**Bodybuild**

Ingegneria del software e progettazione web Progetto A.A. 2024/25

*0307348, 0309540 Adamo Luca, De Santis Marco*

**Indice**

[1. Specifica dei requisiti software 3](#_Toc202862963)

[1.1. Introduzione 3](#_Toc202862964)

[1.1.1. Scopo del documento 3](#_Toc202862965)

[1.1.2. Panoramica del sistema definito 3](#_Toc202862966)

[1.1.3. Requisiti HW e SW 3](#_Toc202862967)

[1.1.4. Sistemi correlati 4](#_Toc202862968)

[1.2. User stories 4](#_Toc202862969)

[Adamo Luca 4](#_Toc202862970)

[US 1 4](#_Toc202862971)

[US 2 4](#_Toc202862972)

[US 3 4](#_Toc202862973)

[De Santis Marco 4](#_Toc202862974)

[US 1 4](#_Toc202862975)

[US 2 5](#_Toc202862976)

[US 3 5](#_Toc202862977)

[1.3. Functional Requirements 5](#_Toc202862978)

[Adamo Luca 5](#_Toc202862979)

[FR 1 5](#_Toc202862980)

[FR 2 5](#_Toc202862981)

[FR 3 5](#_Toc202862982)

[De Santis Marco 5](#_Toc202862983)

[FR 1 5](#_Toc202862984)

[FR 2 5](#_Toc202862985)

[FR 3 5](#_Toc202862986)

[1.4. Use cases 6](#_Toc202862987)

[1.4.1. Overview diagram 6](#_Toc202862988)

[1.4.2. Internal steps 6](#_Toc202862989)

[Richiedi modifica scheda 6](#_Toc202862990)

[Estensioni: 7](#_Toc202862991)

[Prenota corso 7](#_Toc202862992)

[Estensioni: 7](#_Toc202862993)

[2. Storyboard 8](#_Toc202862994)

[3. Design 14](#_Toc202862995)

[3.1. Class Diagram 14](#_Toc202862996)

[3.1.1. VOPC (View Of Partecipating Classes) 14](#_Toc202862997)

[Richiedi modifica 14](#_Toc202862998)

[Prenota Corso 15](#_Toc202862999)

[3.1.2. Design-Level diagram 16](#_Toc202863000)

[3.1.2.1. Richiedi modifica 16](#_Toc202863001)

[3.1.2.2. Prenota corso 17](#_Toc202863002)

[3.2. Design pattern 18](#_Toc202863003)

[3.3. Activity Diagram 19](#_Toc202863004)

[3.3.1. Richiedi modifica 19](#_Toc202863005)

[3.3.2. Penota Corso 20](#_Toc202863006)

[3.4. Sequence Diagram 21](#_Toc202863007)

[3.4.1. Richiedi modifica 21](#_Toc202863008)

[3.4.2. Prenota Corso 22](#_Toc202863009)

[3.5. State Diagram 22](#_Toc202863010)

[3.5.1. Richiedi modifica 22](#_Toc202863011)

[3.5.2. Prenota Corso 23](#_Toc202863012)

[4. Testing 23](#_Toc202863013)

[4.1. Test effettuati da Adamo Luca 23](#_Toc202863014)

[4.2. Test effettuati da Desantis Marco 23](#_Toc202863015)

[5. Sonarcloud 23](#_Toc202863016)

# Specifica dei requisiti software

## Introduzione

### Scopo del documento

L’obiettivo del documento è quello di fornire una panoramica dettagliata del processo di sviluppo del progetto “Bodybuild” e di tutte le fasi di progettazione, dai requisiti alla fase di testing. Viene fornita una panoramica delle funzionalità e dell’architettura del progetto con l’utilizzo di strumenti come il linguaggio di modellazione UML e specificando i requisiti hardware e software necessari per il corretto funzionamento dell’applicazione. Il progetto è sviluppato per il supporto della gestione di una palestra e le relative richieste e prenotazioni come progetto accademico per il corso di Ingegneria del software e progettazione web nell’anno accademico 2024/25.

### Panoramica del sistema definito

“Bodybuild” è un sistema software progettato per il supporto delle palestre e dei personal trainer per poter gestire le richieste in maniera più efficiente e fornire agli utenti un’applicazione che gli semplifichi loro tutto ciò che può servigli per gli allenamenti, offrendo un’interfaccia intuitiva, funzionale e semplice da utilizzare. Nell’applicazione ogni utente viene gestito in base al proprio ruolo per poter offrire delle funzionalità specifiche. L’utente può:

* Inserire e aggiornare i propri dati personali
* Prenotarsi ai corsi in modo semplice
* Richiedere una scheda di allenamento personalizzata
* Richiedere modifiche alla propria scheda di allenamento per poterla adattare al massimo ai proprio bisogni, obiettivi ed esigenze
* Visualizzare i propri progressi nel corso del tempo
* Visualizzare tutte le proprie schede richieste con la visualizzazione in ogni momento di tutti gli esercizi sempre a portata di mano potendo visualizzare per ciascuno di essi anche la relativa esecuzione tramite video o descrizione
* Modificare la prenotazione per qualunque problema

Il personal trainer può:

* Accettare le richieste delle schede ed effettuare le relative modifiche
* Creare schede di allenamento ad hoc per ogni utente
* Visualizzare i progressi degli utente con le sue schede di allenamento

Abbiamo previsto anche la comunicazione tra utente e personal trainer per un supporto diretto ed un aiuto veloce ed immediato, che però non abbiamo inserito per ragioni di tempo.

### Requisiti HW e SW

Requisiti Hardware Minimi:

* Processore: 1 GHz, consigliato 2 GHz
* Memoria (RAM): 2 GB, consigliato 4 GB
* Hard disk: 3 GB, consigliato 5 GB
* Connessione Internet: Necessaria per l'accesso ai servizi di terze parti.

Requisiti Software:

* Sistema Operativo:
  + Windows 7 o successivo
  + macOS 10.15 o successivo
  + distribuzioni Linux con kernel 5.0 o superiore
  + Java Runtime Environment (JRE): Versione 17 o successiva

### Sistemi correlati

* *McFit*

Pro: Ti permette di avere pieno controllo di tutti i dati del contratto, puoi richiedere sospensioni e periodi di pausa a tuo piacimento, con un programma di fedeltà ti permette di avere dei vantaggi. Inoltre puoi anche controllare l’affollamento attuale della palestra.

Contro: Il controllo dell’affollamento si basa solo su dati attuali e non su dati storici per darti una idea più ampia, inoltre non hai contatto diretto con i personal trainer per la richiesta, modifica e visualizzazione delle schede e delle esecuzioni degli esercizi perché si basa maggiormente sulla gestione del contratto e dell’iscrizione

* *Virgin Active*

Pro: Ti permette di prenotarti ad una vastissima scelta di corsi, di monitorare i propri progressi se registrati, di gestire tutti i dati contrattali, e di avere per alcuni livelli di fedeltà il servizio on-demand per poter seguire i corsi comodamente da casa.

Contro: L’accesso ad alcune funzionalità non è garantito a tutti, e anche in questo caso non c’è nessuna opportunità di contatto diretto con i personal trainer per una gestione personalizzata e richiesta di schede di allenamento se non i contatti generali della palestra.

## User stories

### Adamo Luca

#### US 1

Come utente voglio poter vedere le dimostrazioni degli esercizi presenti nella mia scheda di allenamento\* per poterli eseguire correttamente. *(Non implementato)*

#### US 2

Come personal trainer voglio poter accedere per vedere il numero di persone prenotate ad un corso seguito da me. *(Non implementato)*

#### US 3

Come utente voglio poter richiedere modifiche alla mia scheda di allenamento per potermi allenare al meglio.

\*scheda di allenamento: scheda contenente l’elenco di esercizi di ogni allenamento previsto, con il relativo ordine, quante ripetizioni fare per ciascun esercizio, con indicato il recupero tra gli esercizi, e quante volte ripetere un esercizio

### De Santis Marco

#### US 1

Come cliente, voglio poter visualizzare l'elenco dei corsi offerti, per prenotare il corso di mio interesse.

#### US 2

Come cliente, voglio lasciare recensioni sul corso a cui ho partecipato, per condividere la mia esperienza. *(Non implementato)*

#### US 3

Come cliente, voglio poter visualizzare la lista delle prenotazioni effettuate, così da poter tenere traccia delle attività a cui sono iscritto. *(Non implementato)*

## Functional Requirements

### Adamo Luca

#### FR 1

Il sistema dovrà garantire a un utente di poter accedere al proprio account personale utilizzando username e password

#### FR 2

Il sistema permetterà ad un utente di modificare la prenotazione di un allenamento.

#### FR 3

Il sistema invierà all’utente una mail qualora venga confermata ed effettuata la richiesta di una scheda. *(Non implementato)*

### De Santis Marco

#### FR 1

Il sistema deve offrire un profilo personale dove gli utenti possono visualizzare e aggiornare le proprie informazioni

#### FR 2

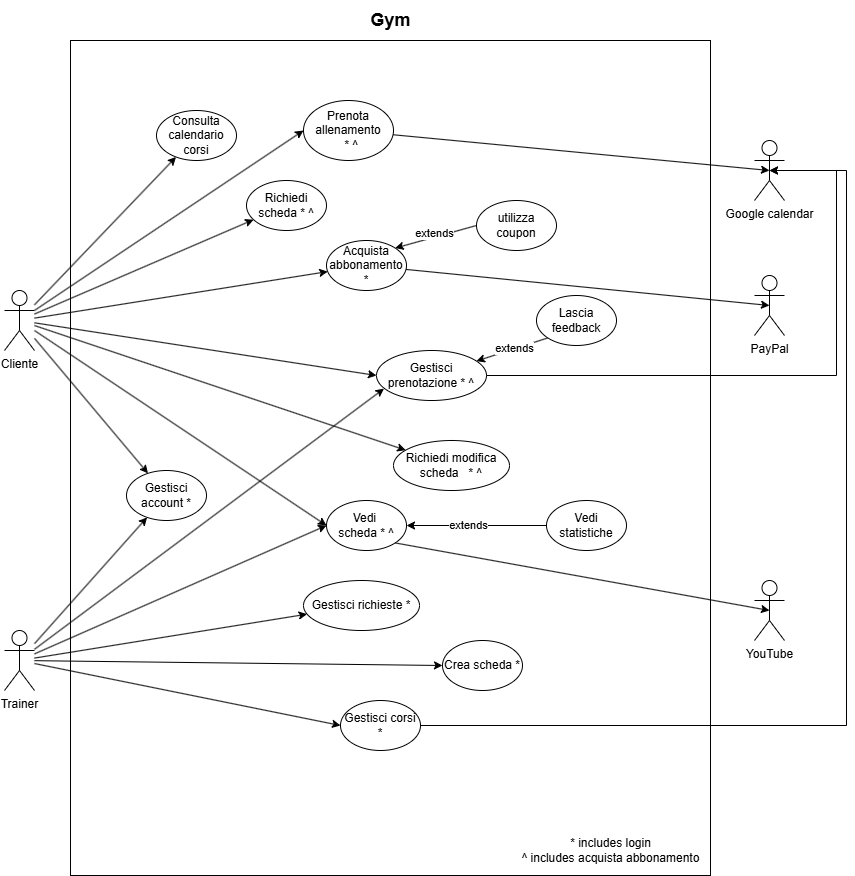
Il sistema deve notificare l'utente con un messaggio a schermo un giorno prima della *sessione d’allenamento*. *(Non implementato)*

#### FR 3

Il sistema deve mostrare all’utente una lista di tutti i corsi disponibili

## Use cases

### Overview diagram



Nota: Sono presenti nell’implementazione reale: login, richiedi modifica scheda, gestisci richieste e prenota allenamento

### Internal steps

#### Richiedi modifica scheda

1. Il caso d’uso inizia quando l’utente visualizza la scheda e seleziona “richiedi modifica scheda”
2. Il sistema mostra una schermata con le varie schede selezionabili
3. L’utente selezionerà la scheda su cui vorrebbe apportare una modifica
4. L’utente indica di aver completato e va alla selezione dell’esercizio
5. Il sistema mostra una schermata con i vari esercizi selezionabili
6. L’utente selezionerà l’esercizio da voler modificare
7. Il sistema mostra un form per permettere all’utente di inserire la motivazione
8. L’utente inserisce la propria motivazione per la relativa modifica
9. L’utente indica di aver completato ed invia la richiesta
10. Il sistema notifica il personal trainer
11. Il caso d’uso termina

#### Estensioni:

1. Fallisce l’interazione con il database: il sistema segnala l’errore e chiede di riprovare
2. Non viene portato alla schermata successiva: l’utente non ha selezionato la scheda
3. Fallisce l’interazione con il database: il sistema segnala l’errore e chiede di riprovare
4. Non viene inviata la richiesta: l’utente non ha inserito la motivazione

Nota: La notifica al personal trainer non è stata implementata per vincoli di tempo

#### Prenota corso

1. Il cliente procede nell’effettuare il login.

2. Il sistema visualizza la schermata principale per gli utenti autenticati.

3. Il cliente seleziona la sezione relativa alla prenotazione dei corsi.

4. Il sistema mostra una lista dei corsi disponibili.

5. Il cliente seleziona un corso di interesse.

6. Il cliente seleziona il giorno e l’orario tra quelli disponibili.

7. Il sistema verifica se ci sono posti disponibili.

8. Il cliente seleziona il pulsante per inviare la richiesta al trainer.

9. Il sistema invia la richiesta e mostra un messaggio di conferma al cliente.

#### Estensioni:

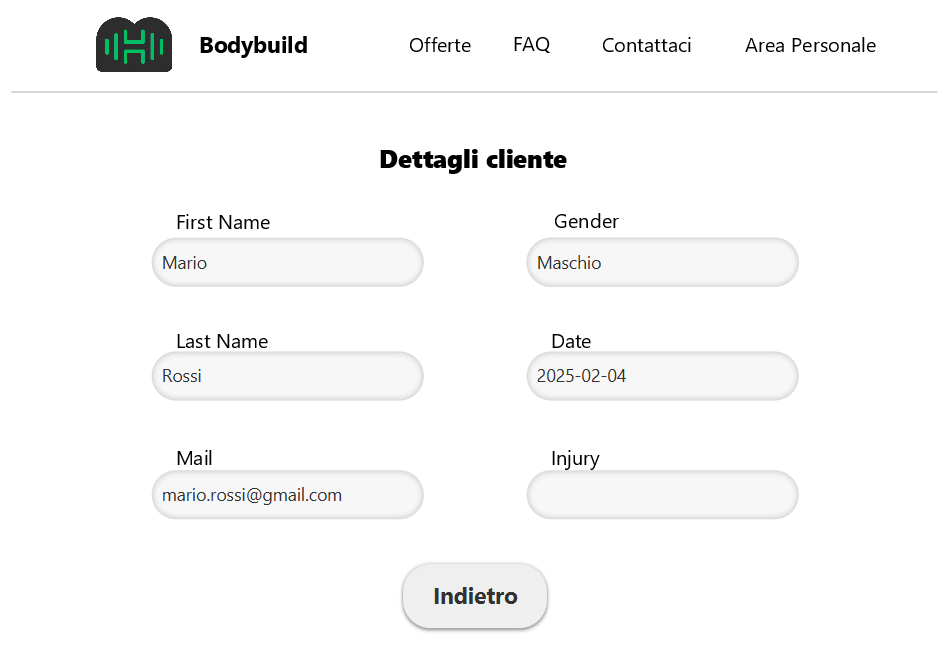
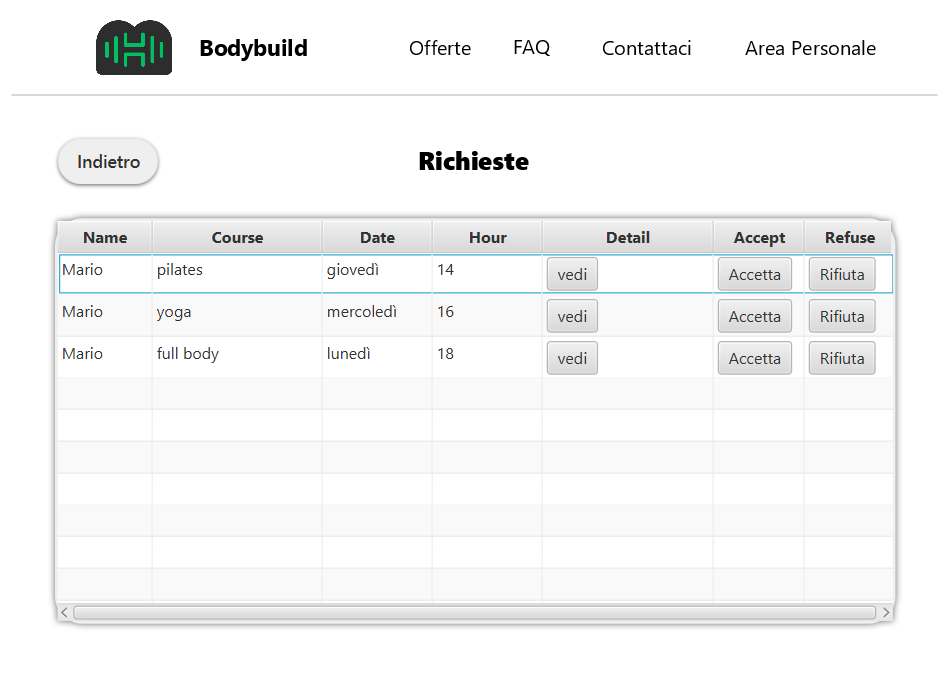
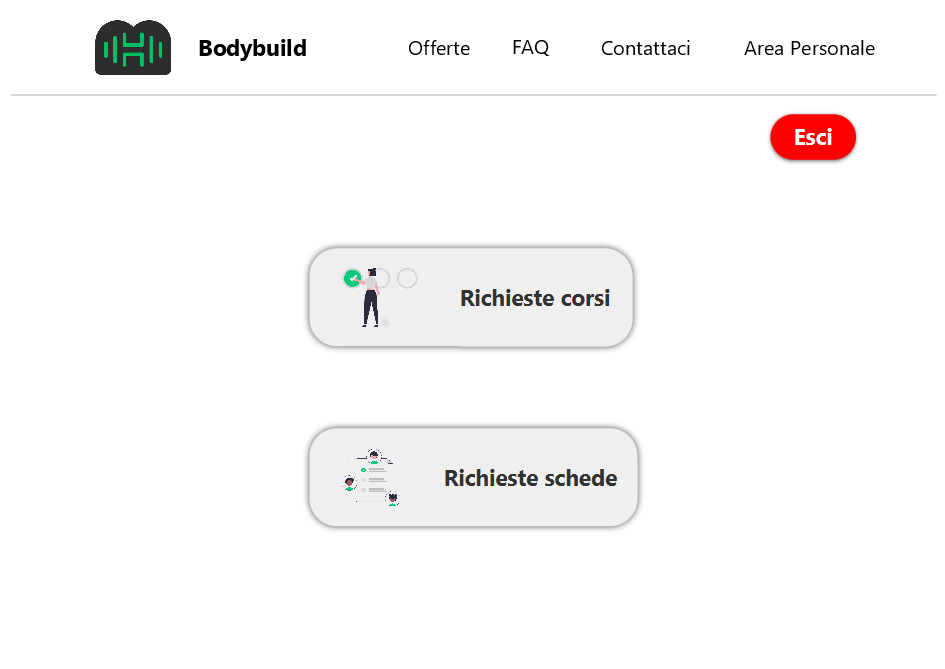
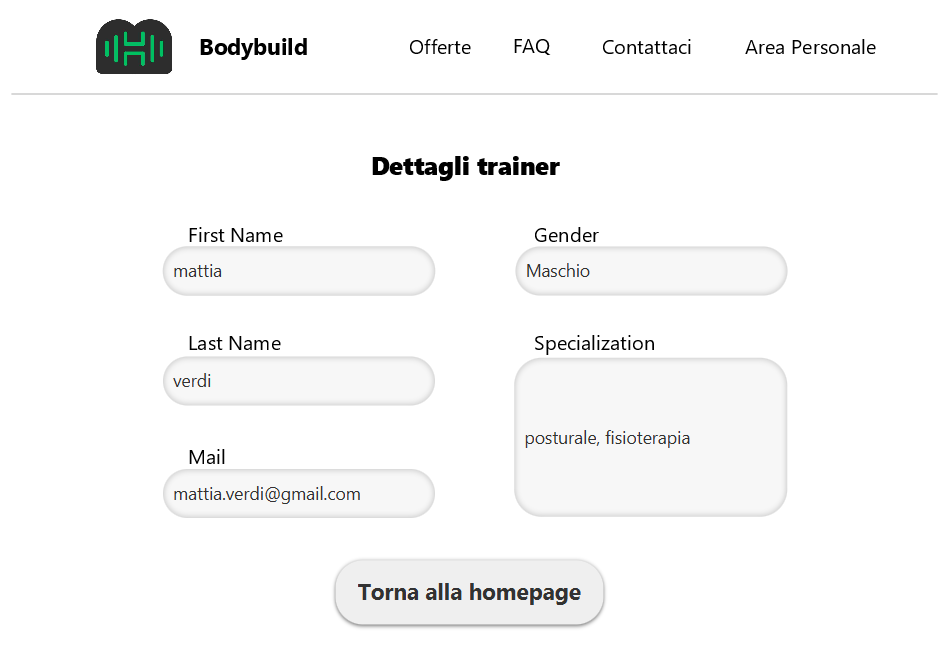
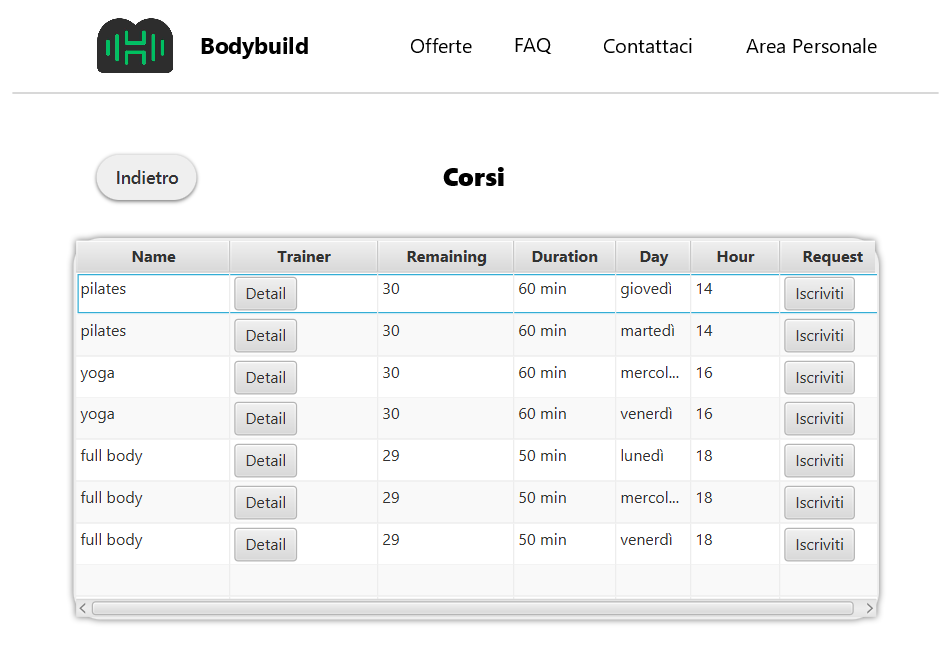
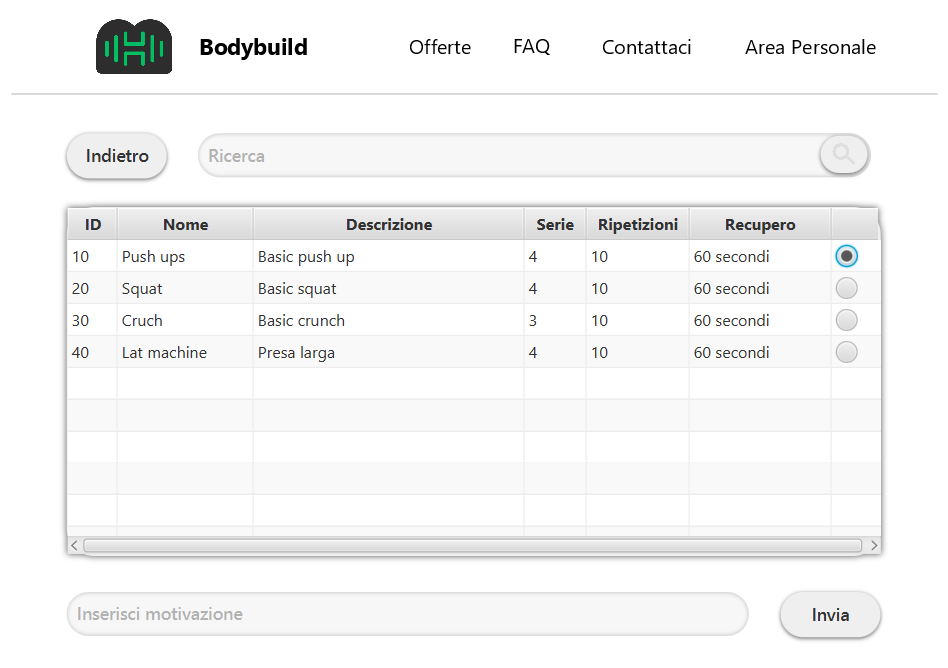
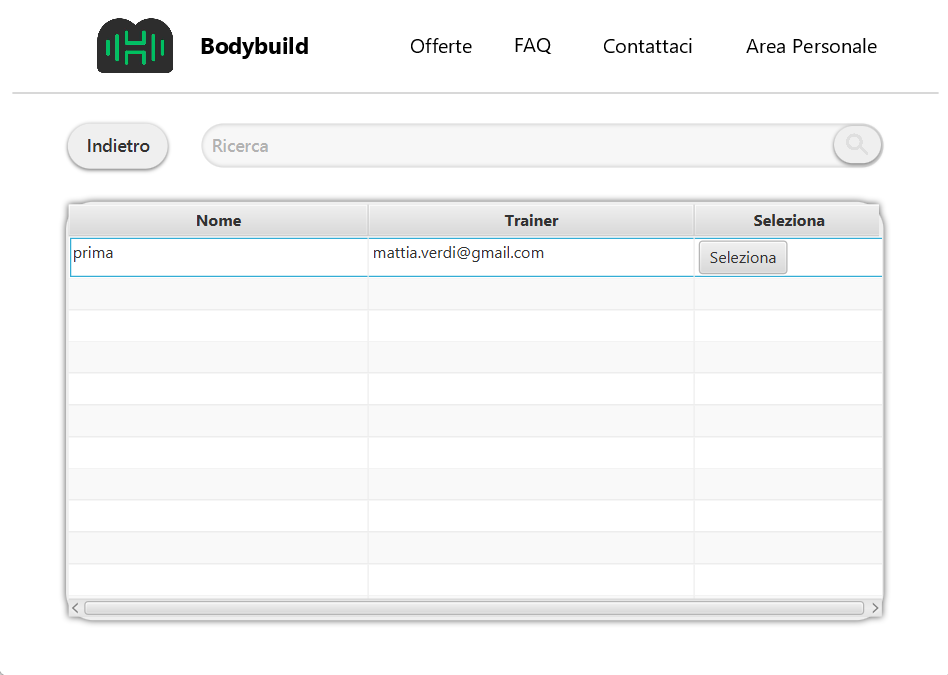
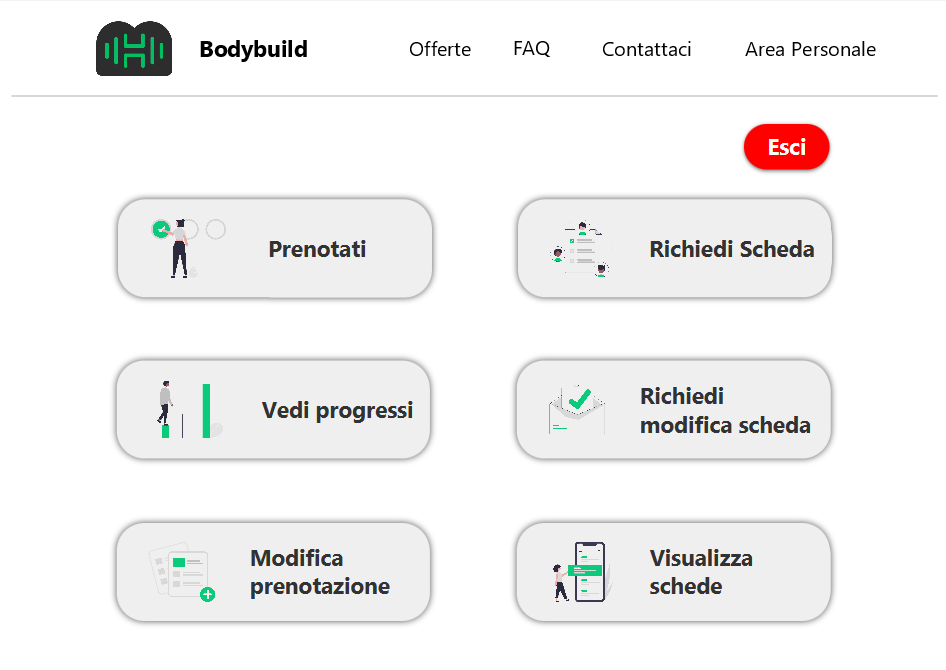
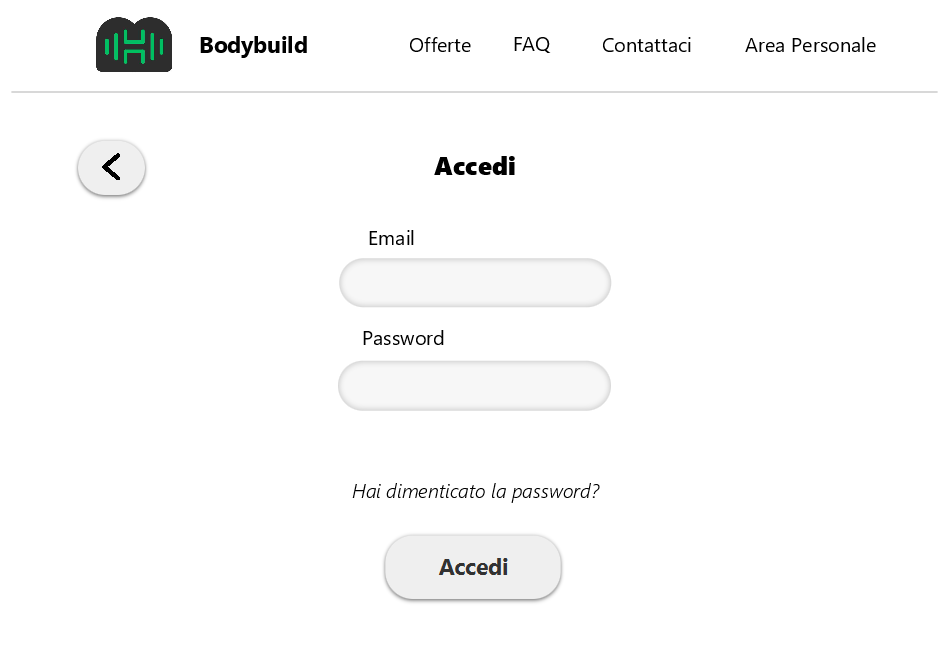
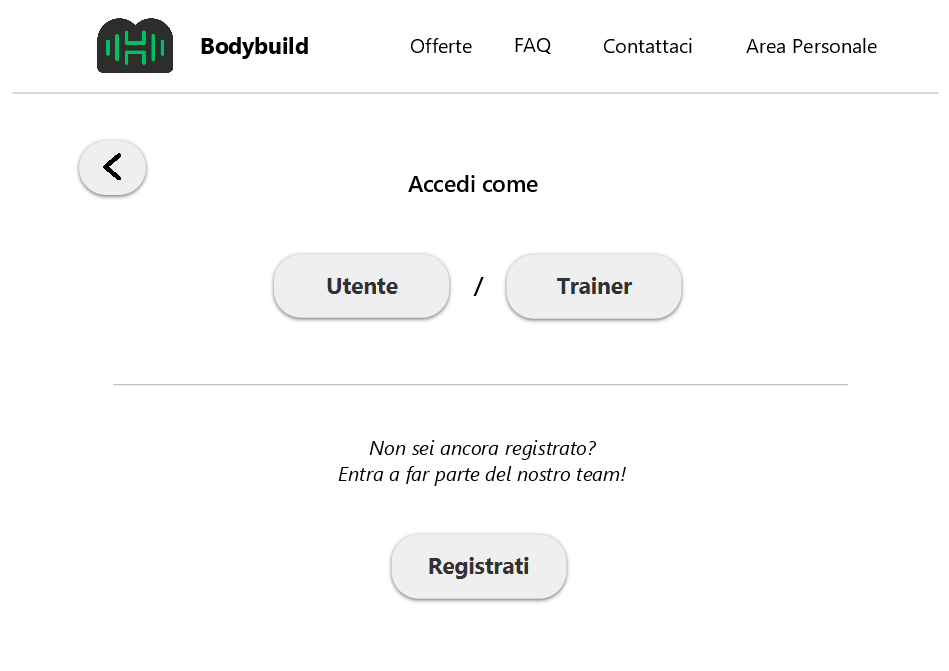
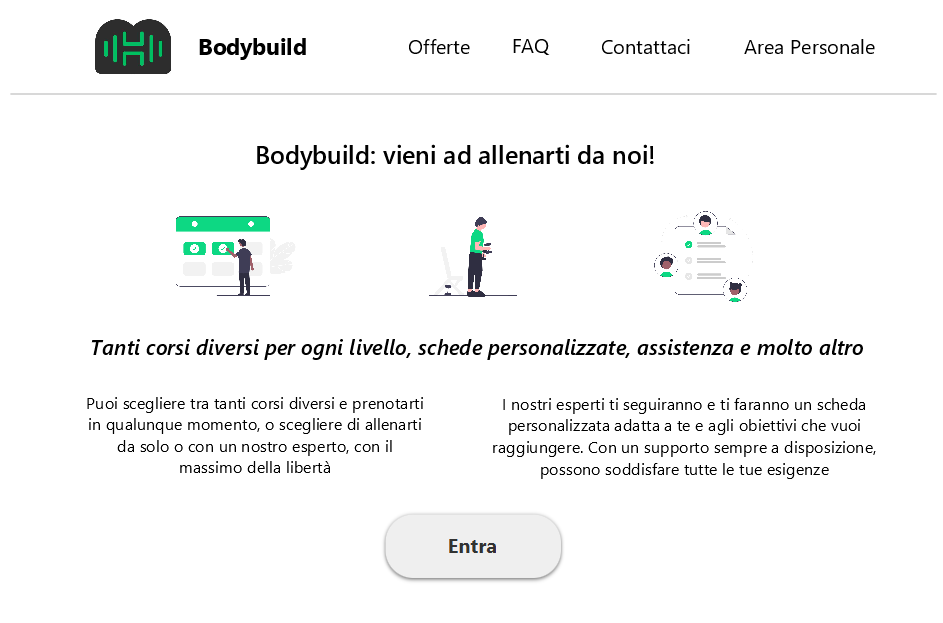
1a. Il cliente non è registrato: Il sistema notifica l’utente e lo invita a registrarsi.

4a. Non ci sono corsi disponibili: il sistema notifica l'utente e torna al punto 2

7a. Non ci sono posti disponibili: Il sistema notifica l'utente e torna al punto 4

8a. Richiesta già inviata: il sistema notifica l'utente e torna al punto 4

# Storyboard

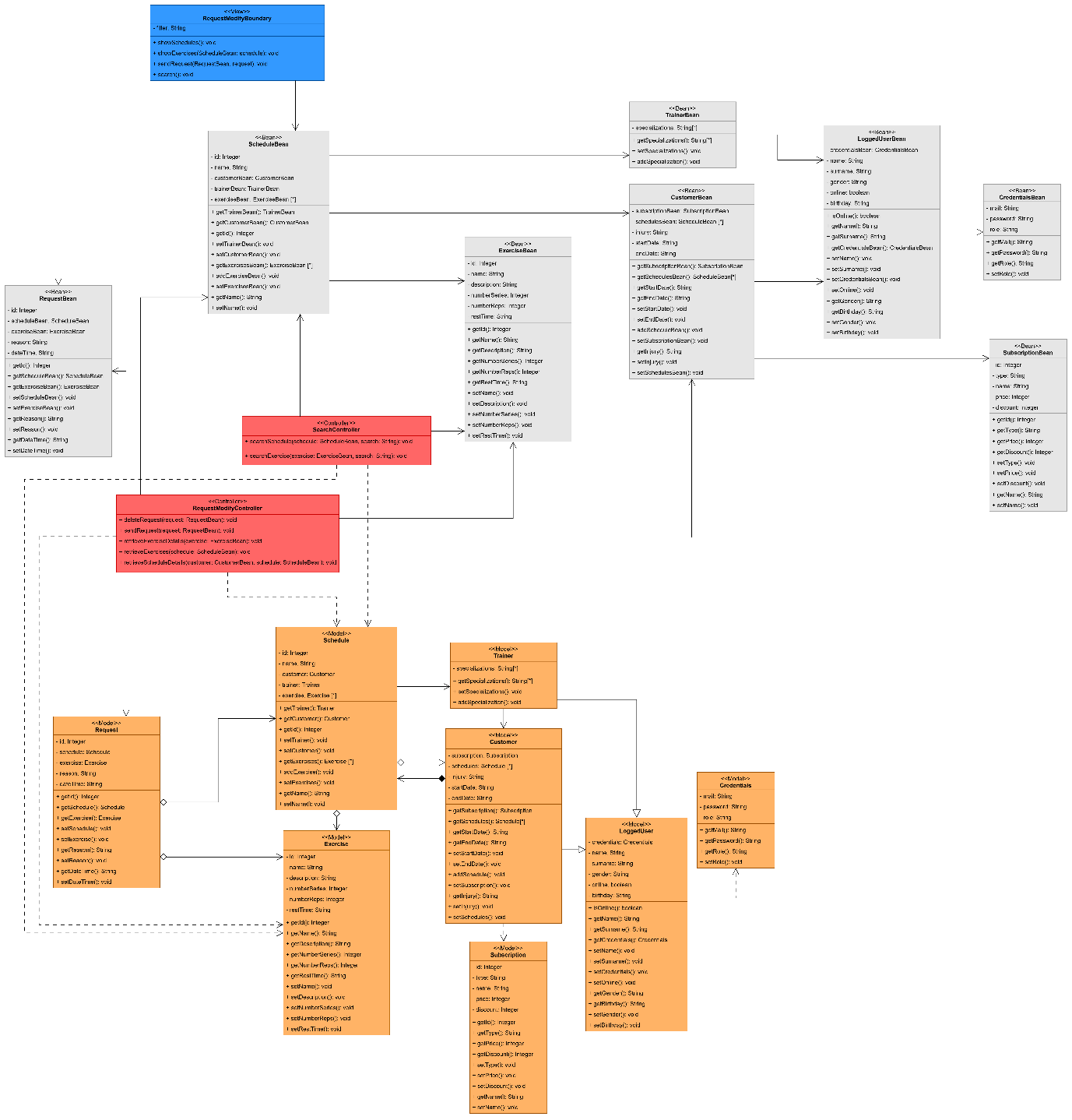


# Design

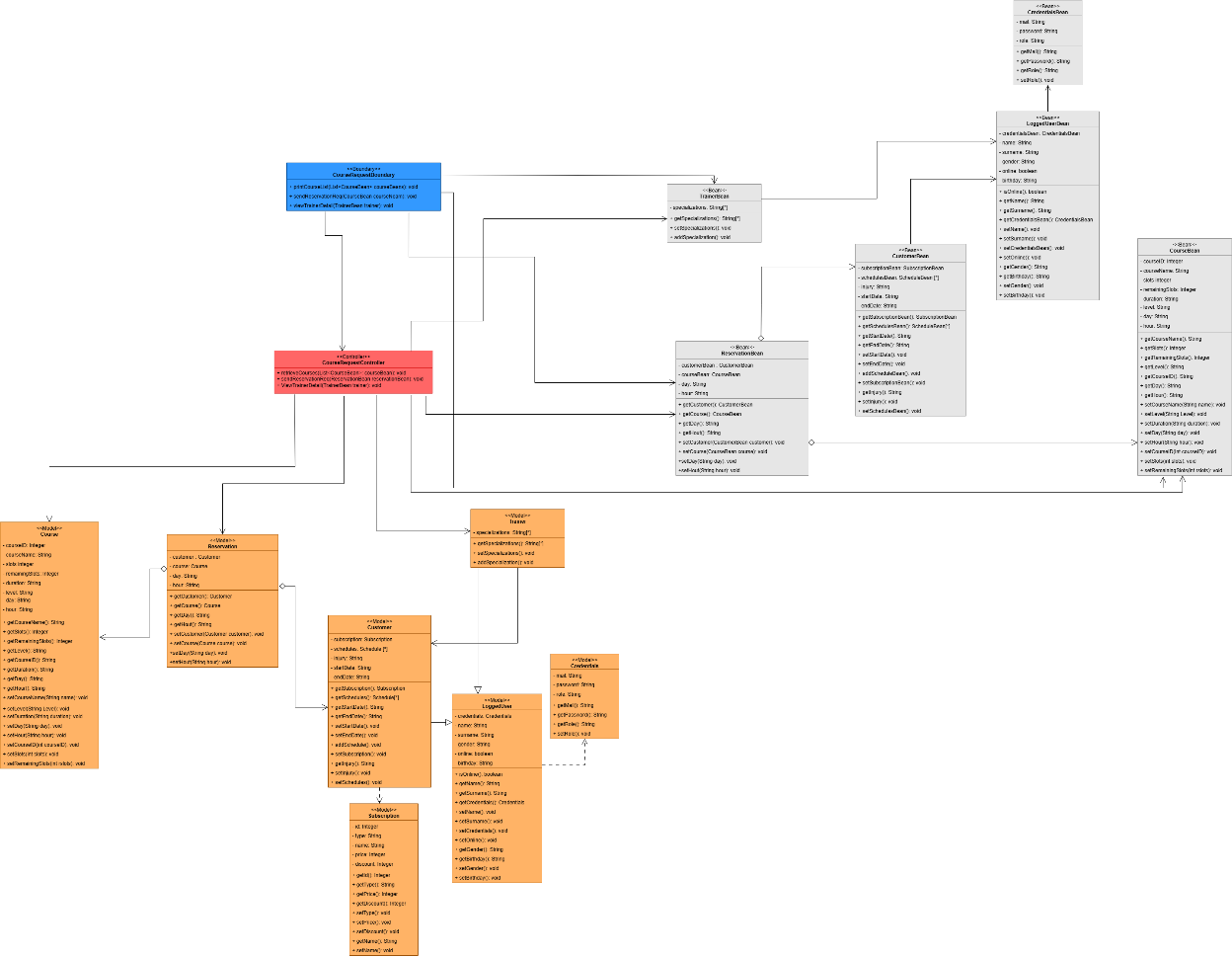
## Class Diagram

### VOPC (View Of Partecipating Classes)

#### Richiedi modifica

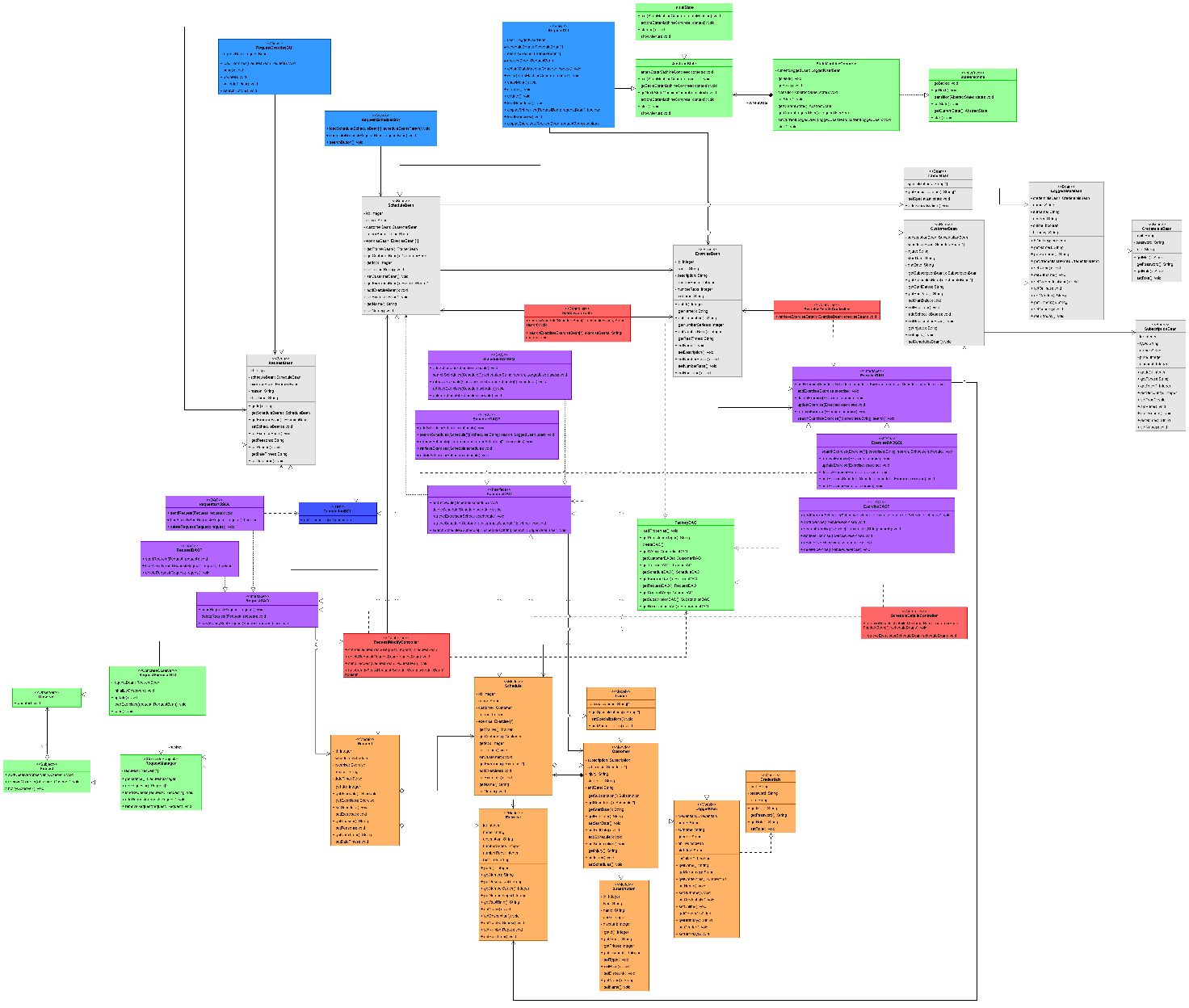


#### Prenota Corso

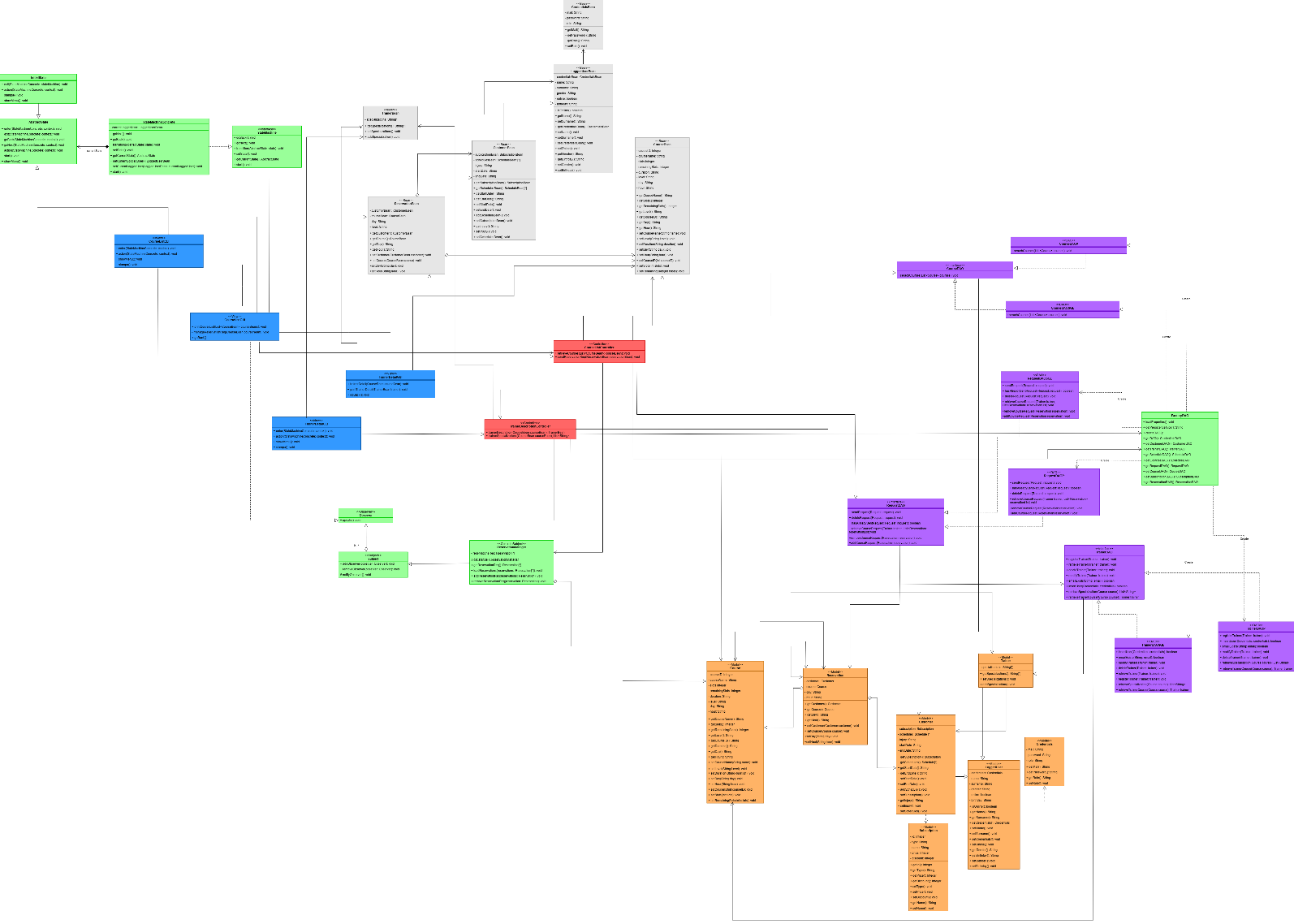


### Design-Level diagram

#### Richiedi modifica



#### Prenota corso



## Design pattern

I pattern GoF utilizzati nel progetto sono:

**Pattern singleton per la versione demo, per i DAO e per i mapper:**

È stato utilizzato per permettere di avere una gestione centralizzata dei dati in memoria, memorizzando i dati in un’unica istanza e garantendo quindi la coerenza delle informazioni, poiché tutti i componenti dell’applicazione accederanno allo stesso set di dati. Nel caso dei DAO ogni DAO specifico viene gestito come un Singleton all'interno della FactoryDAO, assicurando che per ogni tipo di DAO esista una sola istanza nell'applicazione. Invece nel caso dei mapper permette sempre di avere un punto unico e coerente per tutte le operazioni di mappatura all'interno dell'applicazione, garantendo che le conversioni siano gestite in modo uniforme.

**Pattern Observer per le richieste di modifiche e per le richieste dei corsi:**

È stato utilizzato nel caso delle richieste delle modifiche e nelle richieste dei corsi per soddisfare la stessa esigenza di due funzioni diverse, per permettere che la lista delle richieste quando viene visualizzata dal trainer verrà aggiornata nel caso in cui arrivino altre richieste, solo che nel caso delle richieste delle modifiche vengono solo aggiunte le richieste poiché non essendo stato implementato l’aspetto del trainer di gestire le richieste di modifiche, non è stato nemmeno implementato l’aggiornamento della lista tramite observer di conseguenza. Viene utilizzato per garantire che le interfacce utente (GUI) e altri componenti siano automaticamente aggiornati in risposta a modifiche nello stato delle richieste (Request) e delle prenotazioni (Reservation), così che quando viene notificato un cambiamento le interfacce utente possono aggiornarsi.

**Pattern State per l’interfaccia CLI:**

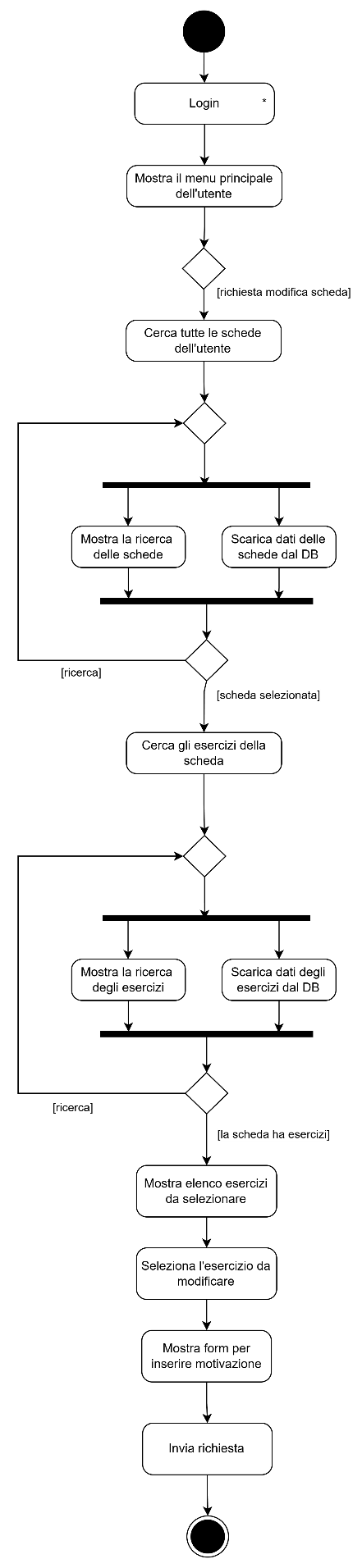
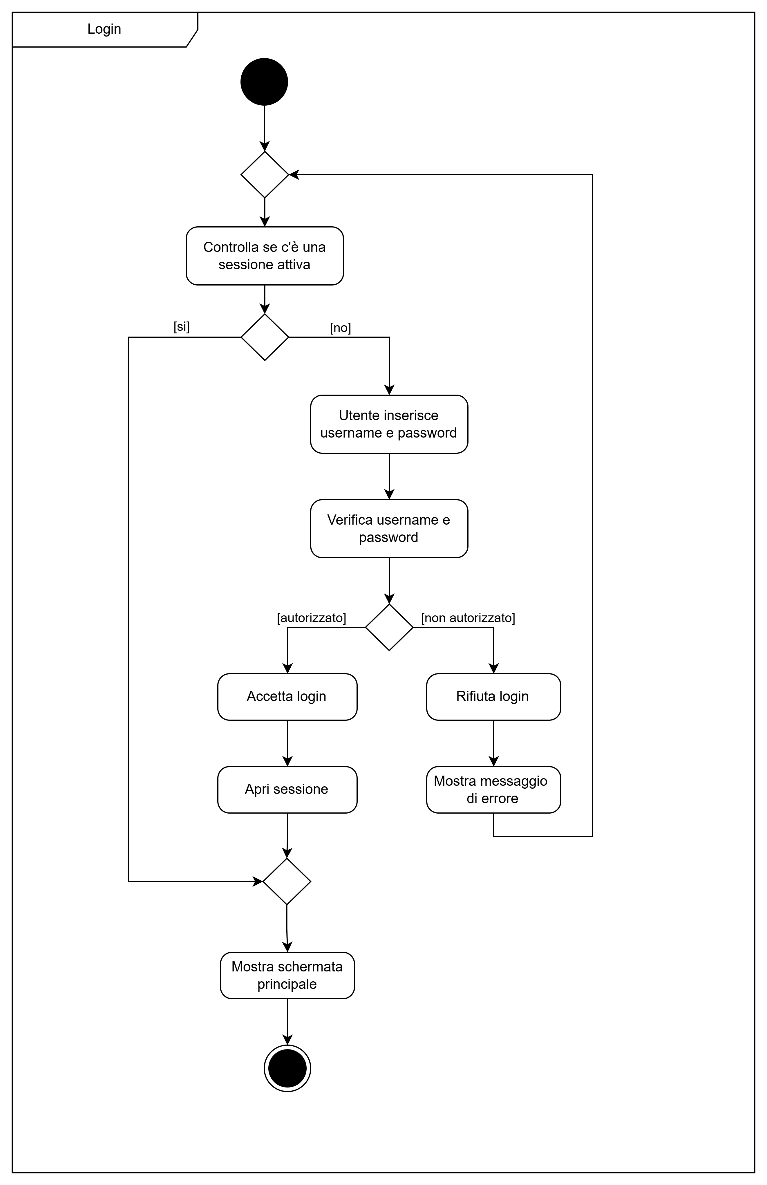
È stato utilizzato per gestire la navigazione e il comportamento dell'applicazione a riga di comando (CLI) in base allo stato corrente dell'utente, perché permette una navigazione più semplice facilitando la gestione di schermate e menu visto che il pattern appunto permette di cambiare il comportamento di un oggetto quando cambia il suo stato che con le transizioni esplicite rende il flusso più semplice, disaccoppiando la logica all’interno di una schermata, dall’implementazione della transizione o della stampa del menu.

**Pattern Factory nella creazione dei DAO e dei mapper:**

È stato utilizzato perchè fornisce un'interfaccia per creare oggetti in una superclasse, ma permette alle sottoclassi di creare diverse implementazioni dei DAO(es. per SQL, per la demo in memoria, o per JSON), così da centralizzare la creazione di oggetti DAO e gestire in maniera centralizzata la strategia di persistenza dell’intera applicazione. Nel caso dei mapper i metodi fromBeanToModel() e fromModelToBean() fungono da "metodi factory" che recuperano il mapper appropriato dal registro della factory in base ai tipi di oggetti coinvolti nella conversione, che ogni mapper concreto dovrà implementare per specificare le regole di conversione.

## Activity Diagram

### Richiedi modifica

### Penota Corso

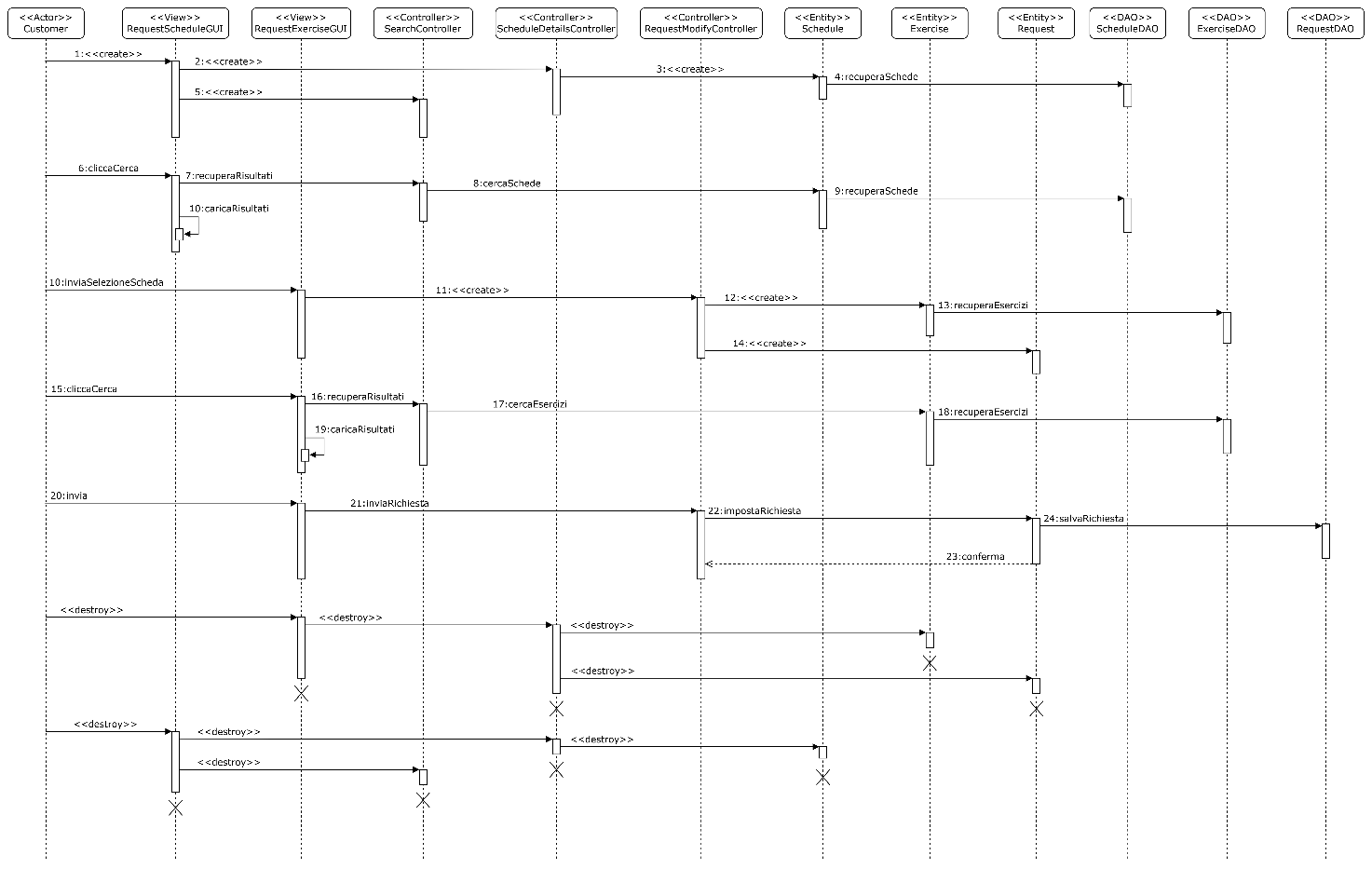
Immagine che contiene diagramma, schizzo, Disegno tecnico, Piano

Il contenuto generato dall'IA potrebbe non essere corretto.

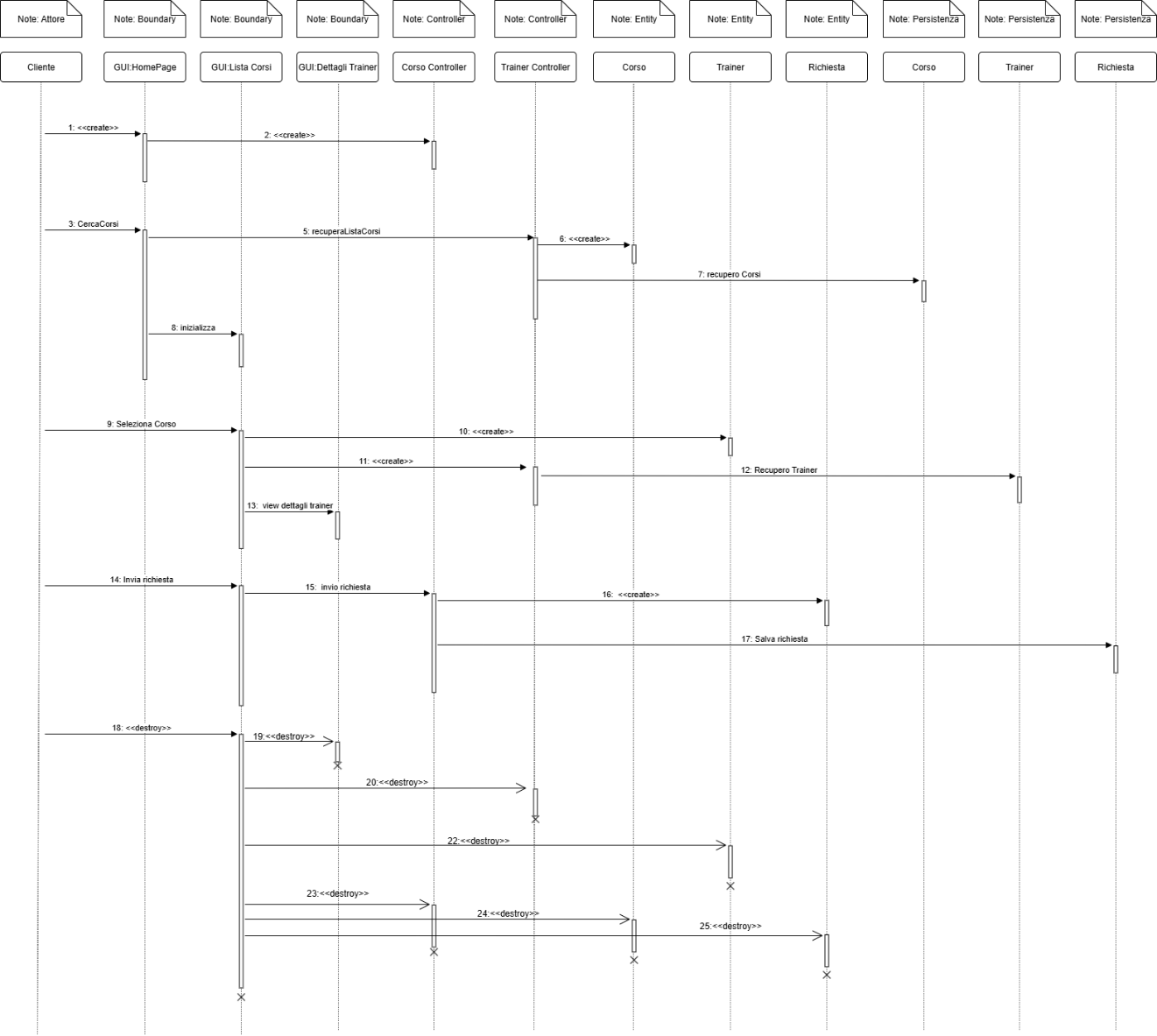
Nota: la notifica del cliente non è stata implementata per i vincoli di tempo.

## Sequence Diagram

### Richiedi modifica

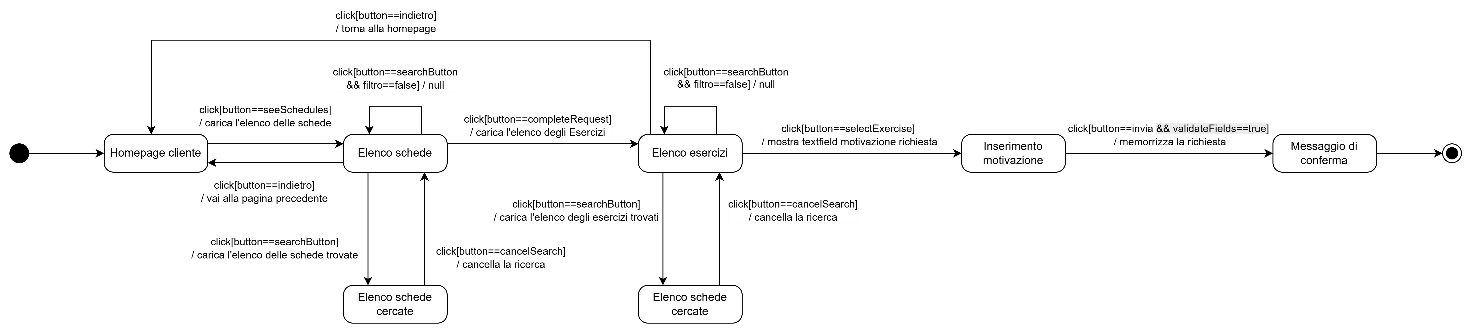


### Prenota Corso

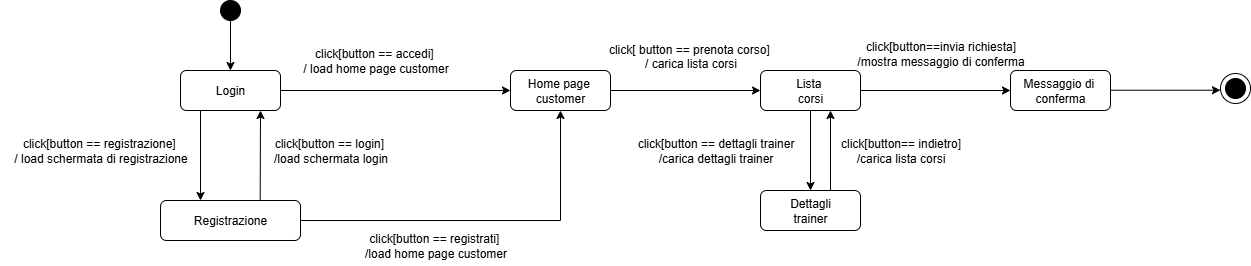


## State Diagram

### Richiedi modifica



### Prenota Corso



# Testing

## Test effettuati da Adamo Luca

1. Verifica la buona riuscita del login per il cliente.
2. Verifica che le schede per il cliente vengano recuperate correttamente.
3. Verifica che la richiesta di una modifica fatta dal cliente venga salvata correttamente.

## Test effettuati da Desantis Marco

1. Verifica che i corsi vengano recuperati correttamente
2. Verifica la corretta registrazione della richiesta al corso
3. Verifica che le prenotazioni vengano recuperate correttamente

# Sonarcloud

Il link di sonarcloud associato al progetto è:

<https://sonarcloud.io/summary/new_code?id=Marco17-2_ISPW2&branch=master>