

O.D.D

Object 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

1. Introduzione

1.1 Obiettivi Design Trade-Offs

✓ Prestazioni vs. Costi

Considerando il sistema che stiamo realizzando, possiamo dire che il non eccessivo budget a nostra disposizione ci ha consentito di realizzare il prodotto utilizzando materiale open source partendo da zero minimizzando così i costi e rendendo il sistema più che soddisfacente.

✓ Interfaccia vs. Tempo di risposta

Il tempo di risposta tra server e interfaccia sono impercettibili dall'utente.

Questo è possibile grazie all'utilizzo del routing dinamico che permette di aggiornare dinamicamente le varie pagine della web app che non ha nulla da invidiare ad un'app progettata per essere eseguita localmente.

✓ Interfaccia vs. Easy-Use

L'interfaccia grazie alla quasi totale assenza dell'inserimento manuale degli input appare pulita, veloce ed adatta ad ogni tipologia di utenza che potrà svolgere qualsiasi operazione senza accorgersi dei vari caricamenti che avverranno in background né di nessuna operazione con il database.

✓ Costi vs. Mantenimento

Grazie all'utilizzo di Angular 5 ed alla sua modularità, il sistema sarà completamente manutenibile e personalizzabile in modo da soddisfare tutte le richieste presenti e future.

Il pacchetto Firebase che si occupa dello storage dei file, dell'autenticazione e del database è free fino ad un massimo di 100 iterazioni al secondo, dopodiché avrà un costo a consumo.

1.2 Linee guida per l'interfaccia

Il sistema è multiutente, ciò prevede che qualsiasi tipo di utenza possa eccedere ad esso attraverso la stessa form di login e sarà il sistema a reindirizzare chi effettua l'accesso verso la propria area fornendo i rispettivi permessi di visualizzazione e modifica.

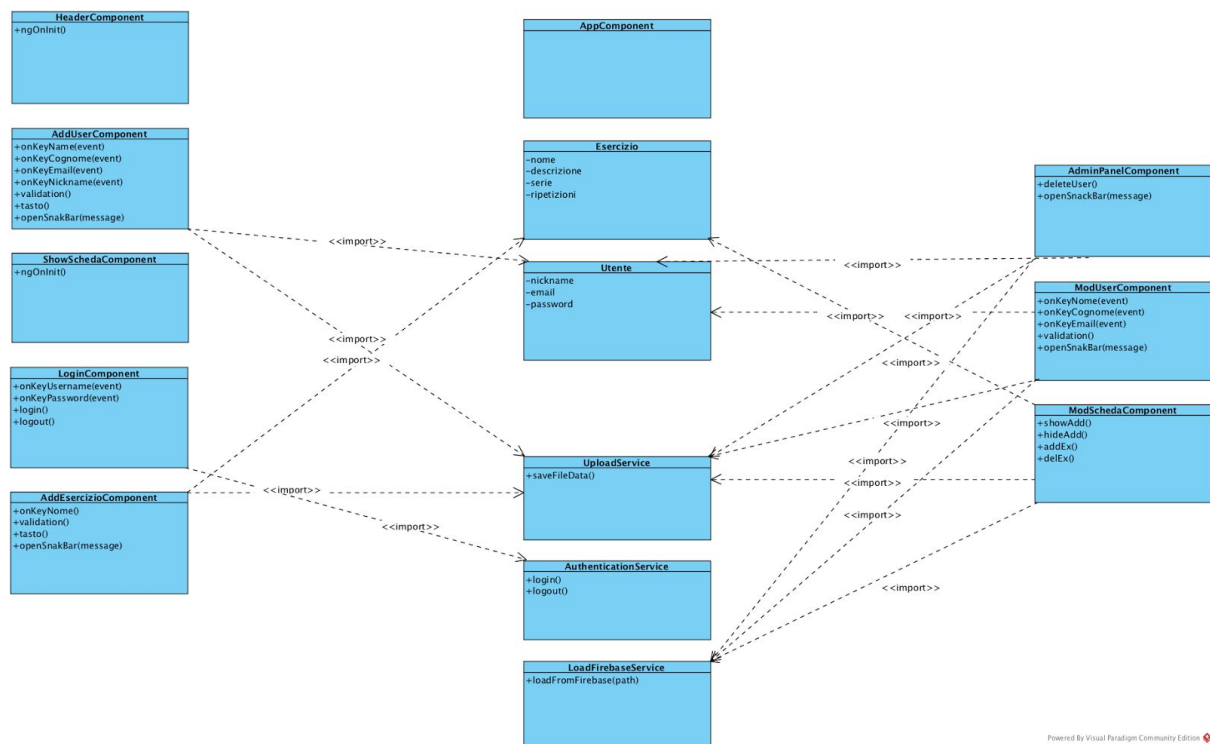
-Utente

Dopo aver effettuato l'accesso sarà reindirizzato verso propria pagina personale dalla quale sarà possibile visualizzare la propria scheda e modificare i propri dati.

-Personal Trainer

Dopo aver effettuato l'accesso sarà reindirizzato verso il pannello attraverso il quale può amministrare gli utenti e gestire le varie schede.

2. Class Interface



2.1 Contratti delle singole classi

-AddUserComponent

Precondizioni: devono essere stati compilati tutti i campi, inoltre l'utente deve essere personal trainer.

Postcondizioni: se tutti i dati sono corretti un nuovo utente sarà inserito nel database.

-LoginComponent

Precondizione: devono essere stati immessi i dati.

Postcondizioni: se i dati sono corretti sarà effettuato l'accesso al servizio.

-AddEsercizioComponent

Precondizione: devono essere stati immessi i dati, inoltre l'utente deve essere personal trainer.

Postcondizione: se i dati sono corretti sarà registrato un nuovo esercizio

-AdminPanelComponent

Precondizione: l'utente deve essere personal trainer.

Postcondizione: il personal trainer può effettuare tutte le operazioni a lui concesse.

-ModUserComponent

Precondizione: l'utente deve inserire i dati corretti ed essere personal trainer.

Postcondizione: il personal trainer avrà modificato i dati utente.

-ModSchedaComponent

Precondizione: l'utente deve inserire i dati corretti ed essere personal trainer.

Postcondizione: il personal trainer avrà modificato la scheda utente.

-UploadService

Precondizione: bisogna essere loggati.

Postcondizione: sarà effettuato l'upload di un oggetto nel database

-LoadFirebaseService

Precondizione: bisogna essere loggati.

Postcondizione: sarà caricato un oggetto dal database

