

S.D.D.

System Design Document

Università degli studi di Salerno

Corso di Ingegneria del Software 2017-2018



TEAM

Nome Cognome	Matricola
Carmine Picariello	0512103604
Prisco Luigi	0512103602
Luca D'Avino	0512103496

Sommario

1. Introduzione

Oggi giorno dovrebbe essere un obiettivo lavorare sulla propria forma fisica non solo con una giusta alimentazione, che esclude naturalmente cibi super-calorici che compromettono il nostro stato di salute provocandone patologie molto frequenti come ad esempio l'obesità, ma coordinando ad essa l'attività fisica per modellare il nostro corpo dal punto di vista estetico. Dato che la tecnologia ormai è entrata a far parte della quotidianità, abbiamo deciso di unire questi due campi per creare la nostra web app: MAIGIM !!

App creata per facilitare l'attività in palestra sia per l'istruttore che per il cliente poiché il primo ha la possibilità di modificare o aggiornare la scheda dell'utente, sulla piattaforma a lui dedicata, e il secondo accedendo alla sua area , tramite un login, potrà visualizzare il programma e svolgere in piena autonomia gli esercizi giornalieri.

2. Obiettivi

- Ridurre lo spreco di carta eliminando l'utilizzo delle schede stampate .

- La scheda cartacea è soggetta ad usura e spesso viene persa o dimenticata , con MaiGim la scheda sarà sempre con te accessibile da qualsiasi dispositivo..
- Fornire a portata di mano una pratica interfaccia user-friendly che permetterà di programmare il proprio allenamento.
- Fornire al personal trainer un'ottima alternativa per poter modificare le schede di ogni utente in tempo reale.
- L'utente riceverà una notifica ogni qual volta il personal trainer modificherà la propria scheda.

Criteri per il Successo del Sistema :

- Facilità d'uso della web app.
- L'accesso alle schede di ogni utente è visibile solo dal personal trainer.
- Ogni utente ha la propria e scheda e visualizza solo quella.
- Utilizzo di un database sicuro
- Possibilità di recupero della password.

3. Design Goal

Il sistema pensato cerca di rispettare tutti i criteri sotto stabiliti.

3.1 Criteri di Performance

Tempo di risposta.

Attraverso una accurata analisi siamo giunti alla conclusione che il tempo di risposta deve essere inferiore ai 2-3 secondi. I tempi di risposta alla sottomissione di dati saranno limitati in quanto i controlli sono effettuati dal client e l'utente avrà subito la risposta di errore.

Throughput

Le prestazioni del sistema hanno come unica limitazione l'utilizzo di un database con linguaggio NoSql utilizzato come repository centrale e del tipo di web server utilizzato per la gestione del portale (per avere maggiori prestazioni bisogna investire maggiori risorse nel tipo di web server utilizzato).

Memoria

Il sistema ha bisogno , a regime , ha bisogno dello spazio per la memorizzazione dei dati relativi agli utenti e alle loro schede. Come sistema di memorizzazione viene utilizzato un database NoSql.

3.2 Criteri di Mantenimento

Estendibilità

La progettazione sarà condotta in modo da agevolare la facile introduzione di nuove funzionalità tramite linguaggi Angular/HTML/CSS.

Leggibilità

Il codice sarà ben strutturato e documentato in modo da permettere facili interventi su di esso. Per documentazione si intende quella fornita di ausilio per l'utente.

Modificabilità

Sarà possibile intervenire sul codice per correggere bugs o implementare nuove funzionalità

Tracciabilità

In ogni fase di sviluppo, fino alla codifica, è possibile risalire facilmente ai requisiti funzionali e viceversa.

3.3 Criteri dell'utente finale

Usability

L'utilizzo del sistema non richiede particolari conoscenze tecniche; il sistema riduce al minimo l'errore utente rendendo l'esperienza di facile apprendimento tramite interfacce intuitive e auto esplicative. Ogni pulsante avrà un'etichetta che ne specifica la funzione, le caselle di testo presenteranno del testo illustrativo per aiutare l'utente nell'immissione dei dati.

Utility

Eventuali errori verranno spiegati all'utente tramite opportuni messaggi.

4. Definizioni, abbreviazioni e acronimi

MaiGim: è il nome della web app che permette agli utenti di avere sempre a portata di mano la propria scheda di allenamento digitale.

Angular: è un framework open source sviluppato da Google e dalla comunità di sviluppatori che ruota intorno al framework nato per affrontare le difficoltà nello sviluppo di applicazioni web.

Web App : indica generalmente un'applicazione accessibile via web per mezzo di un network , come ad esempio una Intranet oppure un sistema client-server.

User-Friendly : termine che indica la facilità d'uso del sistema per l'utente.

Homepage : pagina iniziale attraverso il quale l'utente accede al sistema.

Firebase : è una piattaforma web e mobile , acquisita da Google nel 2014 , offre servizi quale DBMS , Sicurezza , Autenticazione etc...

DBMS : Database Management System , acronimo che descrive il programma che permette di gestire facilmente un database.

DB: acronimo di Database.

5. Materiali di riferimento

Il materiale di riferimento utilizzato comprende:

- Slide Del Docente

Panoramica

Il documento si compone di una prima parte in cui vengono introdotti i design goals. Viene brevemente spiegata l'architettura corrente del software e quella a cui stiamo pensando per la definizione del nostro sistema. Segue la decomposizione del sistema in sottosistemi e l'individuazione dei servizi offerti da ogni sottosistema. Vengono individuati i dati persistenti necessari per il funzionamento di MaiGim e l'approccio per la gestione dei dati persistenti. Vengono individuate le categorie di utenti che possono usare il sistema e

viene definita la politica di accesso. Vengono disciplinate le condizioni di eccezione e i casi limite.

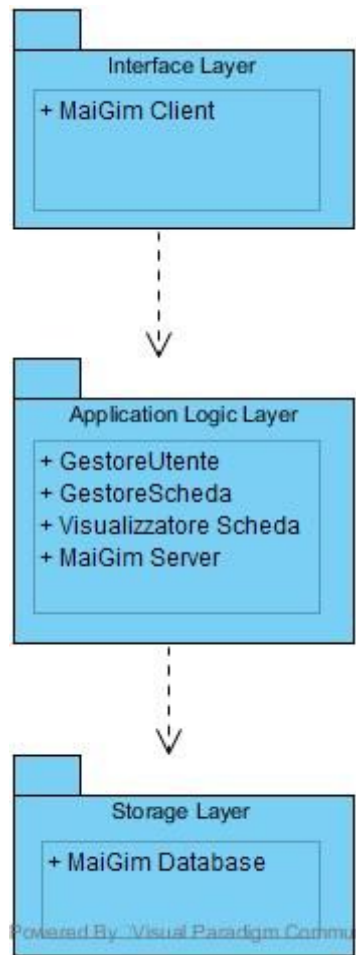
Architettura corrente del software

Ricercando sul web in data : 28/12/2017 abbiamo constatato che non esistono siti che permettono la gestione della scheda di allenamento come viene proposta dal nostro sistema. Cercando sullo store di Android ed Apple abbiamo constatato che esistono moltissime app simili , ma che non permettono la gestione delle schede come la nostra web-app.

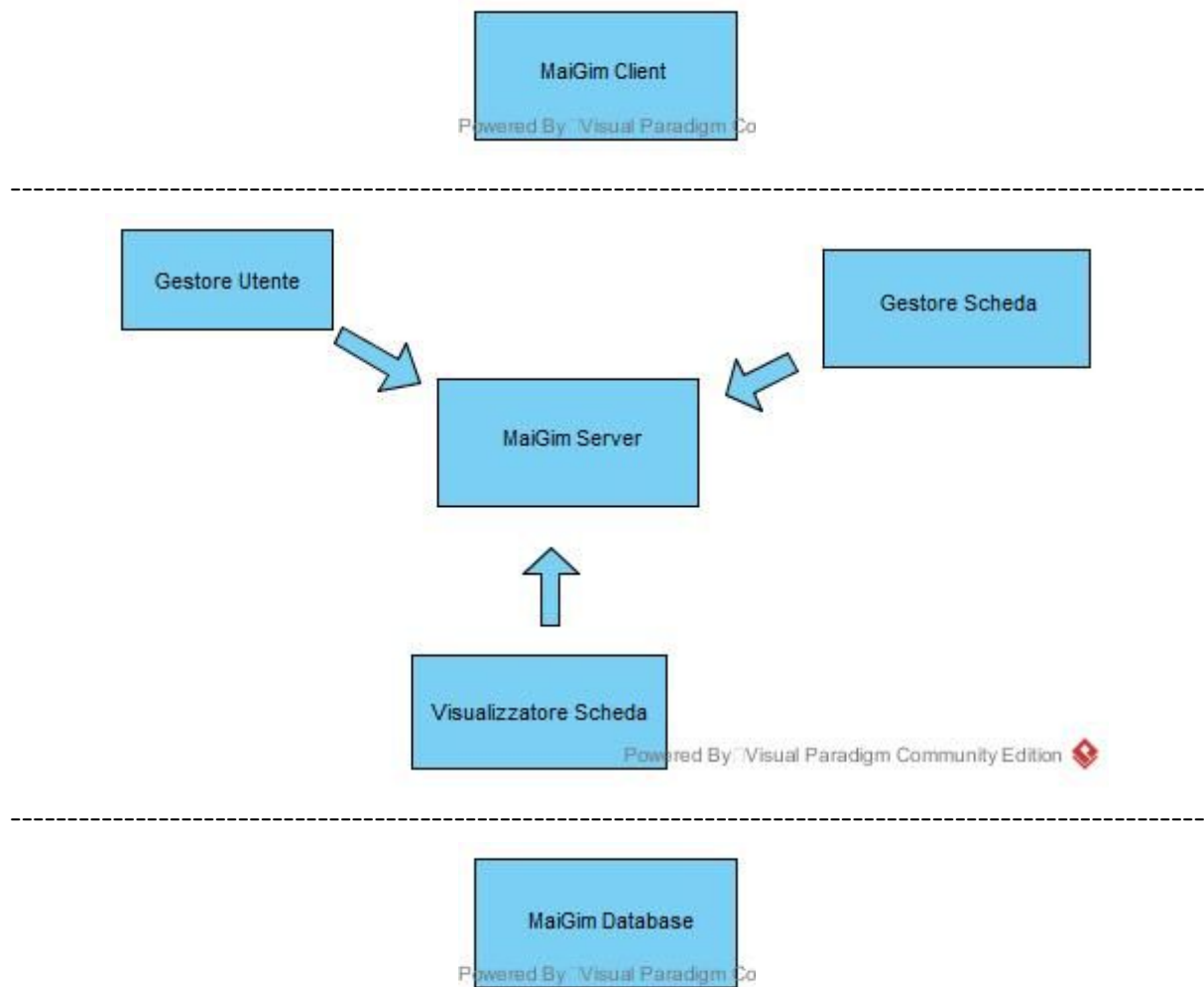
Architettura software proposta

MaiGim è costituita da una web-app , con architettura client-server. Il client richiede l'erogazione al server dei servizi che provvederà a fornirli in tempi di risposta accettabili. Per memorizzare i dati abbiamo optato per un database con linguaggio NoSql chiamato FireBase che garantirà sicurezza e velocità di risposta. Gli utenti non hanno modo di interagire fra loro , ognuno potrà consultare soltanto la propria scheda personale.

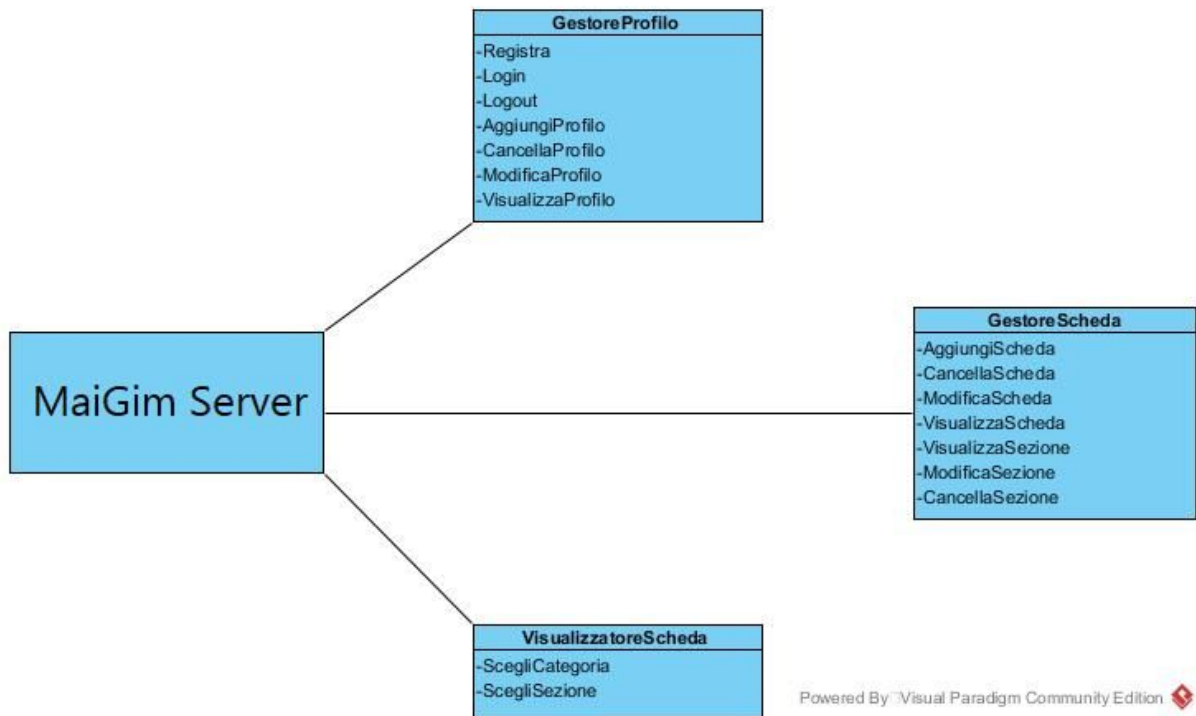
Decomposizione sistema in sottosistemi



Durante la fase di analisi abbiamo individuato una prima divisione logica del sistema in sottosistemi e individuato le relazioni tra di loro. La fase di System Design eredita dalla fase di analisi il seguente schema:

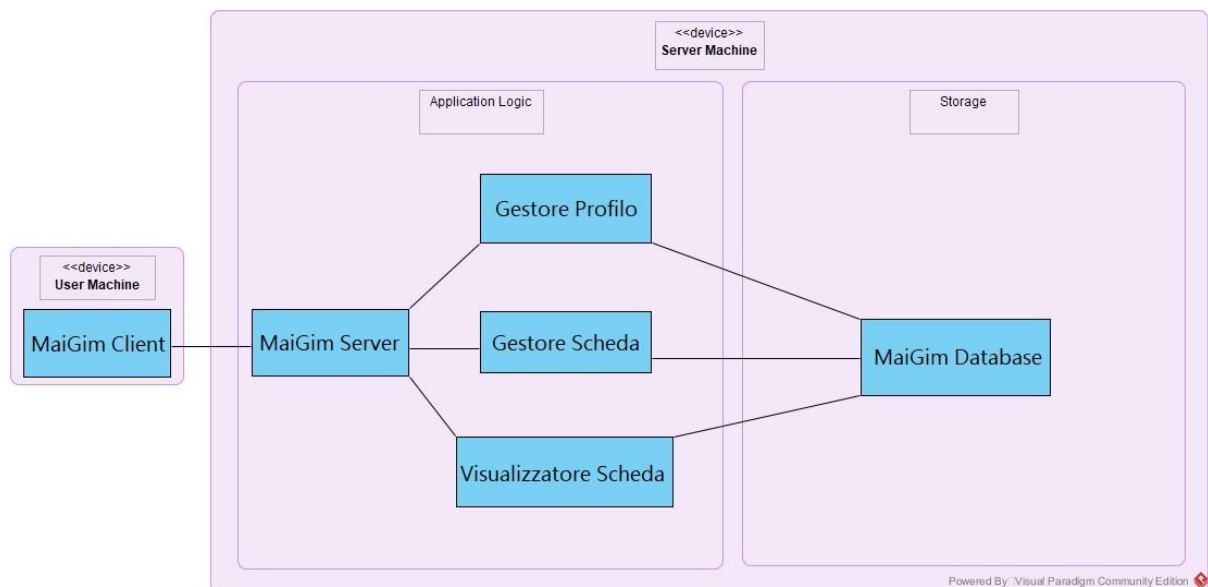


MaiGim è una componente logica che funge da tramite tra client e gestori che utilizzano il database. Il client , ovvero il browser , apre le pagine della web app e richiama questi gestori passando dati o ricevendoli , tutti questi insieme compongono il server.



Hardware e Software Mapping

MaiGim è un sistema basato su web app. Qui si può distinguere un sistema client-server nel quale , il client è il browser , mentre il server è costituito da funzioni che vengono richiamate e poi vi si trova il database. Per la realizzazione del sistema abbiamo usato il linguaggio di Angular 5 , che ha permesso di realizzare la web app e per la memorizzazione dei dati persistenti abbiamo utilizzato un database di tipo NoSql chiamato FireBase , scelto per la sua velocità di risposta e per la sicurezza che garantisce.



Gestione Dati Persistenti

Il sistema utilizza un database di tipo Real Time con linguaggio NoSql , che permette una gestione semplice , veloce e affidabile. Durante la fase di analisi sono emersi vari dati da dover memorizzare , fra i quali :

- Tabella Utente contenente gli utenti registrati al sistema.
- Tabella Scheda contenente le schede assegnate ad ogni utente.
- Tabella Esercizi contenente gli esercizi.

Questi dati devono essere memorizzati in modo permanente e resi accessibili in qualsiasi momento. Non è necessario tener conto degli accessi utente o di quante volte visualizzano la scheda. Come DBMS abbiamo optato per Firebase , nato nel 2011 dalla Firebase Inc. e acquisito nel 2014 da Google per il suo successo. All'interno sono memorizzate tre tabelle :

UTENTE	
Campi	Vincoli
Nickname	Vincolo di unicità , chiave della tabella

I dati da memorizzare nel Database vengono passati come Stringhe JSON , ogni volta che un utente viene registrato , il database crea una sottotabella con i vari dati al proprio interno e come nome assegna il nickname dell'utente.

Nickname	
Campo	Vincoli
Nome	
Cognome	
Nickname	
Email	Vincolo di unicità , chiave della tabella

Ognuna delle sottotabelle avrà memorizzati : Nome , Cognome , Nickname ed Email.

La seconda tabella è quella degli esercizi :

ESERCIZI	
Campi	Vincoli
Categoria	Vincolo di unicità , chiave della tabella

Il campo categoria consiste nel nome della sottotabella che conterrà tutti gli esercizi di quella determinata categoria

Categoria	
Campi	Vincoli
Nome	Vincolo di unicità , chiave della tabella
Categoria	

Altra tabella memorizzata è la tabella Scheda , essa è composta dai nickname degli utenti :

SCHEDA	
Campi	Vincoli
Scheda NomeUtente	

Ogni nome utente conterrà a sua volta : la sottotabella con il proprio nickname , la sottotabella con la sezione che saranno divise in 3 e infine la sottotabella con gli esercizi.

SCHEDA	
Campi	Vincoli
Sottotabella NomeUtente	
Sottotabella Nome Sezione	
Sottotabella Categoria - Esercizio	

.

Matrice controllo degli accessi

Le categorie di utenti di MaiGim sono 3 :

-L'utente non registrato

-L'utente registrato

-Il personal trainer

L'utente non registrato non può eseguire nulla , ovviamente , non essendo registrato significa che non può avere una scheda allenamento.

L'utente registrato può eseguire il login , vedere il proprio profilo o la propria scheda , sia intera che per sezioni , eseguire il logout , modificare alcuni dei propri dati e recuperare la password.

Il personal trainer è simile ad un admin , può eseguire il login e logout , aggiungere , modificare e rimuovere utenti , creare , eliminare e modificare schede per gli utenti , può recuperare la propria password , visualizzare le schede degli utenti , modificare i propri dati.

Matrice controllo accessi web app :

attore\oggetto	Gestore Profilo	Gestore Scheda	Visualizzatore Scheda
utente non registrato	-	-	-
utente registrato	-Login -Logout -Visualizza proprio profilo -Modifica profilo	-	-Visualizza Sezioni -Visualizza Scheda
admin(istruttore)	-Login -Logout -Visualizza proprio profilo -Modifica proprio profilo -Aggiungi utente -Modifica utente -Cancella utente	-Modifica Scheda -Modifica Sezione -Aggiungi esercizio -Rimuovi esercizio	-Visualizza scheda utenti -Visualizza sezioni utenti

Flusso di controllo globale

La web app ha un flusso guidato da eventi(event-driven) : non esiste una sequenza di operazioni prestabilita , è l'utente a scegliere di volta in volta l'operazione da eseguire. L'architettura è composta da un web browser che effettua richieste ad un server che contiene la web app , ogni qual volta arriva una richiesta , vi sono opportuni metodi che eseguono quella determinata richiesta e invieranno l'output al client che lo mostrerà graficamente all'utente.

Condizioni limite

Configurazione

Per ogni oggetto persistente bisogna definire le fasi in cui viene creato , distrutto e archiviato nel database.

Profilo : L'oggetto profilo viene creato dall'admin nel momento in cui l'utente si iscrive alla palestra. , eseguendo il caso d'uso : "Il personal trainer vuole aggiungere un utente" . La rimozione avviene attraverso il caso d'uso : "Il personal trainer vuole cancellare un utente"

Scheda Utente : La scheda utente viene creata dall' admin (istruttore) eseguendo il caso d' uso : "Il personal trainer vuole modificare la scheda utente" e viene così archiviata nel database , la rimozione avviene attraverso la cancellazione dell' utente.

Esercizio : L'aggiunta di un esercizio avviene attraverso il caso d'uso : "Il personal trainer vuole aggiungere un esercizio" , viene così anche archiviato nel database e viene rimosso attraverso il caso d'uso : "Il personal trainer vuole rimuovere un esercizio".

Startup e Shutdown case

MaiGim prevede tre componenti run-time: MaiGim client (il web browser), MaiGim server (che include tutti i sottosistemi a cui il server ha accesso) e MaiGim database. Il Web Browser e il Database Server sono componenti off-the-shelf e sono avviate e spente indipendentemente. Il web browser viene acceso e spento dall'utente. Il server viene acceso dall'amministratore e dovrebbe funzionare 24 ore su 24, viene spento solo in caso di crash del sistema e in fase di aggiornamento. Per amministratore intendiamo un programmatore del sistema in analisi che può controllare il server da remoto o l'amministratore del web server scelto. La connessione al database viene gestita dal server: al momento dello start-up del server, il server stesso apre la connessione al DBMS e accede a MaiGim database. I casi d'uso di configurazione sono stati opportunamente inseriti nel RAD.

Exception use case

MaiGim può essere affetto da tre tipi di eccezioni :

- Fallimento hardware : Fallimento del sistema Server/DBMS.
- Mancata connessione ad internet.
- Errori nel software.

Nel caso di fallimento hardware , il DBMS da noi scelto (Firebase) si occuperà di mantenere i dati in uno stato consistente , in caso di fallimento del Server , i dati salvati non verranno alterati.

Nel caso in cui un dispositivo perda la connessione ad internet , il sistema non può far altro che aspettare che il dispositivo ri-ottenga la connessione.

In caso di errori software il sistema dovrà prevedere a segnalare l'errore all'utente o a ripristinare il bug.

Extension use case : Fallimento Hardware.

ID : Fallimento Hardware
Esteso da : start-up server , start-up database , shutdown server , shutdown database
<p>Il sistema hardware smette di funzionare correttamente</p> <p>Il sistema manda un messaggio di errore all'utente</p> <p>Il dbms cercherà di mantenere i dati consistenti</p> <p>Il dispositivo che ha provocato il fallimento viene chiuso.</p>

Extension use case : Mancata connessione ad internet.
Id : Mancata connessione ad internet
Esteso da : Tutti i casi d'uso.
<p>Il dispositivo non ha accesso ad internet</p> <p>Il sistema manda un messaggio di errore all'utente</p> <p>Il sistema aspetta che l'utente sia di nuovo connesso ad internet.</p>

Servizi dei sottosistemi

Nome sottosistema	Gestore Profilo
Descrizione	Sottosistema che permette di gestire i dati degli utenti registrati.
Servizi offerti	
Servizio	Descrizione
Registrazione	Permette di far registrare l'utente dall'admin del sistema
Visualizza	Permette all'utente di visualizzare il proprio profilo , mentre permette all'admin di visualizzare i profili di tutti gli utenti.
Modifica	Permette di modificare i dati utente
Cancella	Permette di cancellare l'utente
Login	Permette di loggare nel sistema
Logout	Permette di disconnettersi dal sistema

Nome sottosistema	Gestore Scheda
Descrizione	Sottosistema che permette la visualizzazione , la modifica e la cancellazione di una scheda
Servizi offerti	
Servizio	Descrizione
Visualizza Scheda	Permette di visualizzare la scheda di allenamento
Modifica Scheda	Permette di modificare gli esercizi della scheda
Cancella Scheda	Permette di cancellare la scheda
Modifica Sezione	Permette di modificare gli esercizi di una sezione
Cancella Sezione	Permette di ripristinare gli esercizi di una sezione
Visualizza Sezione	Permette di visualizzare una sezione

Sottosistema	Visualizzatore Scheda
Descrizione	Sottosistema che permette di visualizzare le sezioni di una scheda oppure scegliere la categoria di esercizi
Servizi Offerti	
Servizio	Descrizione
Scegli Sezione	Permette di scegliere una determinata sezione
Scegli Categoria	Permette di visualizzare una determinata categoria di esercizi

Glossario