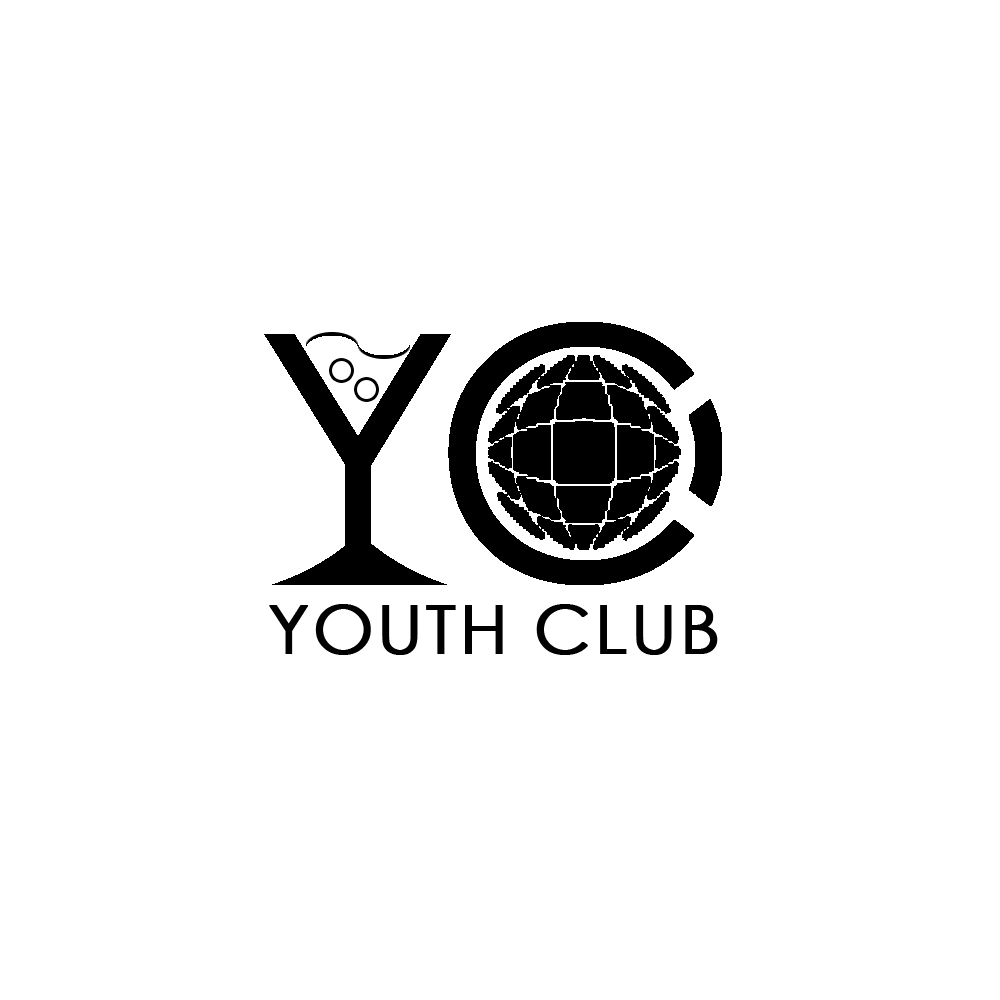
**My Pocket Gym**

**Requirement Analysis**

**Document**

****

***DATA: 17/11/2020***

**Partecipanti del progetto:**

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Vincenzo Liguorino | 0522500840 |
| Aniello Petrosino | 0522501013 |
| Francesco Truono |  |

|  |  |
| --- | --- |
| **Scritto da:** | Tutto il team |

Sommario

[**1.**](#_heading=h.gjdgxs) **Introduzione** 4

[1.1.](#_heading=h.1fob9te) Proposta del sistema 4

[1.2.](#_heading=h.1ci93xb) Scopo del sistema 4

[1.3.](#_heading=h.3znysh7) Ambito del sistema 4

[**2.**](#_heading=h.2et92p0) **Sistema proposto** 5

[2.1.](#_heading=h.tyjcwt) Overview 5

[2.2.](#_heading=h.3dy6vkm) Requisiti funzionali 5

[2.3.](#_heading=h.1t3h5sf) Requisiti non funzionali 5

[2.3.1.](#_heading=h.3whwml4) High Accurancy 5

[2.3.2.](#_heading=h.2bn6wsx) Adaptability 6

[2.3.3.](#_heading=h.4d34og8) Usability 6

[2.3.4.](#_heading=h.2s8eyo1) Reliability 6

[2.3.5.](#_heading=h.17dp8vu) Performance 6

[2.3.6.](#_heading=h.3rdcrjn) Supportability 6

[2.4.](#_heading=h.26in1rg) System model 6

[2.4.1.](#_heading=h.lnxbz9) Scenari 6

[2.4.2.](#_heading=h.35nkun2) Use case model 9

[2.4.3.](#_heading=h.qsh70q) Use case diagram 14

[2.4.4.](#_heading=h.3as4poj) Object model 15

[2.4.5.](#_heading=h.1pxezwc) Class Diagram 16

[2.4.6.](#_heading=h.4i7ojhp) Sequence diagram 17

[2.4.7.](#_heading=h.49x2ik5) Statechart diagram 20

[2.4.8.](#_heading=h.2xcytpi) Mock-up 20

# **Introduzione**

## Proposta del sistema

Il sistema funge da hub virtuale che agevola l’incontro tra amanti del fitness, personal trainer e nutrizionisti provenienti da tutta Italia.

Il sistema è stato creato per tutte le persone che sono alla ricerca di un personal trainer e di avere gli strumenti per essere seguiti da quest’ultimi.

Per semplificare l’usabilità del prodotto, l’app sarà disponibile sui due principali sistemi operativi per mobile: Android e iOS.

Come le classiche applicazioni di fitness, del settore, permette di avere delle statistiche sulle attività di allenamento ed essere seguito passo passo da un personal trainer e un nutrizionista.

## Ambito del sistema

Il sistema si rivolgerà a professionisti del settore quali **personal trainer**, **nutrizionista** e **dietologi** che intendono ampliare la loro clientela o gestire e portare su una piattaforma centralizzata i propri clienti che praticano attività fisica come **appassionati di fitness**, **amatori** o **professionisti sportivi**.

Le funzioni, offerte sono:

* Registrazione e login in base al ruolo
* Gestione delle schede di allenamento (creazione e modifica)
* Supporto all’allenamento
* Gestione di eventi
* Statistiche fisiche e d’allenamento

# **Sistema proposto**

## Overview

Il Requirement Analysis Document si concentra su:

* + Requisiti funzionali
  + Requisiti non funzionali
  + Scenari
  + Use case model
  + Use case diagram
  + Sequence diagram

## Requisiti funzionali

**Utente**

1. **FR1:** l’utente deve avere la possibilità di visualizzare la settimana d’allenamento e la sua dieta rapidamente
2. **FR2:** l’utente ha la possibilità di scegliere un personal trainer e un nutrizionista
3. **FR3:** l’utente può visualizzare la sue schede d’allenamento, compreso i singoli giorni.
4. **FR4:** l’utente seleziona un giorno di allenamento e permette di avviare l’allenamento
5. **FR5:** l’utente ha la possibilità di vedere i suoi dati personali e lo status dell’abbonamento
6. **FR6:** l’utente ha la possibilità di visualizzare la propria dieta
7. **FR7:** l’utente ha la possibilità di vedere le proprie statistiche e filtrare per una data
8. **FR8:** l’utente ha la possibilità di seguire le live organizzate dal personal trainer
9. **FR9:** l’utente ha la possibilità di cambiare il personal trainer e vedere la sua valutazione su una scala da 1 a 5, filtrando anche per luogo.

**Personal Trainer**

1. **FR1:** Il Personal Trainer ha la possibilità di visualizzare gli utenti con l’abbonamento in scadenza.
2. **FR2:** Il Personal Trainer ha la possibilità di creare e modificare le schede di allenamento di un singolo utente
3. **FR3:** Il Personal Trainer ha la possibilità di visualizzare i propri dati e aggiungere eventuali certificazione
4. **FR4:** Il Personal Trainer ha la possibilità di vedere i propri clienti
5. **FR5:** Il Personal Trainer ha la possibilità di vedere le statistiche di un singolo utenti
6. **FR6:** Il Personal Trainer può messaggiare con un singolo cliente
7. **FR7:** Il Personal Trainer può inserire un evento sul calendario disponibile per i propri abbonati

**Nutrizionista**

1. **FR1:** Il Nutrizionista ha la possibilità di visualizzare gli utenti con l’abbonamento in scadenza.
2. **FR2:** Il Nutrizionista ha la possibilità di creare e modificare la dieta
3. **FR3:** Il Nutrizionista ha la possibilità di visualizzare i propri dati e aggiungere eventuali certificazione
4. **FR4:** Il Nutrizionista ha la possibilità di visualizzare i clienti e le statistiche

## Requisiti non funzionali

### Security

Il software essendo integrato in un ambiente cloud eredita le caratteristiche di sicurezza di quest’ultimo. Quali l’integrità, la confidenzialità e l’autenticazione dei dati.

### Scalability

Dato che l’applicazione interagisce in un ambiente cloud, quest’ultima nonostante l’aumento delle richieste da parte degli utenti riesce a scalare e mantenere le performance alte del sistema.

### Usability

Gli utenti fruiscono del servizio mediante un applicazione che si riadatta su due piattaforme : Android e iOS.

### Reliability

Se è presente un malfunzionamento del sistema, l’intera user experience non viene compromessa al più è necessaria rieseguire l’ultima azione

### Performance

Il sistema è in grado di ottimizzare la ricerca e l’esecuzione grazie al uso di un db non relazionale ed ottimizzare il calcolo delle statistiche degli utenti.

### Supportability

Il sistema è totalmente riadattabile al sistema operativo mobile in uso.

## System model

### Scenari

Scenario di inizio allenamento

L’utente apre l’applicazione ed ha già una scheda ed una dieta che può consultare dalla home, ma ha interesse a fare l’allenamento del “Day 1” di conseguenza clicca sulla scheda e si apre il dettaglio del giorno, con l’elenco degli esercizi e in basso inizia allenamento.

L’utente successivamente, cliccando su “Inizia Allenamento” avrà una schermata con una serie di esercizi da fare che saranno cosi composti:

1. Nome Esercizio
2. Ripetizioni
3. Colpi
4. Tempo di recupero

I colpi, rappresentano il numero di volte che quel esercizio deve essere eseguito; alla fine di tutti i colpi l’utente avvierà il timer premendo “start” in modo tale da avviare un cronometro fino al tempo di recupero definito, che successivamente stopperà manualmente con il tasto stop. A ogni colpo è possibile inserire il peso caricato e riprendere.

Questa serie di azioni avverrà per ogni ripetizione alla fine delle ripetizioni comparirà il tasto “Passa all’esercizio successivo”

Scenario Ricerca Personal Trainer o Nutrizionista

L’utente ha intenzione di cambiare Personal Trainer, quindi dopo aver effettuato il login si ritrova sulla homepage. Aprendo il menu laterale, la sidecar, andrà a cliccare su Cambia PT che mostrerà una finestra dove al centro sarà presente una barra di ricerca e un filtro che permette di scegliere la città.

La ricerca può avvenire in due modi:

1. Inserendo il nome utente del Personal Trainer
2. Ricercando tutti i Personal Trainer di un luogo

Entrambe le ricerche restituiranno il nome di un personal Trainer, cliccando si apre un popup che chiede se si è sicuro di cambiare realmente il personal trainer, in caso di conferma verrà effettuato il cambio.

Scenario Inserimento Scheda Allenamento

Il personal Trainer accede all’applicazione mediante le sue credenziali. L’applicazione appena aperta mostra al PT la sua homepage con tutti i piani in scadenza. A questo punto potrà selezionare tramite un click l’utente al quale vuole aggiornare il piano di allenamento. L’applicazione a questo punto lo reindirizzerà all’ultima scheda da lui creata per quell’utente, con la possibilità di modificarla o di crearne una nuova. Il PT decide allora di creare una nuova scheda.

L’applicazione nella pagina di creazione permetterà al PT di inserire mediante delle comboBox, il tipo di esercizio, il numero di ripetizioni, il numero di colpi ed infine il recupero da eseguire a fine esercizio. Una volta completato il giorno di allenamento, cliccherà sul bottone Salva e verrà reindirizzato alla pagina delle schede di allenamento per l’utente selezionato in precedenza.

Scenario Inserimento dieta

Il nutrizionista accede all’applicazione mediante le sue credenziali. L’applicazione appena aperta mostra al nutrizionista la sua homepage con tutti i piani in scadenza. A questo punto potrà selezionare tramite un click l’utente al quale vuole aggiornare il piano di alimentazione. L’applicazione a questo punto lo reindirizzerà alla schermata contenente l’ultima dieta da lui creata per quell’utente, con la possibilità di modificarla o di crearne una nuova. Il nutrizionista decide allora di creare un nuovo piano.

L’applicazione nella pagina di creazione permetterà al nutrizionista di inserire in base alla parte della giornata, gli alimenti che ritiene opportuno. Una volta completato il piano, cliccherà sul bottone Salva e la dieta di quell’utente verrà aggiornata in base a quanto modificato.

### Use case model

|  |  |
| --- | --- |
| **Nome Caso d’uso** | Seleziona tipo di ricerca |
| **Attori** | Utente |
| **Flusso di eventi** | * L’utente apre l’applicazione * All’utente si presenta una pagina contenente dei pulsanti che permettono di selezionare il tipo di ricerca che vuole effettuare * L’utente seleziona il tipo di ricerca |
| **Entry Condition** | * L’utente avvia l’applicazione |
| **Exit Conditions** | * L’utente ha selezionato il tipo di ricerca |

|  |  |
| --- | --- |
| **Nome Caso d’uso** | Stesura recensione |
| **Attori** | Utente |
| **Flusso di eventi** | 1. Il sistema mostra all’utente oltre alle informazioni del locale la possibilità di scrivere una recensione; 2. L’utente tramite la schermata di recensione compila la form con titolo e testo della recensione;    * Inserimento valutazione locale (1 pessimo, 5 ottimo);    * Valutazione qualità del servizio;    * Valutazione qualità del cibo/bevanda; 3. Il sistema valida la recensione e si accerta che non ci sono parole offensive\*; |
| **Entry Condition** | * L’utente ha selezionato un locale |
| **Exit Conditions** | * Il sistema accetta la recensione |
| **Exception** | * Nel punto 3 del flusso di eventi il sistema controlla se ogni campo della form è compilato correttamente, nel caso in cui questo non accade mostra all’utente i campi sbagliati marcandoli in rosso. |

|  |  |
| --- | --- |
| **Nome Caso d’uso** | Ricerca per nome |
| **Attori** | Utente |
| **Flusso di eventi** | 1. L’utente scrive il nome del locale nell’apposita area di testo 2. L’applicazione verifica se è stato rispettato il formato: *[Nome locale, Città]* 3. Il sistema verifica l’esistenza della città sul database. 4. Recupera la lista dei locali associati a quel nome 5. Restituisce il locale (potrebbero essere presenti più locali con lo stesso nome e si trovano nella stessa città |
| **Entry Condition** | * L’utente ha selezionato il tipo di ricerca per nome |
| **Exit Conditions** | * Il sistema ritorna il risultato della ricerca |
| **Eccezione** | * Nel punto 3 in caso di riscontro negativo, esegue caso d’uso “Apprendimento” |

|  |  |
| --- | --- |
| **Nome Caso d’uso** | Ricerca per posizione |
| **Attori** | Utente, GPS |
| **Flusso di eventi** | 1. L’utente chiede la posizione al Gps che gli ritorna la latitudine e longitudine, che verranno scritte nell’apposita area di testo. 1.1 L’utente seleziona una o più categorie: Enoteche, Discoteche, Bar o Pub 2. Il sistema verifica se il luogo è già presente sul database 3. Restituisce la lista dei locali in quella zona |
| **Entry Condition** | * L’utente ha selezionato il tipo di ricerca per posizione |
| **Exit Conditions** | * Il sistema ritorna il risultato della ricerca |
| **Exception** | * Nel punto 1 del flusso di eventi se il GPS è disattivato, il sistema chiede all’utente di attivarlo * Nel punto 2 in caso di riscontro negativo, esegue caso d’uso “Apprendimento” |

|  |  |
| --- | --- |
| **Nome Caso d’uso** | Apprendimento |
| **Attori** | Google API  Foursquare API  Yelp API |
| **Flusso di eventi** | 1. In caso di riscontro negativo, interroga Yelp, Google e Foursquare 2. Esegue il merge dei tre risultati ed esclude i duplicati 3. Salva le informazioni sul database |
| **Entry Condition** | * Il sistema verifica se il luogo è presente nel database |
| **Exit Conditions** | * Aggiornamento del database e restituzione dati |

|  |  |
| --- | --- |
| **Nome Caso d’uso** | Modifica della recensione dalla pagina del locale |
| **Attori** | Utente |
| **Flusso di eventi** | 1. L’utente accede alla pagina del locale utilizzando il caso d’uso di ricerca 2. Il sistema ordina le recensioni in modo tale da mostrare come prima recensione quella dell’utente 3. L’utente chiede al sistema di modificare la recensione desiderata. 4. L’utente apporta le modifiche desiderate. 5. Il sistema valida la recensione e si accerta che non ci siano parole offensive. |
| **Entry Condition** | * L’utente ha selezionato una recensione |
| **Exit Condition** | * Il sistema aggiorna la recensione |
| **Exception** | * il sistema può rifiutare la recensione o mantenere la precedente |

|  |  |
| --- | --- |
| **Nome Caso d’uso** | Modifica della recensione dalla lista delle recensioni |
| **Attori** | Utente |
| **Flusso di eventi** | 1. L’utente accede alla recensione dall’apposita area della lista di recensioni. 2. L’utente chiede al Sistema di modificare la recensione desiderata. 3. L’utente apporta le modifiche desiderate. 4. Il sistema valida la recensione e si accerta che non ci sono parole offensive |
| **Entry Condition** | * L’utente ha selezionato la recensione dalla lista delle recensioni |
| **Exit Condition** | * Il sistema aggiorna la recensione |
| **Exception** | * Il sistema può rifiutare la recensione o mantenere la precedente |

|  |  |
| --- | --- |
| **Nome Caso d’uso** | Ricerca per luogo |
| **Attori** | Utente |
| **Flusso di eventi** | 1. L’utente inserisce nell’apposita area di testo il luogo interessato. 1.1 L’utente seleziona una o più categorie: Enoteche, Discoteche, Bar o Pub 2. Il sistema verifica se il luogo esiste 3. Restituisce la lista dei locali in quella zona |
| **Entry Condition** | * L’utente ha selezionato il tipo di ricerca per luogo |
| **Exit Conditions** | * Il sistema ritorna il risultato della ricerca |
| **Exception** | * Nel punto 2 in caso di riscontro positivo che il luogo è già stato ricercato, esegue caso d’uso “Verifica luogo (Presente)” * Nel punto 2 in caso di riscontro negativo che il luogo inserito è sbagliato, esegue caso d’uso “Verifica luogo (Sbagliato)” * Nel punto 2 in caso di riscontro positivo che il luogo ancora non è stato ricercato, esegue caso d’uso “Verifica luogo (Non Presente)” |

|  |  |
| --- | --- |
| **Nome Caso d’uso** | Verifica luogo (Presente) |
| **Attori** | Geocoder API |
| **Flusso di eventi** | 1. L’utente in caso di riscontro positivo, interroga Geocoder API 2. Memorizza il luogo |
| **Entry Condition** | * Presa in input di un luogo |
| **Exit Condition** | * Il sistema scrive il luogo sul database |

|  |  |
| --- | --- |
| **Nome Caso d’uso** | Verifica luogo (Sbagliato) |
| **Attori** | Geocoder API |
| **Flusso di eventi** | 1. L’utente in caso di riscontro negativo, interroga Geocoder API 2. Restituisce che il luogo inserito è sbagliato |
| **Entry Condition** | * Presa in input di un luogo |
| **Exit Condition** | * Il sistema restituisce che è scorretto |

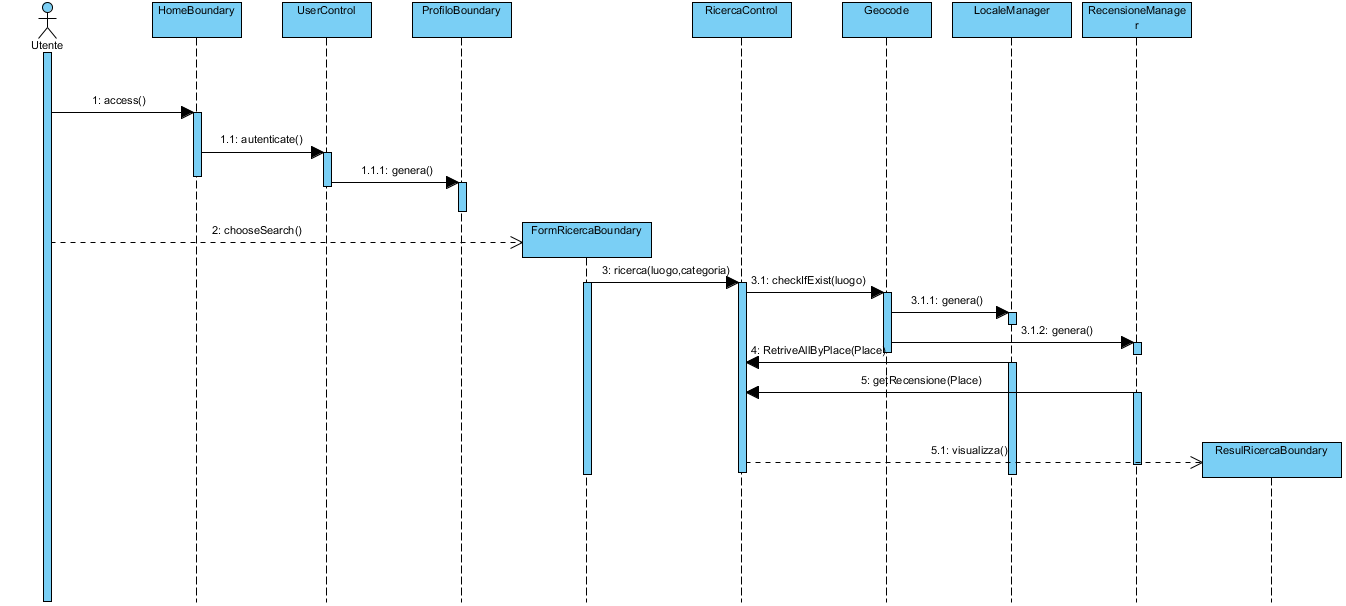
|  |  |
| --- | --- |
| **Nome Caso d’uso** | Verifica luogo (Presente) |
| **Attori** | Geocoder API |
| **Flusso di eventi** | 1. L’utente in caso di riscontro positivo, interroga Geocoder API 2. Memorizza il luogo |
| **Entry Condition** | * Presa in input di un luogo |
| **Exit Condition** | * Il sistema scrive il luogo sul database |

|  |  |
| --- | --- |
| **Nome Caso d’uso** | Verifica luogo (Sbagliato) |
| **Attori** | Geocoder API |
| **Flusso di eventi** | 1. L’utente in caso di riscontro negativo, interroga Geocoder API 2. Restituisce che il luogo inserito è sbagliato |
| **Entry Condition** | * Presa in input di un luogo |
| **Exit Condition** | * Il sistema restituisce che è scorretto |

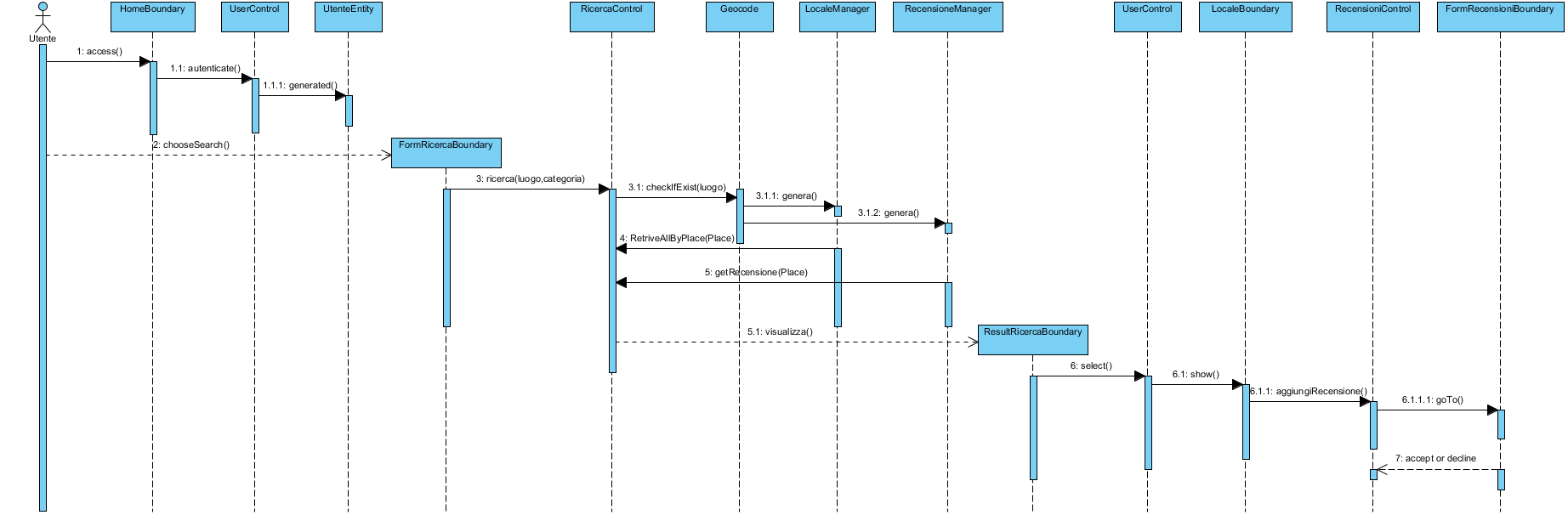
|  |  |
| --- | --- |
| **Nome Caso d’uso** | Verifica luogo (Non Presente) |
| **Attori** | Geocoder API |
| **Flusso di eventi** | 1. L’utente in caso di riscontro negativo, interroga Geocoder API 2. Trasforma il luogo in latitudine e longitudine 3. Salva informazioni sul database |
| **Entry Condition** | * Presa in input di un luogo |
| **Exit Condition** | * Il sistema memorizza i dati |

### Sequence diagram

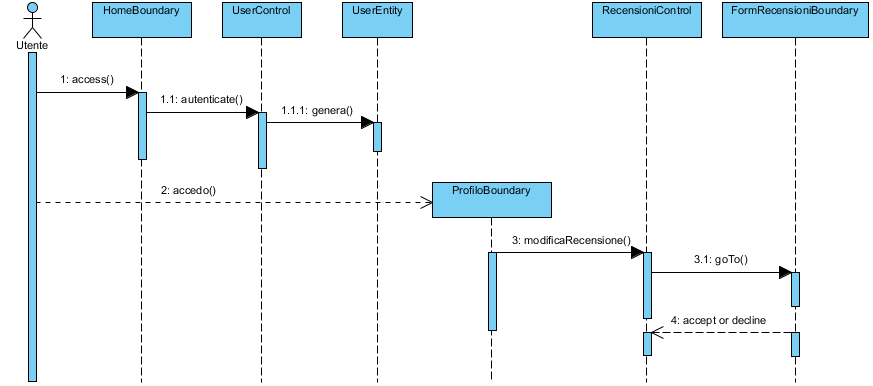
**Sequence diagram: Ricerca**

****

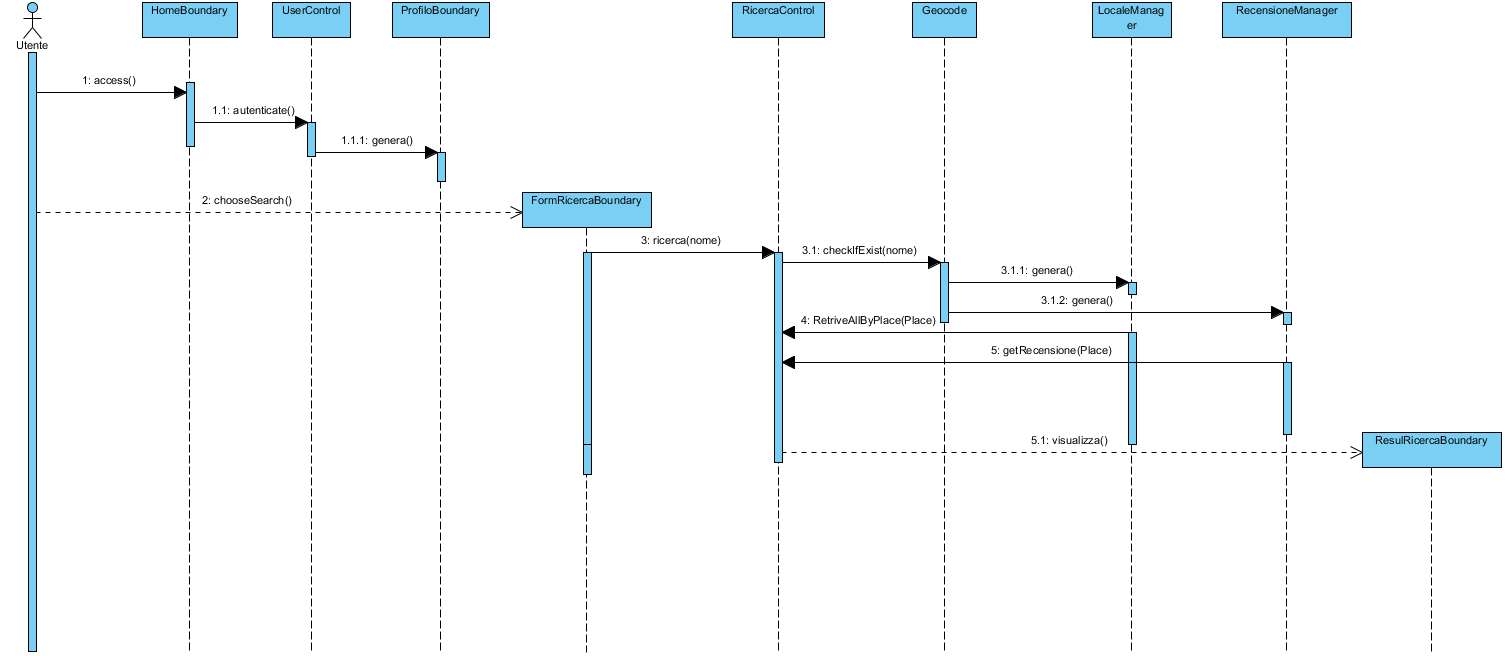
**Sequence diagram: Inserisci recensione**

****

**Sequence diagram: Modifica recensione**

****

**Sequence diagram: Ricerca per nome**

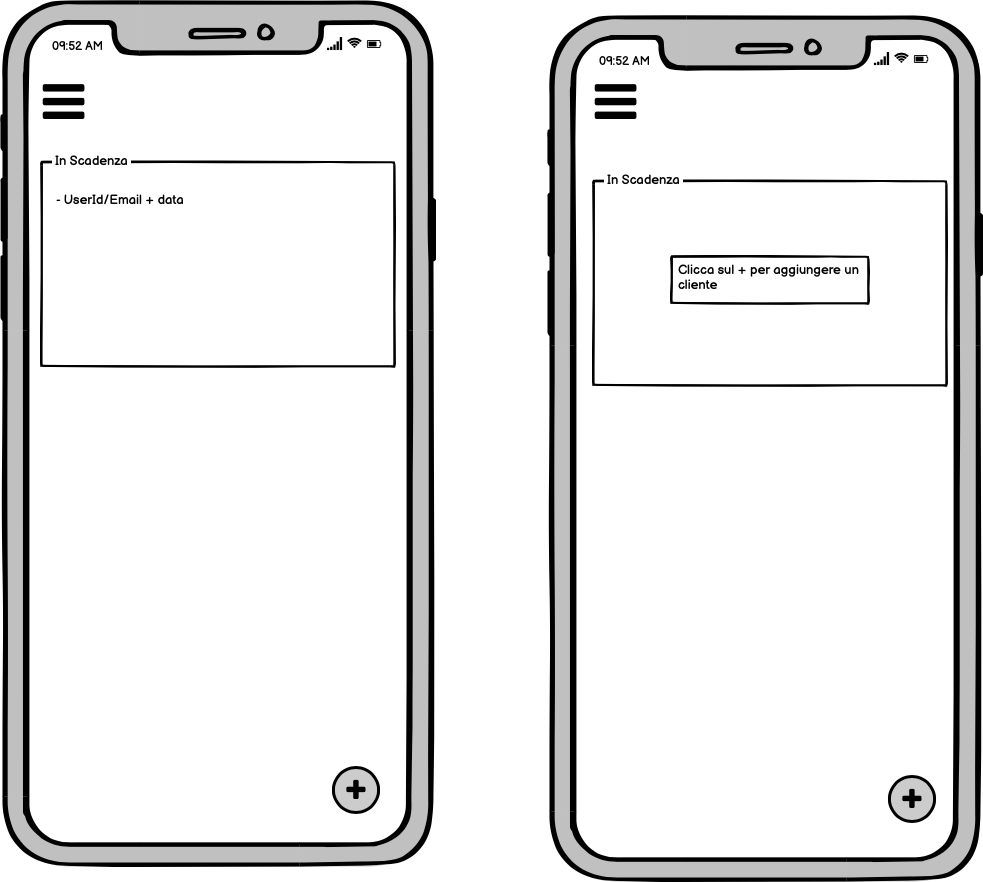
****

### Mock-up

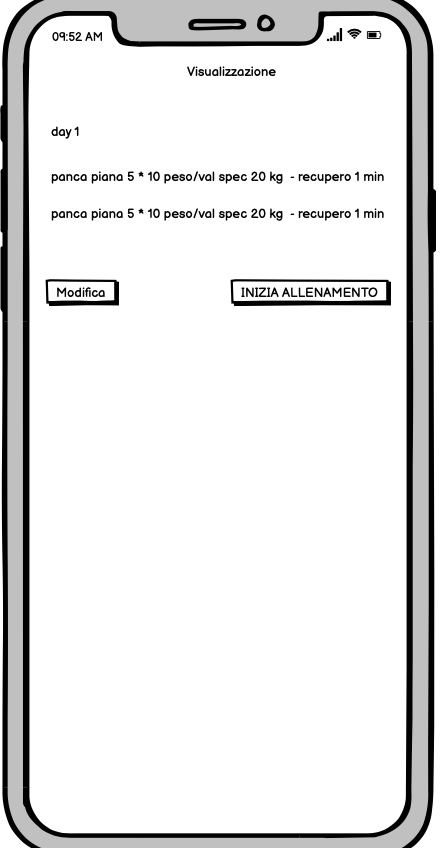
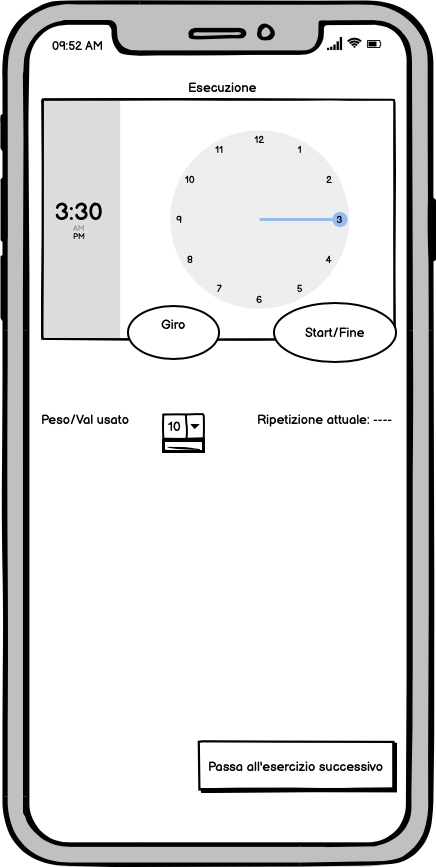
**Visualizzazione HomePage Utente**

****

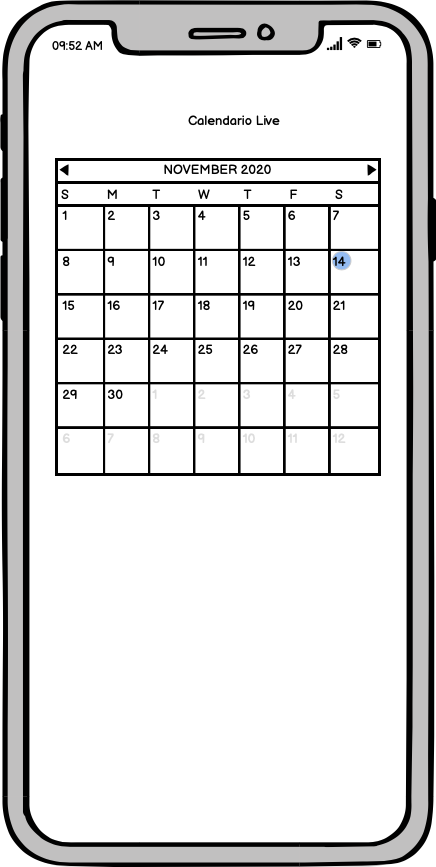
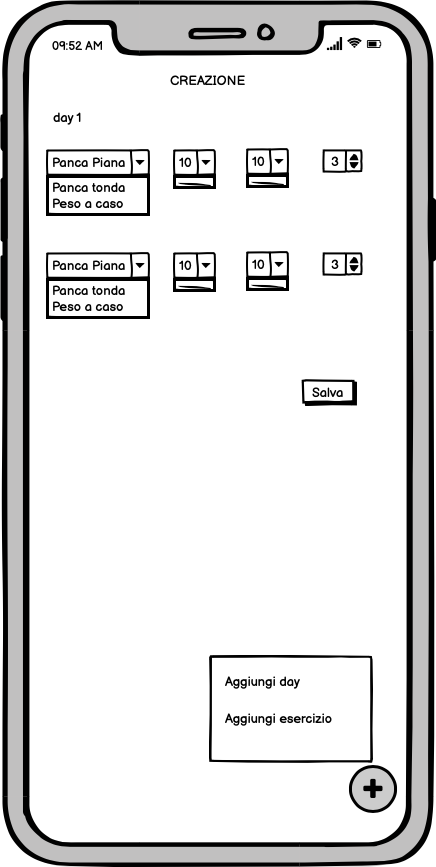
**Visualizzazione HomePage Nutrizionista e Personal Trainer**

****

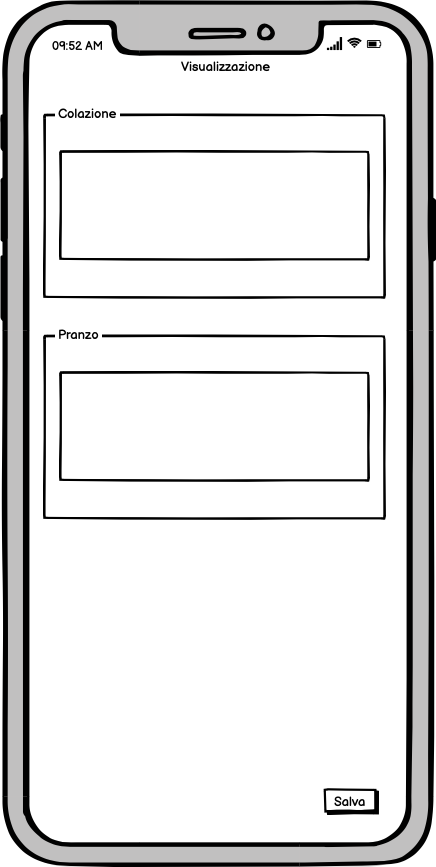
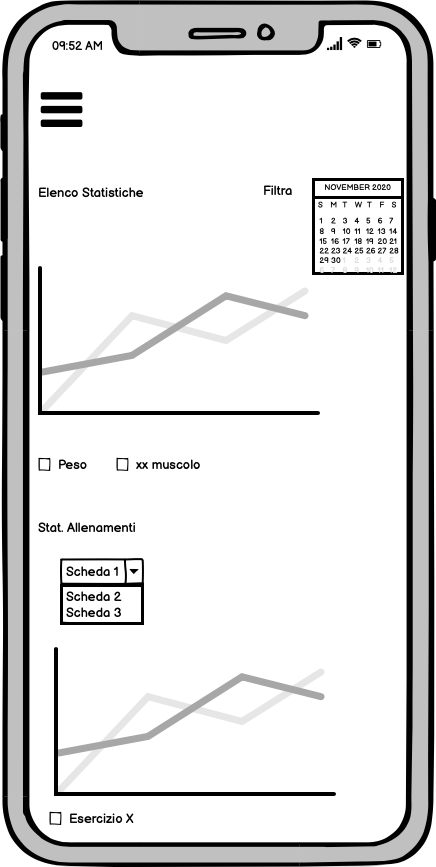
**Visualizzazione Giorno Allenamento Schermata Inizia Allenamento**

** **

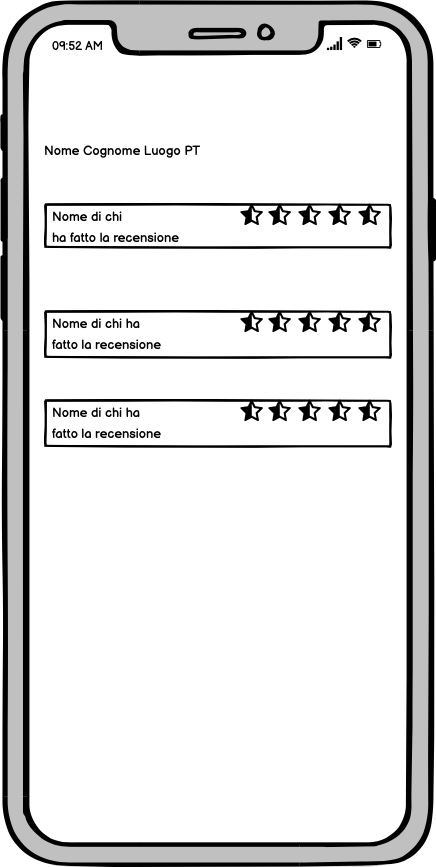
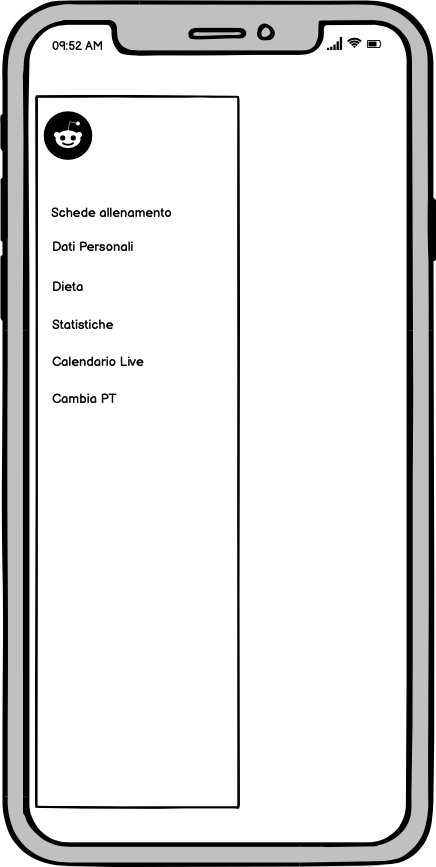
**Visualizzazione Calendario Live Creazione Scheda Allenamento**

** **

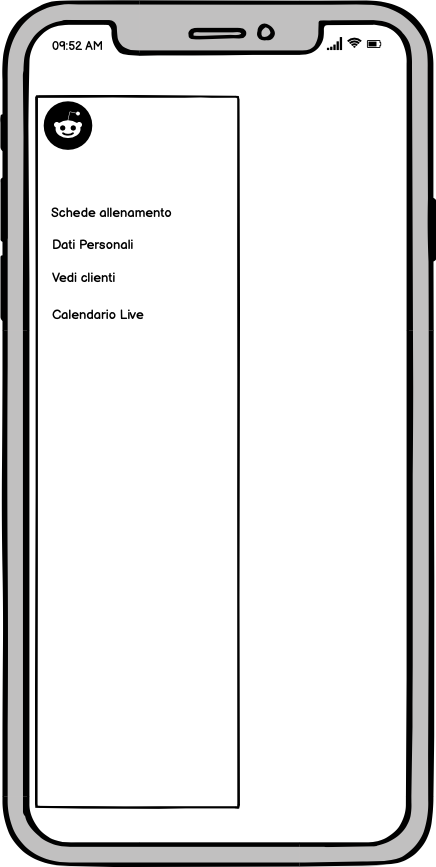
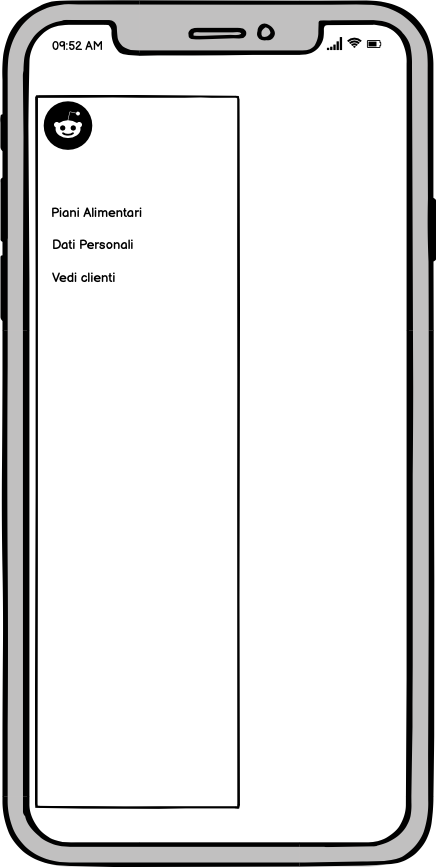
**Creazione Piano Alimentare Statistiche Lato Utente**

** **

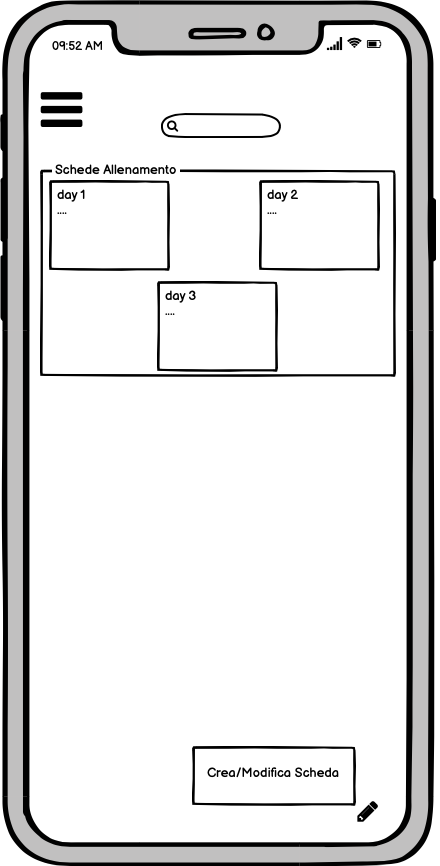
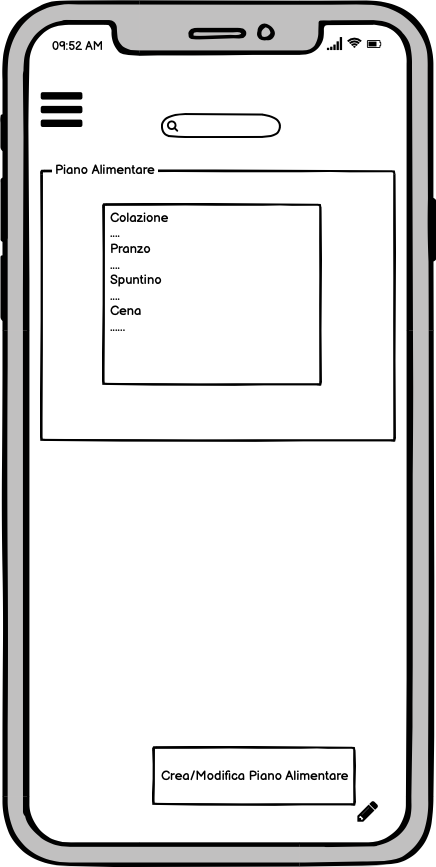
**Recensioni Personal Trainer Menù laterale Utente**

** **

**Menu Laterale Personal Trainer Menù laterale Nutrizionista**

** **

**Info Clienti Personal Trainer Info Clienti Nutrizionista**

** **