



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



System Design Document

Riferimento	C03_SDD
Versione	1.9
Data	06/12/2020
Destinatario	Prof.ssa Filomena Ferrucci
Presentato da	Giosuè Sulipano
Approvato da	Andrea Massaro



Revision History

Data	Versione	Descrizione	Autori
25/11/2020	0.1	Stesura capitolo 1	M. Iannucci E. Iannaccone
26/11/2020	0.2	Stesura paragrafo 3.4, stesura capitolo 3.2, 3.3	Tutto il Team
27/11/2020	0.3	Stesura paragrafo 3.5	Marco sica
27/11/2020	0.4	Stesura paragrafo 3.1	Franzese Umberto
28/11/2020	0.5	Correzione del diagramma della gestione dei dati persistenti e stesura della tabella relativa alle definizioni presenti nello stesso diagramma	G. Ferrara, U. Franzese
29/11/2020	0.6	Modifica paragrafo 3.5 e stesura capitolo 2	Marco sica
01/12/2020	0.7	Correzione paragrafo 3.2,3.3	G. Ferrara, M. Iannucci
02/12/2020	0.8	Correzione Capitoli 1 e 2	U. Franzese E. Manzo G. Ferrara
02/12/2020	0.9	Correzione capitolo 3 e 4	M. Sica, E. Iannaccone, M. Iannucci
03/12/2020	1.0	Revisione	Tutto il Team
05/12/2020	1.1	Formattazione e Revisione del documento	Tutto il Team
16/12/2020	1.2	Revisione Schema Dati persistenti e Revisione della decomposizione	Tutto il Team
18/12/2020	1.3	Modifica gestione dati persistenti	Tutto il team



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

21/12/2020	1.4	Revisione Gestione dei dati persistenti	Tutto il team
24/12/2020	1.5	Revisione Gestione dei dati persistenti	U.Franzese
11/01/2021	1.6	Revisione Gestione dei dati persistenti	U.Franzese
13/01/2021	1.7	Revisione Gestione dei dati persistenti	U.Franzese
14/01/2021	1.8	Revisione del documento, modifica controllo accessi	M.Sica
19/01/2021	1.9	Revisione del documento, controllo della consistenza dei dati.	Tutto il Team



Team

Nominativo	Ruolo
Giosuè Sulipano	PM
Andrea Massaro	Revisore
Edoardo Iannaccone	Sviluppatore
Giampiero Ferrara	Sviluppatore
Emma Manzo	Sviluppatore
Marco Sica	Sviluppatore
Michele Iannucci	Sviluppatore
Umberto Franzese	Sviluppatore



Sommario

Revision History	2
Team	4
Sommario	5
1 Introduzione.....	7
1.1 Scopo del sistema	7
1.2 Design Goals.....	7
1.3 Definizioni, acronimi e abbreviazioni	9
1.4 Riferimenti.....	9
1.5 Panoramica.....	9
2 Architettura del Sistema Corrente.....	10
3 Architettura del Sistema Proposto	10
3.1 Panoramica.....	10
3.2 Decomposizione in sottosistemi.....	10
3.3 Mapping Hardware/Software.....	12
3.4 Gestione dei dati persistenti.....	13
3.5 Controllo degli accessi e sicurezza.....	17
3.6 Controllo globale del software	17
3.7 Condizioni limite	18
3.7.2 Terminazione.....	19
3.7.3 Fallimento del Sistema.....	20
3.7.3.1 Fallimento del Sistema.....	20
3.7.3.2 Mancata connessione al database.....	21
3.7.3.3 Errore di connessione	22
4 Servizi dei sottosistemi	23
4.1 Account	23
4.2 Scheda Allenamento	23
4.3 Agenda.....	24
4.4 Parametri Cliente.....	24
4.5 Abbonamento.....	25



5	Glossario.....	25
---	----------------	----



1 Introduzione

1.1 Scopo del sistema

myPersonalTrainer nasce dall'esigenza di semplificare e migliorare l'esperienza lavorativa di ogni Personal Trainer, supportando la sua attività quotidiana con una WebApp in grado di sostituire la sua confusionaria workspace, fatta di fogli di lavoro digitali e svariati documenti cartacei. Tutto ciò con l'obiettivo di soddisfare ogni tipo di esigenza: la gestione dei suoi clienti, la gestione della sua agenda e dei vari appuntamenti, la gestione e la creazione delle varie schede di allenamento per ogni cliente, e in ultimo il tener traccia degli abbonamenti. La piattaforma myPersonalTrainer rappresenta uno strumento utile ed innovativo anche dal punto di vista del cliente, poiché non avrà più bisogno di stampare le schede contenenti i vari esercizi per svolgere il proprio allenamento. In più, il cliente potrà visualizzare in ogni momento le sue statistiche e i suoi progressi, in modo da poter rendersi conto dell'effettivo andamento dei propri allenamenti e dei conseguenti risultati ottenuti. Un altro importante vantaggio è che myPersonalTrainer elimina completamente l'esigenza del cliente di doversi recare dal Personal Trainer per chiedere un appuntamento, una nuova scheda oppure per pagare il rinnovo dell'abbonamento. Grazie a questa piattaforma il cliente potrà fare tutto ciò comodamente dal proprio dispositivo in qualsiasi momento, ed in qualsiasi luogo esso si trovi.

1.2 Design Goals

I design goal che seguono sono ordinati in base alla priorità 1-3, in cui 1 corrisponde ad un'elevata priorità, 2 media e infine 3 bassa.

Priorità	ID Design	Descrizione design goal	Categoria	Origine
1	DG_1	Sicurezza: Il sistema dovrebbe garantire la protezione dei dati inseriti dagli utenti, difendendoli da accessi non autorizzati.	Dependability	RNF-5 RNF-17
1	DG_2	Leggibilità: Il sistema dovrebbe offrire una buona facilità di lettura del codice sorgente.	Maintenance	RNF-14



1	DG_3	Costi di sviluppo: Il sistema dovrebbe essere sviluppato rispettando il budget prefissato, ossia 50 ore di lavoro per ogni membro del team.	Cost	Top management
1	DG_4	Affidabilità: Il sistema dovrebbe garantire la consistenza dei dati.	Dependability	RNF-6 RNF-7 RNF-15
1	DG_5	Portabilità: Il sistema dovrebbe essere una WebApp responsive, consentendo l'utilizzo sia su piattaforme mobile che desktop.	Maintenance	RNF-2
2	DG_6	Usabilità: Il sistema dovrebbe essere facile da utilizzare, con interfacce grafiche semplici, chiare e intuitive.	End User	RNF-1
2	DG_7	Modificabilità: Il sistema e le sue varie funzionalità dovrebbero essere facilmente modificabili, anche in previsione di esigenze future e di eventuali malfunzionamenti.	Maintenance	RNF-16
2	DG_8	Fault Tolerance: Il sistema dovrebbe notificare ogni eventuale situazione di errore attraverso appositi avvisi.	Dependability	RNF-8
3	DG_9	Tempi di risposta: Il sistema dovrebbe garantire, in condizioni ottimali, tempi di risposta inferiori a 5 secondi.	Performance	RNF_10
3	DG_10	Throughput: Il sistema dovrebbe permettere l'interazione contemporanea di almeno 15 utenti.	Performance	RNF-11
3	DG_11	Estensibilità: Il sistema dovrebbe consentire agevolmente l'integrazione di sistemi deputati alla gestione delle transazioni economiche.	Maintenance	RNF-13 RNF-18

Trade – offs:



1. **Affidabilità vs Tempi di risposta:** È necessario garantire in ogni momento l'affidabilità del sistema, anche se ciò potrebbe comportare tempi di risposta più lunghi.
2. **Fault Tolerance vs Throughput:** Se il sistema va in sovraccarico, bisogna assicurare fault tolerance anche a discapito del throughput, abbassando quindi gli accessi concorrenti.
3. **Leggibilità vs Modificabilità:** La leggibilità deve sempre prevalere sulla modificabilità, per cui ogni modifica delle funzionalità di sistema dovrà rispettare strettamente i criteri di leggibilità previsti.
4. **Costi di sviluppo vs Estensibilità:** Nel caso in cui i tempi di rilascio siano stretti, si rispetteranno i costi di sviluppo a discapito dell'estensibilità, prevendo quindi nelle release successive funzionalità aggiuntive di bassa priorità.

1.3 Definizioni, acronimi e abbreviazioni

DG: Design Goal

WebApp : Abbreviazione di applicazione web

1.4 Riferimenti

- Requirement Analysis Document relativo a questo progetto.
- Statement of Work relativo a questo progetto.
- B. Bruegge, A.H. Dutoit, Object Oriented Software Engineering – Using UML, Patterns and Java, Prentice Hall

1.5 Panoramica

Nei capitoli successivi, verranno presentati: l'architettura del sistema corrente, l'architettura del sistema proposto ed in seguito verranno illustrati i servizi esposti da ciascun sottosistema.



2 Architettura del Sistema Corrente

Non c'è un sistema attuale con cui confrontarsi, da migliorare o su cui basarsi; sarà un sistema nuovo creato interamente da zero.

3 Architettura del Sistema Proposto

3.1 Panoramica

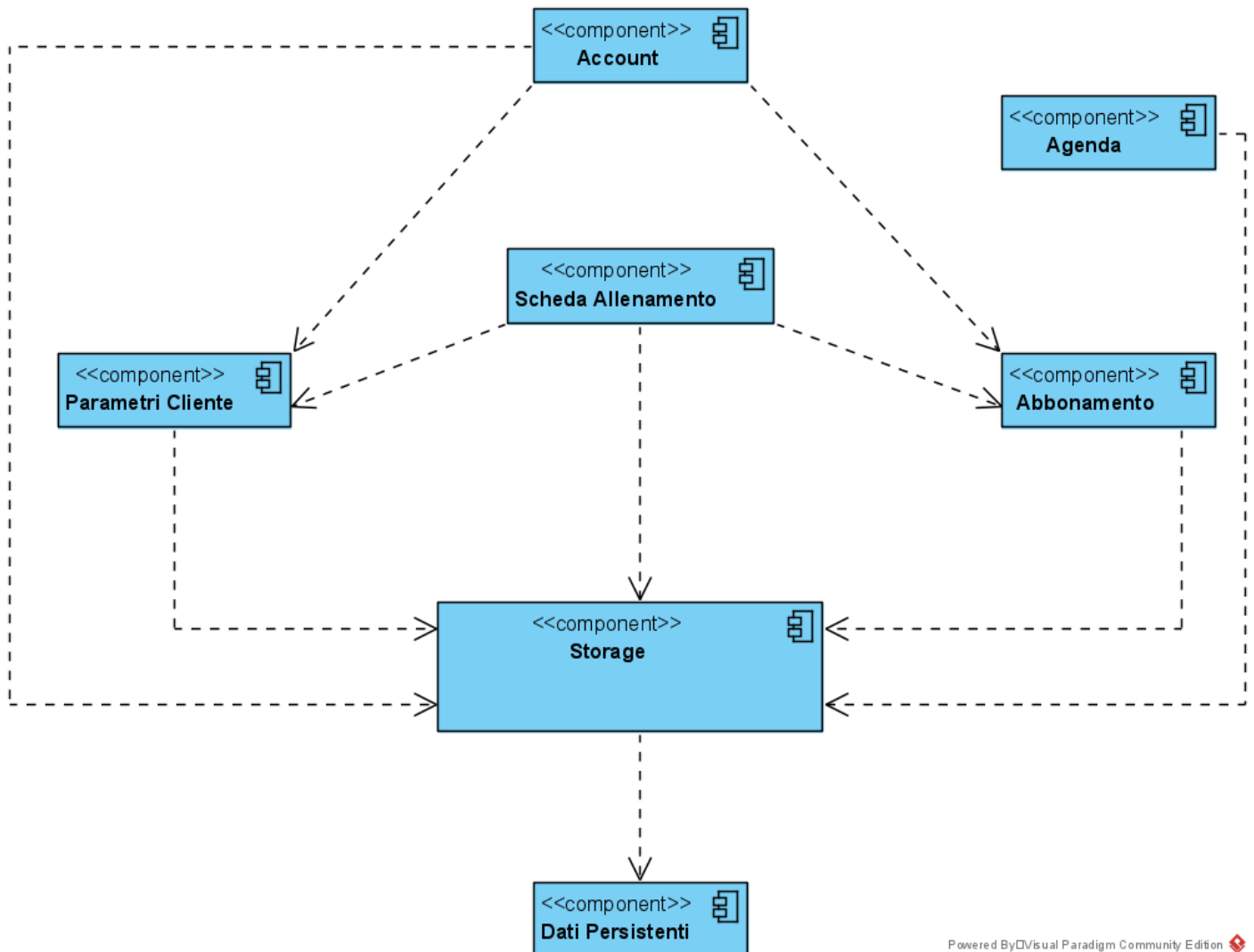
Il sistema da noi proposto sarà una WebApp gestita attraverso l'architettura MVC. La scelta di questo tipo di architettura nasce dal voler suddividere i vari elementi del sistema, agevolandone la sua manutenibilità e comprensione generale. I dati persistenti del sistema saranno salvati su un database non relazionale, Firebase. Firebase è stato scelto per la sua peculiarità di essere un cloud service, i dati quindi vengono mantenuti sul cloud, consentendo maggiore interoperabilità in caso di future estensioni ed una maggiore scalabilità.

3.2 Decomposizione in sottosistemi

Dopo l'analisi effettuata sul sistema, si è deciso di suddividerlo nei seguenti sottosistemi, per motivi di efficienza. Quindi, si è deciso di modellare i singoli componenti in modo da ottenere un basso accoppiamento e un'alta coesione, così da garantire bassa dipendenza tra i vari sottosistemi. I sottosistemi individuati sono:

- Agenda: si occupa della gestione degli appuntamenti e delle disponibilità.
- Account: si occupa della gestione degli account.
- Scheda Allenamento: si occupa della gestione delle schede di allenamento, dalla creazione alla visualizzazione.
- Abbonamento: si occupa di visualizzare lo stato degli abbonamenti dei clienti e ne permette il rinnovo.
- Parametri Cliente: si occupa della gestione dei parametri personali relativi ai vari clienti.
- Dati Persistenti: si occupa di gestire la persistenza dei dati attraverso una base di dati non relazionale.
- Storage: si interpone tra gli altri sottosistemi e il sottosistema di Persistenza.

Di seguito sono mostrate le dipendenze tra i sottosistemi tramite un UML component diagram.



Powered By UML Visual Paradigm Community Edition

L'architettura utilizzata è l'MVC. I sottosistemi si dividono nel seguente modo:

Il livello Controller prevede la gestione di cinque sottosistemi:

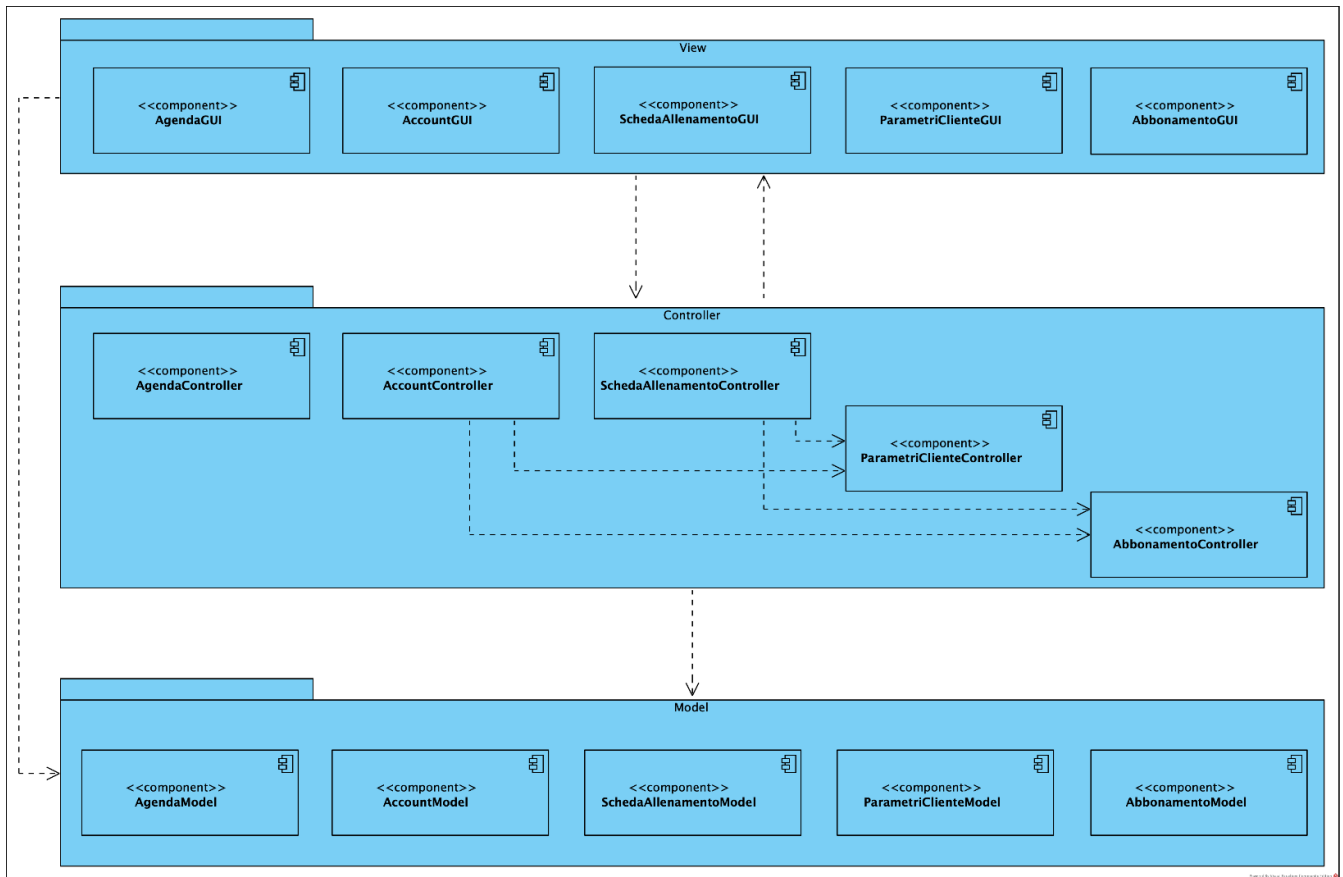
- Account
- Scheda di Allenamento
- Subscription
- Agenda
- Parametri

Il livello View prevede la gestione di cinque sottosistemi:

- Account GUI
- Scheda di Allenamento GUI
- Abbonamento GUI
- Agenda GUI
- Parametri Cliente GUI

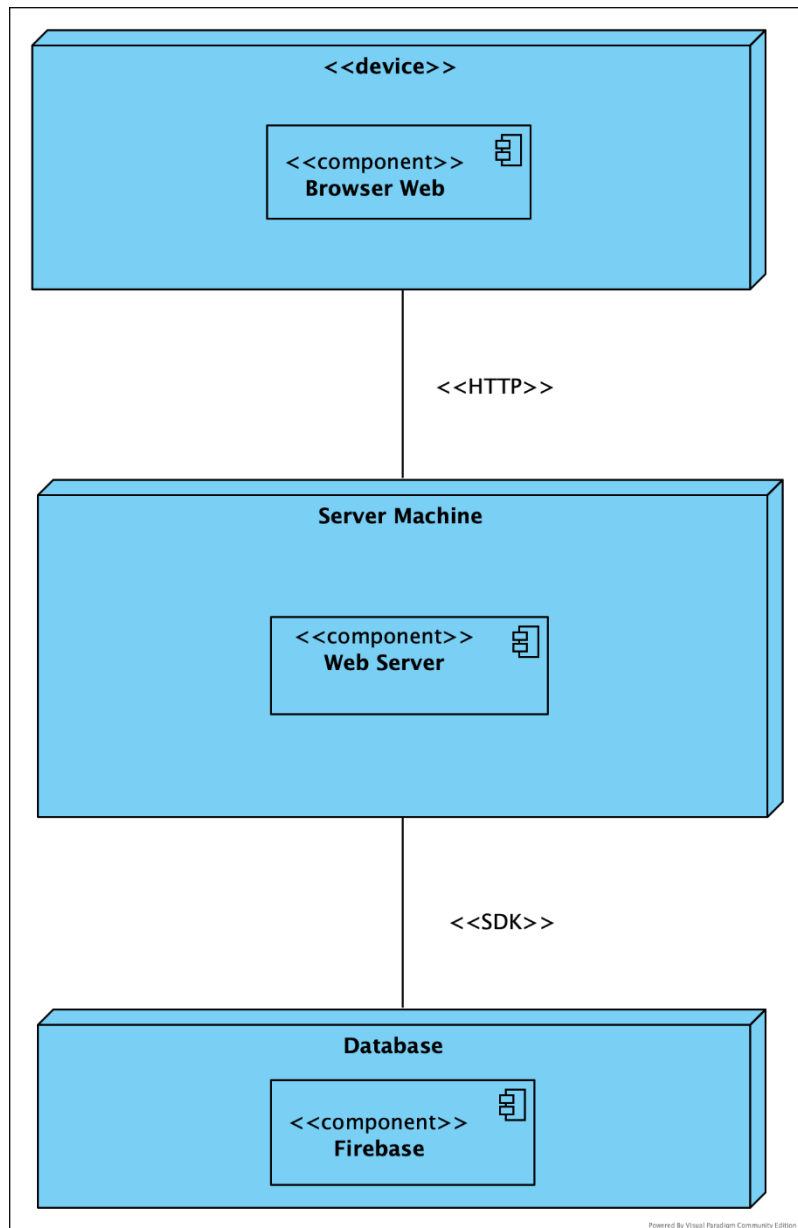
Il livello Model prevede la gestione di un sottosistema:

- Dati Persistenti



3.3 Mapping Hardware/Software

Il sistema si sviluppa su tre nodi principali su cui si basa l'intera architettura della nostra web-app. Il primo nodo è <<device>>, che permette all'utente di poter accedere alla piattaforma. I device possono essere differenti, sia desktop che mobile. Il secondo nodo permette al nostro servizio web based, di poter essere ospitato su un server disponibile in rete. Ogni device si connette al web server attraverso il protocollo HTTP. Il web server a sua volta si interfaccia, tramite le SDK di Google, al Database Firebase in cui sono memorizzati tutti i dati persistenti.



3.4 Gestione dei dati persistenti

Per il nostro progetto utilizzeremo un database NoSQL chiamato Firebase.

Firebase è un potente servizio online che permette di salvare e sincronizzare i dati elaborati da applicazioni web e mobile. La scelta di questo database è nata dalla volontà di usare un database versatile e utilizzabile su diversi tipi di tecnologie hardware e software, permettendo una protezione più che adeguata dei dati degli utenti del nostro sistema. Inoltre, Firebase, grazie al suo orientamento ai documenti (composti da un insieme di chiave-valore), permetterà uno scambio di messaggi più omogeneo da e verso



il database. Altro vantaggio di Firebase è la sua scalabilità data la sua peculiarità di esser un database che salva i dati in cloud. L'utilizzo di questo database sarà possibile attraverso l'utilizzo delle SDK ufficiali di Google.

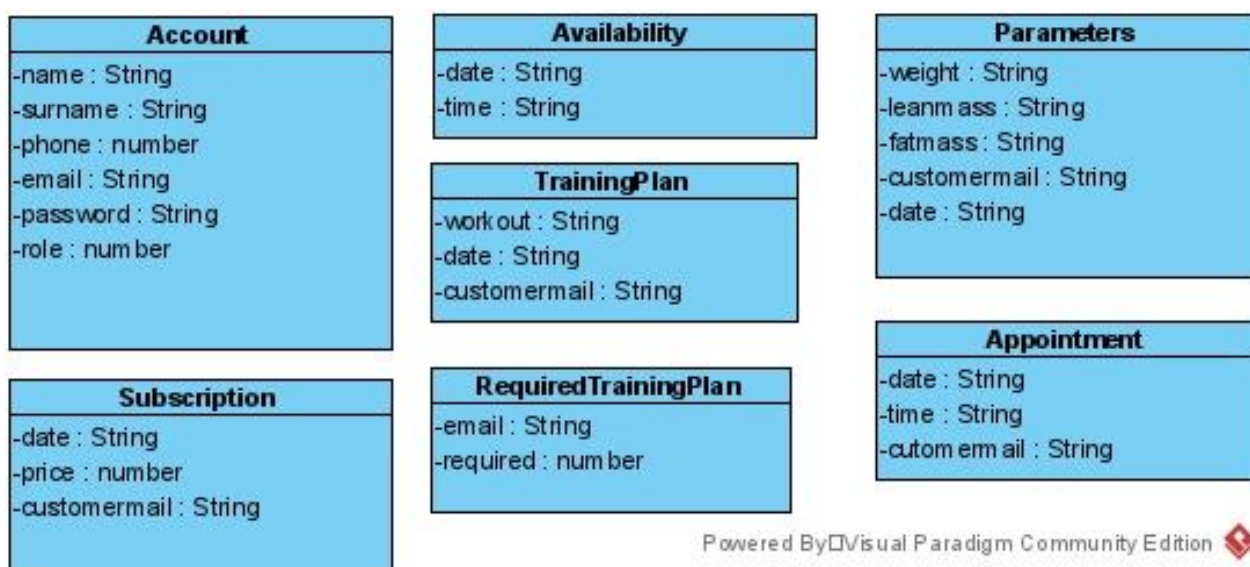
Il diagramma che segue permette di rappresentare la gestione dei dati persistenti. Non abbiamo utilizzato un classico schermo ER poiché Firebase è un database NoSQL, pertanto non vi sono relazioni. Le varie entità sono rappresentate mediante documenti.

A differenza del Class Diagram presente nel RAD, si è deciso di eliminare la generalizzazione User, optando per un'unica classe Account in grado di rappresentare sia la figura del Personal Trainer, che del Customer.

Essi, infatti, avranno in comune tutti gli attributi già previsti, con l'aggiunta di un nuovo parametro "Ruolo", in grado di distinguere i due tipi di utente.

Inoltre è stata introdotta una nuova entità "Required Training Plan", al fine di poter gestire e disaccoppiare la richiesta di una scheda, sia dall'account che la richiede, che dalla scheda che verrà effettivamente creata.

Tale scelta permetterà di avere una maggiore indipendenza tra le entità coinvolte, facilitandone la gestione, manutenzione e comprensione.



Appointment

Name	Type	Null	Descrizione
------	------	------	-------------



Date	String	Not Null	Data dell'appuntamento
Time	String	Not Null	Ora dell'appuntamento
CustomerMail	String	Not Null	Cliente che ha prenotato l'appuntamento

Subscription

Name	Type	Null	Descrizione
expDate	String	Not Null	Data dell'iscrizione
Price	Number	Not Null	Prezzo da pagare
CustomerMail	String	Not Null	Mail del proprietario dell'abbonamento
sentNotification	Number	-	Flag che indica se la notifica di imminente scadenza è stata inviata

Availability

Name	Type	Null	Descrizione
Date	String	Not Null	Data in cui il Personal Trainer è disponibile
Time	String	Not Null	Ora in cui il Personal Trainer è disponibile nella data

Account

Name	Type	Null	Descrizione
Name	String	Not Null	Nome del cliente
Surname	String	Not Null	Cognome del cliente
Phone	Number	Not Null	Numero di telefono del cliente
Email	String	Not Null	Mail del cliente
Password	String	Not Null	Password del cliente



Role	Number	Not Null	Tipo di utente registrato nel sistema. Il 0 corrisponde al ruolo di Cliente, mentre 1 al Personal Trainer.
------	--------	----------	--

Parameters

Name	Type	Null	Descrizione
Weight	String	Not Null	Peso indicato dal cliente
LeanMass	String	Not Null	Massa magra indicata dal cliente
FatMass	String	Not Null	Massa magra indicata dal cliente
mailClient	String	Not Null	Cliente che ha indicato questi parametri
insertionDate	String	Not Null	Data dell'inserimento dei parametri

TrainingPlan

Name	Type	Null	Descrizione
exercises	String	Not Null	Esercizi che ha assegnato il Personal Trainer
Date	String	Not Null	Data di assegnazione della scheda
CustomerMail	String	Not Null	Cliente a cui è stata assegnata la scheda

RequiredTrainingPlan

Name	Type	Null	Descrizione
email	String	Not Null	Email del cliente che richiede la scheda di allenamento
required	number	Not Null	Flag che indica se il cliente vuole o no la scheda



3.5 Controllo degli accessi e sicurezza

Gli accessi sono controllati tramite e-mail e password attraverso una schermata dedicata; inoltre, la password sarà crittografata. Si terrà traccia di un utente loggato tramite la sessione del server.

- Un cliente può accedere: alle sue schede di allenamento, alle sue statistiche, ai suoi dati di account, allo stato del suo abbonamento, agli appuntamenti che lo vedono coinvolto e può consultare le disponibilità del Personal Trainer. Infine, può anche richiedere e cancellare un appuntamento.
- Il Personal Trainer invece può: accedere ai dati degli utenti (abbonamenti e statistiche), creare nuovi account, definire la sua disponibilità, consultare e cancellare gli appuntamenti in agenda.

Il cliente accede alle varie funzionalità del sistema, ma non visualizzerà gli stessi dati e non avrà le stesse funzionalità del Personal Trainer; il quale avrà una gestione più ampia.

Oggetto Attore	Account	Appointment	Availability	Subscription	Training Plan	Parameters
Personal Trainer	<<create>>					
	authentication	<<delete>>				
	logout	checkAppointment		check	<<create>>	getParameter
	getInfo	nt	<<create>>		>>	rs
	manageData	<<create>>	<<delete>>			
Cliente	authentication				requestTr	
	logout	<<delete>>	getAvailability	renew	ainingPla	<<create>
	getInfo	reserve	y	check	n	>
	manageData				consult	getParameter
					TrainingPlan	rs
					getPDF	

3.6 Controllo globale del software

La piattaforma myPersonalTrainer fornisce diverse funzionalità, infatti è necessario richiedere un accesso simultaneo a oggetti diversi, così da poter permettere l'esecuzione di diverse operazioni ogni qual volta



vengano richieste. È necessario, quindi, utilizzare una gestione di flusso globale di tipo event-driven, i sottoinsiemi forniranno un gestore ad eventi per gestire la moltitudine di richieste. Il sistema rimane in attesa di un'azione da parte dell'utente. L'azione scatena un evento che sarà gestito dal corrispettivo handler, che a sua volta reindirizzerà il controllo del flusso al sottosistema che si occupa della logica di controllo e il gestore del controllo poi si rivolge ai servizi per la logica applicativa.

3.7 Condizioni limite

In questo capitolo verranno illustrati use case riguardanti le condizioni limite, ovvero accensione, spegnimento e le eccezioni:

3.7.1 Start-up del Sistema

Identificativo	Accensione del sistema	Data	28/11/2020
UC_GS_1		Vers.	0.00.001
		Autori	Manzo Emma
			Iannaccone Edoardo
Descrizione	Definisce come il Personal Trainer può avviare il sistema.		
Attore Principale	Personal Trainer È interessato ad avviare il sistema		
Attori secondari	NA		
Entry Condition	La macchina su cui è installato il sistema è accesa		
Exit Condition	Il sistema è stato avviato.		
On success			
Exit Condition	Il sistema resta spento.		
On failure			
Rilevanza/User Priority	Alta		
Frequenza stimata	4/anno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Personal Trainer:	Esegue l'inizializzazione del server inserendo i suoi dati.	



2	Sistema:	Mostra un bottone con scritto “Accendi Sistema”.
3	Personal Trainer:	Clicca sul bottone.
4	Sistema:	Notifica che l’accensione è avvenuta correttamente.

3.7.2 Terminazione

Nel momento in cui il Personal Trainer deciderà di non usufruire più dei servizi del sistema, gli basterà procedere con la terminazione del sistema.

Identificativo	Spegnimento del Sistema	Data	28/11/2020
UC_GS_2		Vers.	0.00.001
		Autori	Manzo Emma
			Iannaccone Edoardo
Descrizione	Definisce come il Personal Trainer può terminare l'esecuzione del sistema.		
Attore Principale	Personal Trainer		
	Ha interesse nel terminare il sistema.		
Attori secondari	NA		
Entry Condition	Il Personal Trainer ha effettuato il login.		
Exit Condition	Il sistema termina correttamente.		
On success			
Exit Condition	Il sistema resta acceso.		
On failure			
Rilevanza/User Priority	Alta		
Frequenza stimata	4/anno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Personal Trainer:	Effettua il logout dalla piattaforma.	



2	Sistema:	Notifica che il logout è avvenuto con successo e mostra nuovamente la schermata di login.
3	Personal Trainer:	Il Personal Trainer spegne il server dall'apposito servizio.
4	Sistema:	Notifica che lo spegnimento è avvenuto correttamente.

3.7.3 Fallimento del Sistema

Di seguito, sono presentate situazioni dove il sistema incorre in fallimento:

3.7.3.1 Fallimento del Sistema

Identificativo	Fallimento del Sistema	Data	28/11/2020
UC_GS_3		Vers.	0.00.001
		Autori	Manzo Emma
			Iannaccone Edoardo
Descrizione	Definisce come il Personal Trainer può visualizzare le notifiche di fallimento del sistema.		
Attore Principale	Personal Trainer È interessato a mantenere il sistema e a ricevere le notifiche in seguito ad errori o problemi del sistema.		
Attori secondari	NA		
Entry Condition	Il sistema rileva un errore.		
Exit Condition	Il sistema viene ripristinato.		
Rilevanza/User Priority	Alta		
Frequenza stimata	4/anno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Sistema:	Inoltra un messaggio di segnalazione contenente la causa di terminazione.	
2	Personal Trainer:	Riceve il messaggio di errore proveniente dal sistema e lo ripristina.	



3	Sistema:	Il sistema funziona correttamente.
---	----------	------------------------------------

3.7.3.2 Mancata connessione al database

Identificativo	Mancata connessione al database	Data	28/11/2020
UC_GS_4		Vers.	0.00.001
		Autori	Manzo Emma Iannaccone Edoardo
Descrizione	Definisce in che modo il Personal Trainer può visualizzare le notifiche di fallimento del sistema.		
Attore Principale	Personal Trainer È interessato a mantenere il sistema e a ricevere le notifiche in seguito ad una mancata connessione al database.		
Attori secondari	NA		
Entry Condition	Il sistema rileva un errore dovuto ad una mancata connessione al database.		
Exit Condition	La connessione risulta ristabilita.		
Rilevanza/User Priority	Alta		
Frequenza stimata	4/anno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Sistema:	Segnala l'errore, impedendo al Personal Trainer di poter accedere ai dati contenuti al suo interno.	
2	Personal Trainer:	Riceve il messaggio di errore proveniente dal sistema e prova a ristabilire la connessione col database.	
3	Sistema:	Il sistema ripristina la connessione e lo notifica al Personal Trainer.	



3.7.3.3 Errore di connessione

Identificativo		Errore di connessione	Data	28/11/2020
UC_GS_5			Vers.	0.00.001
			Autori	Manzo Emma
				Iannaccone Edoardo
Descrizione		Definisce in che modo il Personal Trainer può visualizzare le notifiche di fallimento di connessione del sistema.		
Attore Principale		Personal Trainer È interessato a mantenere il sistema ed a ricevere le notifiche in seguito a problemi di connessione.		
Attori secondari		NA		
Entry Condition		Il sistema rileva un errore dovuto alla connessione.		
Exit Condition		Il sistema si connette correttamente.		
Rilevanza/User Priority		Alta		
Frequenza stimata		4/anno.		
Extension point		NA		
Generalization of		NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO				
1	Sistema:	Invia un messaggio relativo all'errore.		
2	Personal Trainer:	Riceve il messaggio di errore proveniente dal sistema e prova a ristabilire una connessione di rete affidabile, per poter riutilizzare le sue funzionalità.		
3	Sistema:	La connessione risulta ristabilita.		



4 Servizi dei sottosistemi

4.1 Account

Servizio	Descrizione	Interfaccia
Login	Servizio che dà la possibilità ad un qualsiasi utente, Personal Trainer o cliente, di accedere alla piattaforma.	AccountService
ManageAccount	Servizio che permette di gestire le proprie informazioni personali, offrendo anche la possibilità di modificare la password.	AccountService
CreateAccount	Servizio che consente al Personal Trainer di creare un nuovo account Cliente da fornire al proprio Cliente.	AccountService
ViewCustomerAccount	Servizio che consente al Personal Trainer di visualizzare tutte le informazioni relative ad un account Cliente.	AccountService
NotifyCustomerRegistration	Servizio che consente al cliente di ricevere una mail con le credenziali di accesso al proprio account.	AccountService

4.2 Scheda Allenamento

Servizio	Descrizione	Interfaccia
ViewTrainingPlan	Servizio che permette al cliente di visualizzare la propria scheda di allenamento.	TrainingPlanService
RequestTrainingPlan	Servizio che permette al cliente di richiedere una nuova scheda di allenamento.	TrainingPlanService



ExportTrainingPlan	Servizio che permette al cliente di TrainingPlanService esportare e scaricare la scheda in formato PDF.
CreateTrainingPlan	Servizio che permette al Personal Trainer di TrainingPlanService stilare una nuova scheda.

4.3 Agenda

Servizio	Descrizione	Interfaccia
RequestAppointment	Servizio che permette al cliente di richiedere un nuovo appuntamento.	AgendaService
ShowAgenda	Servizio che permette al cliente di consultare l'agenda del Personal Trainer.	AgendaService
InsertAvailability	Servizio che permette al Personal Trainer di inserire le proprie disponibilità.	AgendaService
ShowAppointment	Servizio che permette al Personal Trainer di visualizzare gli appuntamenti prenotati.	AgendaService
CancelAppointment	Servizio che permette al cliente e al Personal Trainer di cancellare gli appuntamenti previsti.	AgendaService
NotifyCancellation	Servizio che notifica al cliente e al Personal Trainer la cancellazione di un appuntamento.	AgendaService

4.4 Parametri Cliente

Servizio	Descrizione	Interfaccia
ManageParameters	Servizio che permette al cliente di gestire i propri parametri fisici.	ParametersService
ShowParameters	Servizio che permette al cliente di visualizzare i propri parametri fisici.	ParametersService



ShowCustomerParameters	Servizio che permette al Personal Trainer di visualizzare i parametri fisici dei propri clienti.	ParametersService
-------------------------------	--	-------------------

4.5 Abbonamento

Servizio	Descrizione	Interfaccia
ShowSubscription	Servizio che permette al cliente di visualizzare le informazioni relative al proprio abbonamento.	SubscriptionService
PaySubscription	Servizio che permette al cliente di pagare il rinnovo del proprio abbonamento.	SubscriptionService
ShowExpiringSubscription	Servizio che permette al Personal Trainer di visualizzare gli abbonamenti in scadenza.	SubscriptionService
NotifyExpiringSubscription	Servizio che notifica al cliente ed al Personal Trainer che l'abbonamento è in scadenza.	SubscriptionService

5 Glossario

Nome	Descrizione
SDK	Indica genericamente un insieme di strumenti utilizzati per lo sviluppo del software.
Browser	Applicazione per l'acquisizione, la presentazione e la navigazione di risorse sul web.
Database	Insieme di dati strutturati in versione digitale.
Firebase	Firebase è una piattaforma per la creazione di applicazioni per dispositivi mobili e web sviluppata da Google.



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