Università degli Studi di Salerno

Corso di Ingegneria del Software

MyGym

Versione 5.0

LOGO PROGETTO

Data: 02/10/2025

Progetto: Nome Progetto	Versione: X.Y
Documento: Titolo Documento	Data: GG/MM/AAAA

Coordinatore del progetto:

Nome	Matricola
Raffaele Cirillo	0512119545
Luigi Aquino	0512120291

Partecipanti:

Nome	Matricola
Raffaele Cirillo	0512119545
Luigi Aquino	0512120291
Gerardo Aquino	0512115624
Vincenzo Giordano	0512119470

Scritto da:	Raffaele Cirillo, Luigi Aquino, Gerardo Aquino, Vincenzo Giordano

Revision History

Data	Versione	Descrizione	Autore
02/10/2025	1	Software prenotazioni in palestra	Raffaele Cirillo, Luigi Aquino, Gerardo Aquino, Vincenzo Giordano
08/10/2025	2	Rivisitazione della parte introduttiva del progetto con aggiunta degli attori	Raffaele Cirillo, Luigi Aquino, Gerardo Aquino, Vincenzo Giordano
14/10/2025	3	Problem statement	Raffaele Cirillo, Luigi Aquino, Gerardo Aquino, Vincenzo Giordano
28/10/2025	4	Requirements Analysis Document	Raffaele Cirillo, Luigi Aquino, Gerardo Aquino, Vincenzo Giordano
11/11/2025	4.2	Dynamic model and Object model	Raffaele Cirillo, Luigi Aquino, Gerardo Aquino, Vincenzo Giordano
25/11/2025	5.0	System Design	Raffaele Cirillo, Luigi Aquino, Gerardo Aquino, Vincenzo Giordano

Sommario

1 OBIETTIVI DESIGN

Illustriamo nella seguente tabella gli obiettivi di design per il sistema e le relative priorità (a numeri più bassi corrispondono priorità più elevate). Per ogni obbiettivo riportiamo anche l'origine, facendo riferimento, in particolare, all'identificativo del requisito non funzionale ad esso associato.

Priorità	ID	Descrizione	Categoria	Origine
1	DG_1	**Robustezza**: Il sistema deve sopravvivere agli input errati degli utenti. In un caso del genere il sistema non deve accettare l'input e notificare l'utente invitandolo a correggere l'errore	Affidabilità	RNF_2
3	DG_2	**Sicurezza**: Il sistema deve comunicare tramite protocollo HTTPS in modo da assicurare una maggiore sicurezza.	Sicurezza	RNF_8
2	DG_3	**Usabilità**: Il sistema deve essere facile da utilizzare senza necessariamente consultare la documentazione anche tramite l'utilizzo di un menu contestuale che permette di muoversi all'interno del sito in qualsiasi momento. I contenuti dovranno essere fruibili attraverso dispositivi sia	Usabilità	RNF_1

		desktop che mobile ed accessibili attraverso un numero ridotto di interazioni.		
3	DG_4	**Prestazioni**: Il sistema deve essere in grado di supportare fino a 1000 utenti simultaneamente e	Prestazioni	RNF_3
3	DG_5	**Tempi di risposta**: Il sistema deve elaborare le richieste e produrre output in meno di 2 secondi.	Prestazioni	RNF_3

1.2 Compromessi

Tempo di rilascio vs Funzionalità

Il team si impegna nel consegnare il sistema completo di tutte le sue funzionalità nei tempi stabiliti tenendo conto di un possibile ritardo nella consegna.

Prestazioni vs Costi

Il team cercherà di ottenere le migliori prestazioni per non sforare il budget a disposizione.

1.3 Definizioni, acronimi ed abbreviazioni

- Piattaforma = Applicazione web
- CRUD = Create, Read, Update, Delete

1.4 Riferimenti

- Requisiti funzionali: Sezione 3.2 del Problem Statement
- Requisiti non funzionali: Sezione 3.3 del Problem Statement

1.5 Panoramica

L'architettura segue il pattern MVC adattato a un'architettura a microservizi/three-tier:

- Il **model** si occupa della di gestire i dati e quindi sarà responsabile dell'interazione con il database sottostante
- La **view** si occupa di curare l'interazione con l'utente e quindi avrà il compito di gestire la formattazione dei dati che verranno visualizzati.

● Il **controller** dopo aver ricevuto i comandi forniti dall'utente si occuperà di elaborare i dati, passarli al model se necessario e inviare la risposta al view appropriato.

L'utilizzo del modello MVC comporta numerosi vantaggi, tra i quali ricordiamo la possibilità di suddividere il lavoro più facilmente tra i vari componenti del team e la maggiore agilità negli interventi di manutenzione.

All'interno del nostro sistema:

● Il model verrà realizzato utilizzando classi Java appropriate.
● La parte di view verrà implementata utilizzando pagine HTML e JavaScript.
● I control saranno realizzati tramite Servlet.

2 Decomposizione del Sistema

Per realizzare il sistema è stata usata un'architettura three-tier. Questo è un caso particolare di un'architettura multi-tier in cui la logica dell'applicazione viene suddivisa in tre layer:

• Presentation layer: composto dalle interfacce grafiche e in particolare dai boundary object come le form che vengono compilate dall'utente (Frontend, ovvero l'interfaccia che gli utenti utilizzeranno per interagire col sistema).

• Application layer: composto dagli oggetti che di occupano della gestione del controllo, dell'elaborazione dati e di notificare i cambiamenti al Presentation layer. Questo strato interagisce con il database sottostante tramite lo data layer.

• Data layer: si occupa della memorizzazione dei dati persistenti e del loro recupero dal database ad esempio eseguendo delle query.

2.1 Gestione Sottosistema

Per ogni sottosistema riportiamo responsabilità, attori, servizi e endpoint.

User Interface

- Interfacce: pagine React per web, schermi mobile.
- Funzioni: ricerca corsi, visualizzazione sessioni, prenotazione, gestione profilo.

Gestione Account

- Attori: Utente, Istruttore, Admin.
- Funzionalità: registrazione, login, logout, reset password, gestione ruoli.

Gestione Corsi e Sessioni

- Funzionalità: CRUD corsi, gestione calendario e capienza per sessione.

Gestione Prenotazioni

- Funzionalità: prenota, cancella, gestione waitlist, regole di conferma.

- Requisito critico: operazione atomica per evitare overbooking.

Pagamento

- Funzionalità: checkout, webhook, riconciliazione pagamenti e rimborsi.

Notifiche

- Funzionalità: invio email, push, SMS; retry e log degli invii.

Presenze

- Funzionalità: validazione token QR, conteggio affluenza in tempo reale.

Report

- Funzionalità: generazione report amministrativi, esportazione su S3.

2.2 Hardware/Software Mapping

L'architettura del sistema MyGym è di tipo client/server. Il server riceve le richieste da parte del client, e risponde in tempo utile. I motivi di questa scelta sono:

● **Usabilità**: il sistema potrà essere utilizzato su una varietà di macchine e sistemi operativi, da computer fissi a dispositivi mobili;

●**Sicurezza**: per ogni tipologia di utente che effettua l'accesso al sistema, vi sarà un'interfaccia grafica apposita, tramite la quale ogni attore potrà eseguire le operazioni ad esso riservate.

● **Affidabilità**: entrambi i componenti client e server devono essere affidabili ed essere in grado di mantenere i propri dati anche in seguito a guasti, quindi deve essere possibile effettuare dei backup automatici giornalieri.

Web Server

Il server utilizzato è Apache Tomcat.

Interface layer

L'utente utilizza il sistema mediante un Browser installato all'interno del suo dispositivo (ad es. Chrome, Firefox).

Application Logic layer

Il sistema, e quindi le funzionalità, sono implementate in linguaggio Java (Servlet). Il codice in JavaServlet verrà tradotto in linguaggio HTML e JSP, inoltre, il codice risultante viene inviato al browser del client.

Database Server

Il DBMS usato è MySQL, il quale presenta molte API che permettono l'interazione tra sistema e database.

2.3 Gestione dei dati persistenti

Per la gestione dei dati persistenti si è deciso di utilizzare MySQL, questo per due motivi principali:

- I dati sono ben strutturati e per questo si prestano particolarmente ad essere memorizzati in questo DBMS
- Per abbattere ulteriormente i costi, strutturare dei file avrebbe richiesto maggior lavoro e quindi più costi; inoltre, MySQL è gratuito e di conseguenza comporta un costo in meno.

Infine, ovviamente MySQL ci permette di trattare i dati ed effettuare operazioni con estrema semplicità e di aggiungere un ulteriore livello di sicurezza per l'accesso ai dati.

Le seguenti entità, proveniente dal Class Diagram verranno rese persistenti:

- Corsi
- Clienti
- Prenotazioni
- Istruttori
- Amministratore

2.4 Controllo degli accessi e sicurezza

Il controllo degli accessi viene gestito attraverso l'inserimento di e-mail e password personali, che garantiranno l'accesso alle operazioni permesse in base al tipo di utente registrato. Si ricorrerà all'utilizzo della sessione del server per tenere traccia dell'utente loggato. Per questioni di efficienza, la sessione sarà attiva per soli 10 minuti dopo l'ultima interazione dell'utente col sistema. Il salvataggio delle password nel database sarà criptato. Di seguito

viene elencata la Global Access Table che descrive per ogni cella della matrice quali operazioni può effettuare ogni attore su un oggetto.

	Account	Corsi	Prenotazioni	Istruttori	Amministratore
Cliente non registrato	Registrazione	-Visualizza corsi -Visualizza dettagli corsi			
Cliente registrato	-Login -Modificare password -Visualizzare storico prenotazioni effettuate -Logout	-Visualizza corsi -Visualizza dettagli corsi	-Effettuare una prenotazione -Disdire una prenotazione -Confermare una prenotazione		
Gestore prenotazioni	-Login -Logout		-Modificare una prenotazione -Disdire una prenotazione		
Gestore Corsi	-Login -Logout			-Aggiunge un corso -Rimuove un corso	

				-Modifica un corso	
Amministratore	-Login -Logout	-Visualizza corsi -Visualizza dettagli corsi			-Aggiunta nuovo gestore prenotazioni -Aggiunta nuovo gestore corsi

2.5 Global software control

Il controllo software globale è di tipo event-driven. Il controllo risiede in un dispatcher che chiama le funzioni mediante callback. Il Web Server si occupa di gestire le richieste effettuate dagli utenti (client). Il server ordina le richieste ai Control (dunque Java Servlet), che gestiranno la richiesta eventualmente interagendo con i Model. Il Web Server aggiornerà poi le View che saranno visualizzate al client attraverso la creazione di codice HTML dalla pagina JSP.

2.6 Boundary Conditions

Le condizioni limite riguardano l'inizializzazione, la terminazione e le failure del sistema.

-Inizializzazione del sistema:

Identificativo: Start Server
Attori: Amministratore
Flusso di eventi:
1) L'amministratore decide di avviare il server, accede alla macchina Server e preme su 'Start'
2) Il sistema, con le opportune procedure di avvio, attiva il server e i relativi servizi in remoto rendendosi disponibile ad eventuali richieste.
3) Il sistema notifica il successo della procedura.
Condizione di uscita: L'amministratore visualizza la notifica di successo

-Terminazione del sistema:

Identificativo: Stop Server
Attori: Amministratore
Flusso di eventi:
1) L'amministratore decide di arrestare il server, accede alla macchina Server e preme su 'Stop'
2) Il sistema riceve la richiesta, effettua le operazioni ancora in corso e spegne il server

Condizione di uscita: L'amministratore visualizza la notifica di successo.

-Fallimenti del sistema:

Nel caso di fallimento hardware/software, si procede ad un riavvio del sistema. Per quanto riguarda invece la gestione dei dati persistenti, si procederà ad un backup periodico per minimizzare il rischio di perdita dovuto a guasti al DBMS.

Identificativo: Restart Server
Attori: Amministratore
Flusso di eventi:

1) L’amministratore decide di riavviare il server, accede alla macchina Server e preme su ‘Reset’
2) Il sistema riceve la richiesta, effettua le operazioni ancora in corso e riavvia il server
Condizione di uscita: L’amministratore visualizza la notifica di successo.

3 Subsystem Service

Sottosistema	**Descrizione**
Account	Consente ad un Utente non registrato di effettuare la registrazione e a un Utente registrato di effettuare il login. Permette l’inserimento e la rimozione del personale (Gestore ordine o Gestore prodotti) compreso login e logout.

Servizio	**Descrizione**
Login	Consente ad un Utente di accedere al sistema tramite username e password
Logout	Consente un Utente, Gestore prenotazioni, Gestori corsi e amministratore di rimuovere l’accesso al sistema
Login gestore prenotazioni	Consente ad un Gestore Prenotazioni di accedere alle rispettive funzionalità protette
Login gestore corsi	Consente ad un Gestore Corsi di accedere alle rispettive funzionalità protette
Registrazione utente	Permette ad un Utente non registrato di creare un account nel sistema
Modifica password utente	Consente ad un Utente registrato di modificare la password relativa al proprio account
Login Amministratore	Consente ad un Amministratore di accedere alle rispettive funzionalità protette
Registrazione istruttore	Consente ad un Amministratore di inserire le informazioni relative all’account di un Istruttore
Rimuovi istruttore	Consente ad un Amministratore di rimuovere le informazioni relative all’account di un Istruttore

Visualizzazione storico prenotazioni	Questa funzionalità permette al cliente di visualizzare lo storico delle prenotazioni effettuate.

Sottosistema	**Descrizione**
Corso	Permette l'aggiunta, la modifica e la rimozione di un corso da parte di un Gestore corsi. Consente inoltre la visualizzazione di tutto l'elenco dei corsi.

Servizio	**Descrizione**
Aggiungi Corso	Permette ad un Gestore corsi di inserire un nuovo corso alla categoria selezionata
Rimuovi Corso	Permette ad un Gestore corsi di rimuovere un corso
Modifica Corso	Permette ad un Gestore corsi di modificare un corso
Visualizza Corso	Permette di visualizzare i corsi presenti nel sito
Visualizzare Corsi in base alla categoria	Permette al cliente (registrato e non) di visualizzare i corsi della categoria selezionata

Sottosistema	**Descrizione**
Prenotazione	Consente di inserire i dati di pagamento e ad un Utente registrato per completare una prenotazione. Permette la conferma o la disdetta di una prenotazione.

Servizio	**Descrizione**
Conferma Prenotazione	Permette ad un Gestore prenotazioni di poter confermare una prenotazione.
Disdire Prenotazione	Permette ad un Gestore prenotazioni di poter disdire una prenotazione.
Visualizzare Prenotazione	Permette ad un utente registrato di poter visualizzare la prenotazione
Inserire dati pagamento	Permette di inserire i dati della carta di credito, necessari per effettuare il checkout