolo preparatore.

3.4.1 Entity Class Diagram Ristrutturato



3.4.2 Schema Logico



3.4.3 Dizionario dei dati

3.4.3.1 Report

Nome Entità	Report		
Descrizione	Contiene dati relativi ad un Report		
Nome Campo	Tipo	Vincolo di chiave	Altri vincoli
id	Long	PRIMARY KEY	NOT NULL
cliente_id	Long	FOREIGN KEY (Utente)	NOT NULL
peso	Float		NOT NULL
peso_stimato	Float		NOT NULL
crf_bicipite	Float		NOT NULL
crf_addome	Float		NOT NULL
crf_quadricipite	Float		NOT NULL
data_creazione	DateTime		NOT NULL
data_aggiornamento	DateTime		NOT NULL

3.4.3.2 Utente

Nome Entità	Utente		
Descrizione	Contiene dati relativi ad un Utente		
Nome Campo	Tipo	Vincolo di chiave	Altri vincoli
id	Long	PRIMARY KEY	NOT NULL
ruolo_id	Long	FOREIGN KEY (Ruolo)	NOT NULL
preparatore_id	Long	FOREIGN KEY (Utente)	
nome	Varchar(50)		NOT NULL
cognome	Varchar(50)		NOT NULL
email	Varchar(50)		NOT NULL
password	Varchar(255)		NOT NULL
attivo	Boolean		NOT NULL
data_nascita	Date		NOT NULL
sesso	Varchar(1)		
telefono	Varchar(10)		
via	Varchar(50)		
cap	Varchar(5)		
città	Varchar(20)		
data_creazione	DateTime		NOT NULL
data_aggiornamento	DateTime		NOT NULL

3.4.3.3 Protocollo

Nome Entità	Protocollo		
Descrizione	Contiene le informazioni relative ad un Protocollo		
Nome Campo	Tipo	Vincolo di chiave	Altri vincoli
id	Long	PRIMARY KEY	NOT NULL
cliente_id	Long	FOREIGN KEY (Utente)	NOT NULL
preparatore_id	Long	FOREIGN KEY (Utente)	NOT NULL
data_scadenza	Date		NOT NULL
data_creazione	DateTime		NOT NULL
data_aggiornamento	DateTime		NOT NULL

3.4.3.4 Scheda alimentare

Nome Entità	Scheda alimentare		
Descrizione	Contiene le informazioni relative ad una Scheda alimentare		
Nome Campo	Tipo	Vincolo di chiave	Altri vincoli
id	Long	PRIMARY KEY	NOT NULL
protocollo_id	Long	FOREIGN KEY (Protocollo)	NOT NULL
kcal_assunte	Float		NOT NULL

3.4.3.5 Alimento

Nome Entità	Alimento		
Descrizione	Contiene le informazioni relative ad un Alimento		
Nome Campo	Tipo	Vincolo di chiave	Altri vincoli
id	Long	PRIMARY KEY	NOT NULL
scheda_alimentare_id	Long	FOREIGN KEY (Scheda Alimentare)	NOT NULL
nome	Varchar(50)		NOT NULL
pasto	Varchar(20)		NOT NULL
giorno	Varchar(10)		NOT NULL
kcal	Float		NOT NULL
grammi	Varchar(20)		NOT NULL

3.4.3.6 Scheda allenamento

Nome Entità	Scheda allenamento		
Descrizione	Contiene le informazioni relative ad una scheda allenamento		
Nome Campo	Tipo	Vincolo di chiave	Altri vincoli
id	Long	PRIMARY KEY	NOT NULL
protocollo_id	Long	FOREIGN KEY (Protocollo)	NOT NULL
frequenza	Varchar(50)		NOT NULL

3.4.3.7 Esercizio

Nome Entità	Esercizio		
Descrizione	Contiene le informazioni relative ad un Esercizio		
Nome Campo	Tipo	Vincolo di chiave	Altri vincoli
id	Long	PRIMARY KEY	NOT NULL
scheda_allenamento_id	Long	FOREIGN KEY (Scheda Allenamento)	NOT NULL
nome	Varchar(50)		NOT NULL
serie	Varchar(20)		NOT NULL
ripetizioni	Varchar(20)		NOT NULL
recupero	Varchar(10)		NOT NULL
numero_allenamento	Varchar(10)		NOT NULL
categoria	Varchar(20)		NOT NULL

3.4.3.8 Ruolo

Nome Entità	Ruolo		
Descrizione	Contiene le informazioni relative ai possibili ruoli all'interno del sistema		
Nome Campo	Tipo	Vincolo di chiave	Altri vincoli
id	Long	PRIMARY KEY	NOT NULL
nome	Varchar(20)		NOT NULL
data_creazione	DateTime		NOT NULL
data_aggiornamento	DateTime		NOT NULL

3.4.3.9 ImmaginiReport

Nome Entità	Ruolo		
Descrizione	Contiene le informazioni relative ai possibili ruoli all'interno del sistema		
Nome Campo	Tipo	Vincolo di chiave	Altri vincoli
id	Long	PRIMARY KEY	NOT NULL
report_id	Long	FOREIGN KEY(Report)	NOT NULL
url	Varchar(255)		NOT NULL

3.5 Controllo degli Accessi e Sicurezza

Attori / Oggetti	Cliente	Preparatore	Admin
Gestione utenza	Login, Logout, VisualizzazioneProfilo, InserimentoDatiPersonali, Notifica ConfermaRegistrazione, ModificaDatiPersonali	Login, Logout, VisualizzazioneProfilo, Visualizzazione ListaClienti, CreazioneNuovoCliente, Registrazione, Visualizzazione ProfiloCliente, Disattivazione Cliente, ModificaDatiPersonali, DisiscrizionePrepartore	Login, Logout, VisualizzazioneProfilo, VisualizzazioneListaUtenti, Eliminazione Utente
Gestione protocollo	Visualizzazione SchedaAlimentazione, Visualizzazione SchedaAllenamento, Visualizzazione StoricoProtocolli, Scaricamento	CreazioneProtocollo, Caricamento SchedaAlimentazione, Caricamento SchedaAllenamento, ModificaScheda Alimentazione, ModificaScheda	

	SchedaAlimentare, Scaricamento SchedaAllenamento	Allenamento, Visualizzazione SchedaAlimentazione, Visualizzazione SchedaAllenamento, Visualizzazione Protocollo, Visualizzazione StoricoProtocolli, Scaricamento SchedaAlimentare, Scaricamento SchedaAllenamento	

Attori / **Oggetti**	**Cliente**	**Preparatore**	**Admin**
Gestione abbonamento		Acquisto Abbonamento	
Gestione report	Inserimento ReportProtocollo, Visualizzazione ReportProtocollo, Visualizzazione StoricoProgressiFisici	Visualizzazione StoricoProgressiFisici, Visualizzazione ReportProtocollo	
Gestione stima progressi	Visualizzazione StimaProgressi	Visualizzazione StimaProgressi	

3.6 Controllo Flusso Globale del Sistema

Il sistema FitDiary fornisce una funzionalità che richiede una continua interazione da parte dell'utente, per questo motivo abbiamo adottato un controllo del flusso globale del sistema di tipo event-driven, ovvero guidato dagli eventi. Il flusso globale di tipo event-driven permette di rappresentare un sistema interattivo per cui ogni funzionalità viene avviata in seguito ad un comando impartito dall'utente tramite l'uso di un'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 il quale verrà gestito dal suo handler, esso 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.

3.7 Boundary Conditions

Nella seguente sezione vengono riportate le condizioni limite del sistema, come l'avvio, la terminazione e i fallimenti del sistema.

In particolare, poiché l'architettura del sistema prevede l'utilizzo di due server che l'amministratore gestisce in modo molto simile, ci limitiamo a descrivere il comportamento relativo al WebServerBackend, specificando le differenze qualora necessario.

3.7.1 Start up

Identificativo	UC_SU	**Data**	01/12/2021
Nome	Avvio del server	**Versione**	1.0
		Autori	Davide La Gamba, Ilaria De Sio
Descrizione	Lo UC fornisce la funzionalità all'amministratore di avviare il server.		
Attore Principale	**Amministratore** L'amministratore ha interesse ad avviare il server.		
Attori Secondari	NA		
Entry Condition	L'amministratore accede al server.		
Exit Condition On Success	Il server viene avviato con successo.		
Exit Condition On Failure	Il server non viene avviato con successo.		
Rilevanza/User Priority	Elevata		
Frequenza Stimata	1 usi/mese		
Extention Point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE / MAIN SCENARIO			

1	Amministratore:	Usa il 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 mostra all'amministratore la sua area riservata.
Scenario / Flusso eventi Alternativo: Il server non era stato arrestato correttamente.		
4.1	Sistema:	Notifica l'amministratore del precedente errore nell'arresto del server e lo avvia.
Scenario / Flusso eventi di Errore: La connessione al database non è stata inizializzata correttamente		
2.1	Sistema:	Notifica l'amministratore dell'errore e non procede all'avvio del server.
Scenario / Flusso eventi di Errore: I dati persistenti non risultano integri		
3.1	Sistema:	Notifica l'amministratore di errori sull'integrità dei dati persistenti nel database e non procede all'avvio del server.
3.2	Amministratore:	Accede al database e corregge gli errori sui dati persistenti.
3.3	Amministratore:	Riesegue lo step 1.
NOTE		
	NA	
Special Requirements		
	NA	

Diversamente, il WebServerFrontend non inizializza alcuna connessione con il database, e quindi non presenta i passi 2 e 3 con i relativi flussi eventi di errore/alternativi.

3.7.2 Shut down

Identificativo	UC_SD	**Data**	01/12/2021
Nome	Arresto del server	**Versione**	1.0
		Autori	Davide La Gamba, Ilaria De Sio
Descrizione	Lo UC fornisce la funzionalità all'amministratore di arrestare il server.		
Attore Principale	**Amministratore** L'amministratore ha interesse ad arrestare il server.		
Attori Secondari	NA		
Entry Condition	Il server è correttamente avviato e l'amministratore ha effettuato l'accesso al sistema.		
Exit Condition On Success	Il server viene arrestato con successo.		
Exit Condition On Failure	Il server non viene arrestato con successo.		
Rilevanza/User Priority	Elevata		
Frequenza Stimata	1 usi/mese		
Extention Point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE / MAIN SCENARIO			
1	Amministratore:	Usa il comando per arrestare il server	
2	Sistema:	Effettua il salvataggio dei dati persistenti.	
3	Sistema:	Chiude le connessioni attive ed arresta il server.	
Scenario / Flusso eventi di Errore: Il salvataggio dei dati persistenti non è andato a buon fine			

2.1	Sistema:	Notifica l'amministratore dell'errore e non procede all'arresto del server.
NOTE		
	NA	
Special Requirements		
	NA	

Il WebServerFrontend, non avendo legami con il database, non presenta il passo 2 con il relativo flusso di eventi di errore.

3.7.3 Failures

Identificativo	UC_FA	**Data**	01/12/2021
Nome	Errore di accesso ai dati persistenti	**Versione**	1.0
		Autori	Davide La Gamba, Ilaria De Sio
Descrizione	Lo UC descrive il comportamento del sistema al verificarsi di un errore di accesso ai dati persistenti.		
Attore Principale	**Amministratore** L'amministratore ha interesse a risolvere la condizione di fallimento.		
Attori Secondari	NA		
Entry Condition	Il sistema ha riscontrato un errore nell'accesso ai dati persistenti e notifica l'amministratore.		
Exit Condition On Success	Il sistema riprende il corretto funzionamento.		
Exit Condition On Failure	Il server riprende il corretto funzionamento.		
Rilevanza/User Priority	Elevata		
Frequenza Stimata	1 usi/mese		
Extention Point	NA		
Generalization of	NA		

FLUSSO DI EVENTI PRINCIPALE / MAIN SCENARIO		
1	Amministratore:	Riceve la notifica di errore dal sistema.
2	Sistema:	Non prende in carico ulteriori richieste.
3	Amministratore:	Arresta forzatamente il sistema tramite l'apposito comando.

4	Amministratore:	Risolve gli errori sui dati persistenti.
5	Amministratore:	Include(UC_SU)
NOTE		
	NA	
Special Requirements		
	NA	

3.7.4 Gestione dei fallimenti

- Per sopperire alla possibilità di errori riguardanti i dati persistenti, il sistema prevede di effettuare una procedura di backup automatica periodica dei dati persistenti, sotto forma di codice SQL. In tal modo, è garantito il successivo ripristino del database.
- Il sistema non prevede specifiche misure per sopperire a fallimenti critici dell'hardware. In tal caso, l'amministratore deve necessariamente riavviare il sistema occupandosi degli errori manualmente.
- In caso di interruzione imprevista dell'alimentazione, il sistema non prevede meccanismi di salvataggio dei dati. Al ripristino, l'amministratore si limiterà a caricare l'ultimo backup automatico effettuato dal sistema.

Riguardo i fallimenti del WebServerFrontend, in caso di errore in una richiesta per un servizio del WebServerBackend, il sistema mostra una notifica di errore all'utente.

4. Servizi dei Sottosistemi

4.1 Gestione Utenza

Servizio	Descrizione	Interfaccia
Login	Questa funzionalità permette di effettuare l'accesso al sistema tramite le proprie credenziali per sfruttare tutte le funzionalità offerte.	Gestione UtenzaService
Registrazione	Questa funzionalità permette al preparatore di registrarsi al sistema.	Gestione UtenzaService
Visualizza Profilo	Questa funzionalità permette di visualizzare il proprio profilo.	Gestione UtenzaService
Modifica Dati Personali	Questa funzionalità permette di modificare i propri dati personali.	Gestione UtenzaService
Inserimento Dati Personali	Questa funzionalità permette al cliente di inserire i propri dati personali.	Gestione UtenzaService
Notifica Conferma Registrazione	Questa funzionalità permette di ricevere una notifica dell'avvenuta registrazione.	Gestione UtenzaService
Visualizza Lista Clienti	Questa funzionalità permette al preparatore di visualizzare la lista dei suoi clienti e di filtrarla.	Gestione UtenzaService
Creazione Nuovo Cliente	Questa funzionalità permette al preparatore di inserire un nuovo cliente.	Gestione UtenzaService

Servizio	Descrizione	Interfaccia

Visualizza Profilo Cliente	Questa funzionalità permette al preparatore di visualizzare il profilo di un cliente.	Gestione UtenzaService
Disattivazione Cliente	Questa funzionalità permette al preparatore di disattivare il profilo di un cliente.	Gestione UtenzaService
Visualizza Lista Utenti	Questa funzionalità permette all' admin di visualizzare la lista degli utenti e di filtrarla.	Gestione UtenzaService
Eliminazione Utente	Questa funzionalità permette all' admin di eliminare un utente.	Gestione UtenzaService
Disiscrizione Preparatore	Questa funzionalità permette al preparatore di disiscriversi dalla piattaforma.	Gestione UtenzaService

4.2 Gestione Protocollo

Servizio	Descrizione	Interfaccia
Creazione Protocollo	Questa funzionalità permette al preparatore di creare un protocollo per un cliente.	Gestione ProtocolloService
Caricamento Scheda Alimentazione	Questa funzionalità permette al preparatore di caricare all'interno di un protocollo la scheda alimentare.	Gestione ProtocolloService
Caricamento Scheda Allenamento	Questa funzionalità permette al preparatore di caricare all'interno di un protocollo la scheda allenamento.	Gestione ProtocolloService
Modificare Scheda Alimentazione	Questa funzionalità permette al preparatore di modificare all'interno di un protocollo la scheda alimentare.	Gestione ProtocolloService
Modificare Scheda Allenamento	Questa funzionalità permette al preparatore di modificare all'interno di un protocollo la scheda allenamento.	Gestione ProtocolloService
Visualizzazione Scheda Alimentazione	Questa funzionalità permette di visualizzare la scheda alimentare di un protocollo.	Gestione ProtocolloService
Visualizzazione Scheda Allenamento	Questa funzionalità permette di visualizzare la scheda di allenamento di un protocollo.	Gestione ProtocolloService
Visualizzazione Protocollo	Questa funzionalità permette di visualizzare il protocollo.	Gestione ProtocolloService
Visualizzazione Storico Protocolli	Questa funzionalità permette di visualizzare lo storico dei protocolli.	Gestione ProtocolloService

Servizio	Descrizione	Interfaccia
Scaricamento Scheda Alimentare	Questa funzionalità permette di scaricare la scheda alimentare.	Gestione ProtocolloService
Scaricamento Scheda Allenamento	Questa funzionalità permette di scaricare la scheda di allenamento.	Gestione ProtocolloService

4.3 Gestione Abbonamento

Servizio	Descrizione	Interfaccia
Acquisto Abbonamento	Questa funzionalità permette al preparatore di acquistare un abbonamento per poter usufruire della piattaforma.	Gestione AbbonamentoService
Conferma Pagamento	Questa funzionalità permette di comunicare l'avvenuto pagamento da parte del preparatore.	Gestione AbbonamentoService
Disattiva Abbonamento	Questa funzionalità permette di disattivare l'abbonamento al preparatore.	Gestione AbbonamentoService

4.4 Gestione Report

Servizio	Descrizione	Interfaccia
Inserimento Report Protocollo	Questa funzionalità permette al cliente di inserire un nuovo report.	Gestione ReportService
Visualizzazione Report Protocollo	Questa funzionalità permette di visualizzare un report all'interno del protocollo.	Gestione ReportService
Visualizzazione Storico Progressi Fisici	Questa funzionalità permette di visualizzare lo storico dei progressi fisici del cliente.	Gestione ReportService

4.5 Gestione Stima Progressi

Servizio	Descrizione	Interfaccia
Generazione Stima Progressi	Questa funzionalità permette di generare la stima dei progressi di un cliente.	GestioneStima ProgressiService

5. Glossario

Sigla/Termine	Definizione
Preparatore	Il professionista che si occupa della preparazione fisica e/o alimentare del cliente.
Cliente	Colui che usufruisce del servizio fornito dal preparatore.
Admin	Rappresenta la figura amministrativa del sistema, in grado di cancellare e visualizzare gli utenti che utilizzano il sistema.
Protocollo	Documento prodotto da un preparatore per un cliente contenente scheda di allenamento e/o alimentazione. Per essere creato deve contenere almeno una scheda (di allenamento o di alimentazione). Modificare un protocollo consiste nell'aggiungere o modificare una scheda.
Report	Rappresenta i progressi di un cliente, raggiunti a fine protocollo espressi in termini di dati numerici come peso, circonferenze del corpo, eventualmente foto che il cliente può allegare e contiene la stima dei progressi prevista per quel report.
Stima Progressi	Previsione del peso ad un mese di distanza dalla consegna del report.