

# Documentazione

Link GitHub: <https://github.com/mariadianacalugaru/fitnet>

La nostra applicazione fornisce un servizio di fitness che dà la possibilità di richiedere e fornire schede di allenamento personalizzate secondo gli obiettivi di ogni singolo cliente.

Suddivisione del lavoro (approssimativamente) :

- Maria Diana Calugaru: permessi utente, modifica profilo, gestione richiesta e recensione, test rspec
- Rosapia Laudati: permessi admin, gestione messaggi, eliminazione utenti, schede, richieste e personal trainer, aggiornamento esercizi con api esterne, richieste e messaggi, feature review
- Samuele Cervo: permessi personal trainer, gestione delle richieste, creazione e visualizzazione schede di allenamento, feature request
- Andrea Di Vincenzo: homepage, navbar (About us, Our PTs), sidebar, login e registrazione, parte grafica pagine html, features login e registrazione

## User Stories:

### Unregistered User

- As an UnRegistered User I want to login with Google so that I can become a User.
- As an UnRegistered User I want to login with Google so that I can become a Personal Trainer.
- As an UnRegistered User I want to sign up with my e-mail so that can become a User.
- As an UnRegistered User I want to sign up with my e-mail so that can become Personal Trainer.
- As an Unregistered User I want to have access to the homapage so that I can get info about the services our application provides.
- As an Unregistered User I want to have access to the homapage so that I can read customers' reviews.
- As an Unregistered User I want to have a page so that I can visualize the list of personal trainers.

### User

- As a User I want to have settings so that I can login with e-mail and password.
- As a User I want to have settings so that I can upload profile picture.
- As a User I want to have settings so that I can delete my account.
- As a User I want to have settings so that I can logout.
- As a User I want to have profile so that I can set informations about myself.
- As a User I want to have profile so that I can change informations about myself.
- As a User signed in with Google i want to have settings so i can edit my profile without providing a password.
- As a User I want to have settings so that I can switch to Personal Trainer.
- As a User signed in with Google i want to have settings so my profile picture can be uploaded automatically.
- As a User I want to have a page so that I can visualize a list of the available Personal Trainers.
- As a User I want to make a request so that I can get a workout schedule from a specific Personal Trainer.
- As a User I want to make a request so that I can get a workout schedule from any Personal Trainer.
- As a User I want to have a page so that I can visualize the list of my workout schedules.
- As a User I want to have a page so that I can visualize a single workout schedule.
- As a User I want to have a page so that I can add review.
- As a User I want to have a page so that i can see my dashboard.

### Personal Trainer

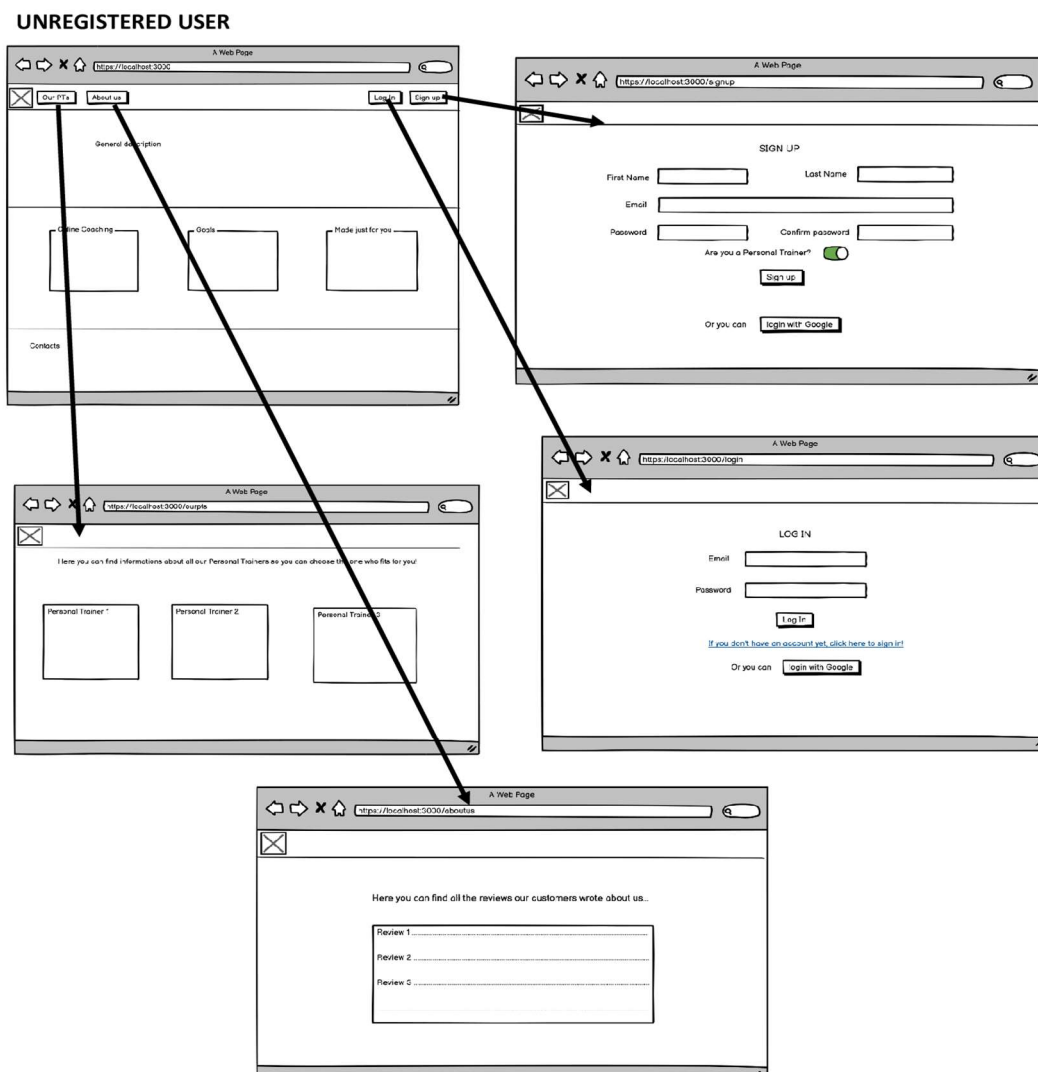
- As a Personal Trainer I want to have settings so that I can login with e-mail and password.

- As a Personal Trainer I want to have settings so that I can upload profile picture.
- As a Personal Trainer I want to have settings so that I can delete my account.
- As a Personal Trainer I want to have settings so that I can logout.
- As a Personal Trainer I want to have profile so that I can upload my certificate.
- As a Personal Trainer I want to have profile so that I can upload my personal information.
- As a Personal Trainer I want to have a page so that I can see requests for workout schedules.
- As a Personal Trainer I want to select a request so that I can create a workout schedule.
- As a Personal Trainer I want to select exercises so that I can add them to the workout schedule.
- As a User I want to have a page so that I can see my dashboard.

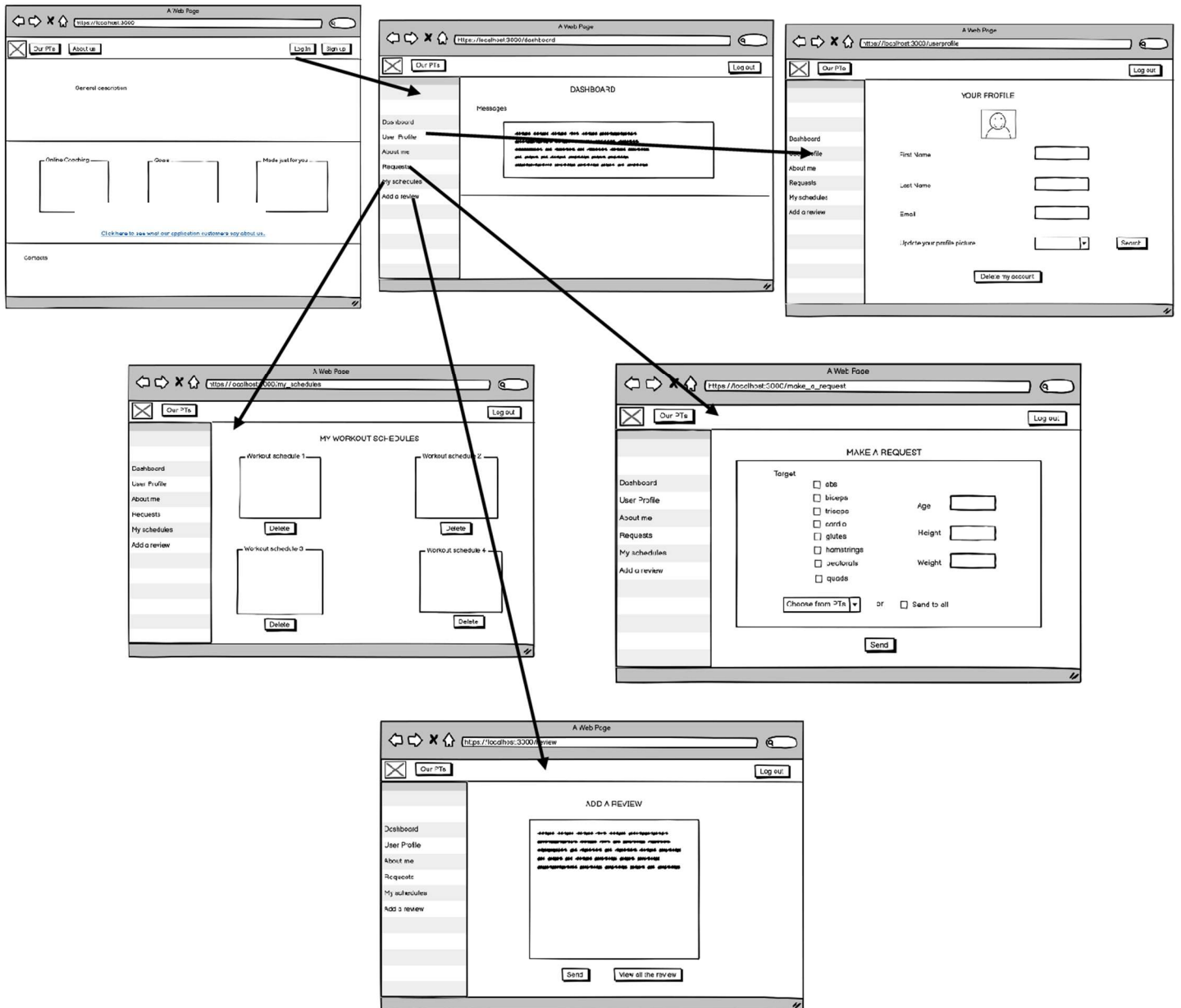
## Admin

- As an Admin I want to have settings so that I can login with e-mail and password.
- As an Admin I want to have special settings so that I can delete Users.
- As an Admin I want to have special settings so that I can delete Personal Trainers.
- As an Admin I want to have special settings so that I can see all the workout schedules.
- As an Admin I want to have special settings so that I can delete workout schedules.
- As an Admin I want to have special privileges so that I can see all profiles of all Users.
- As an Admin I want to have special privileges so that I can see all profiles of all Personal Trainers.
- As an Admin I want to have special settings so that I can send messages to everyone.

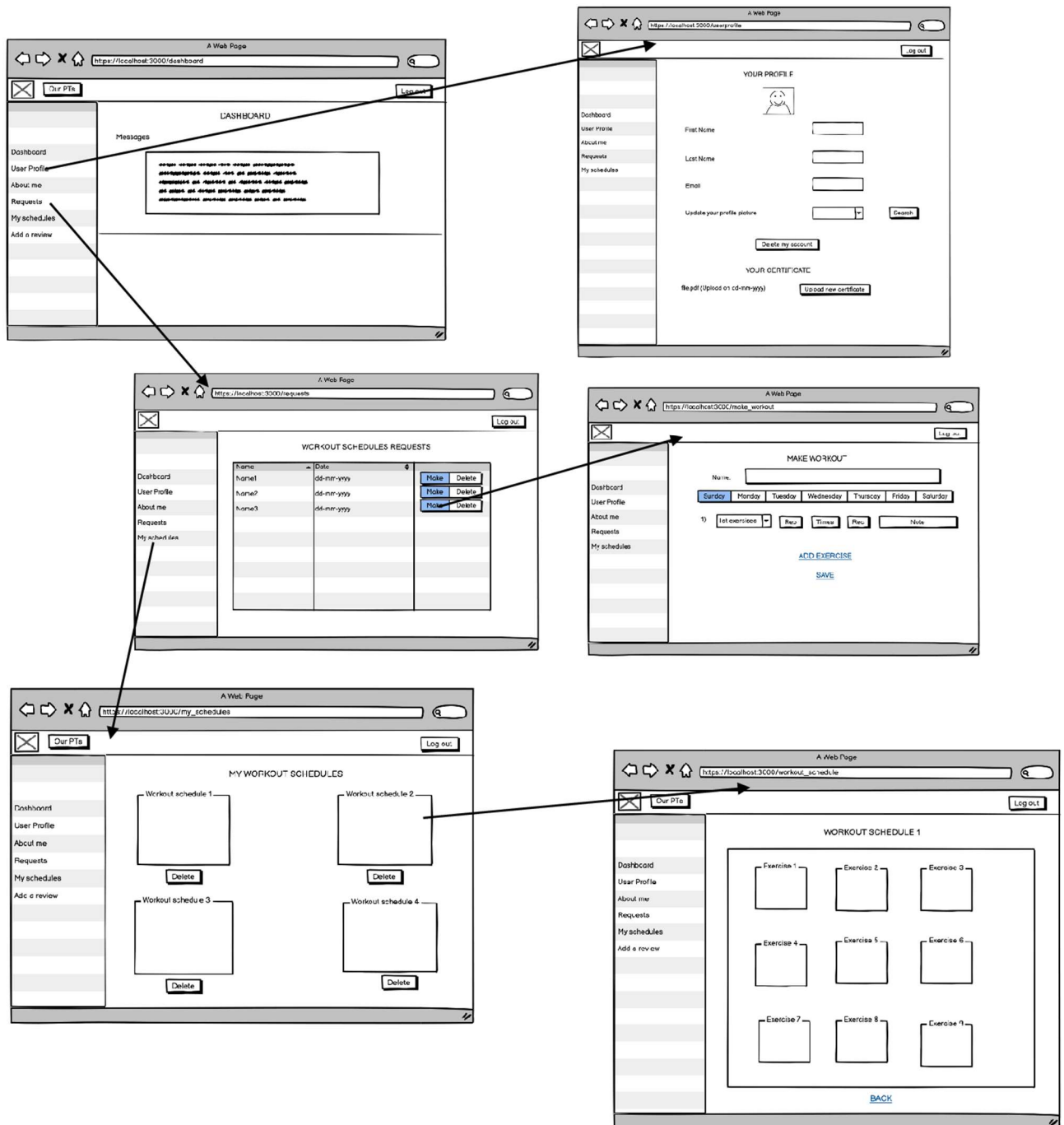
## Mockups:



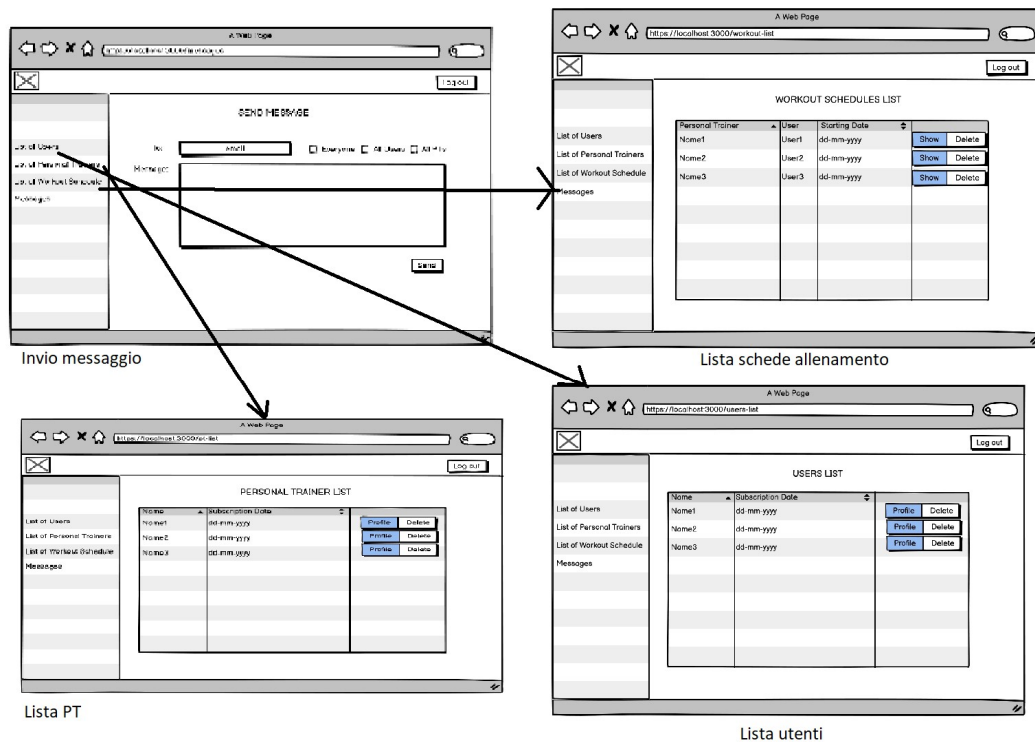
# USER



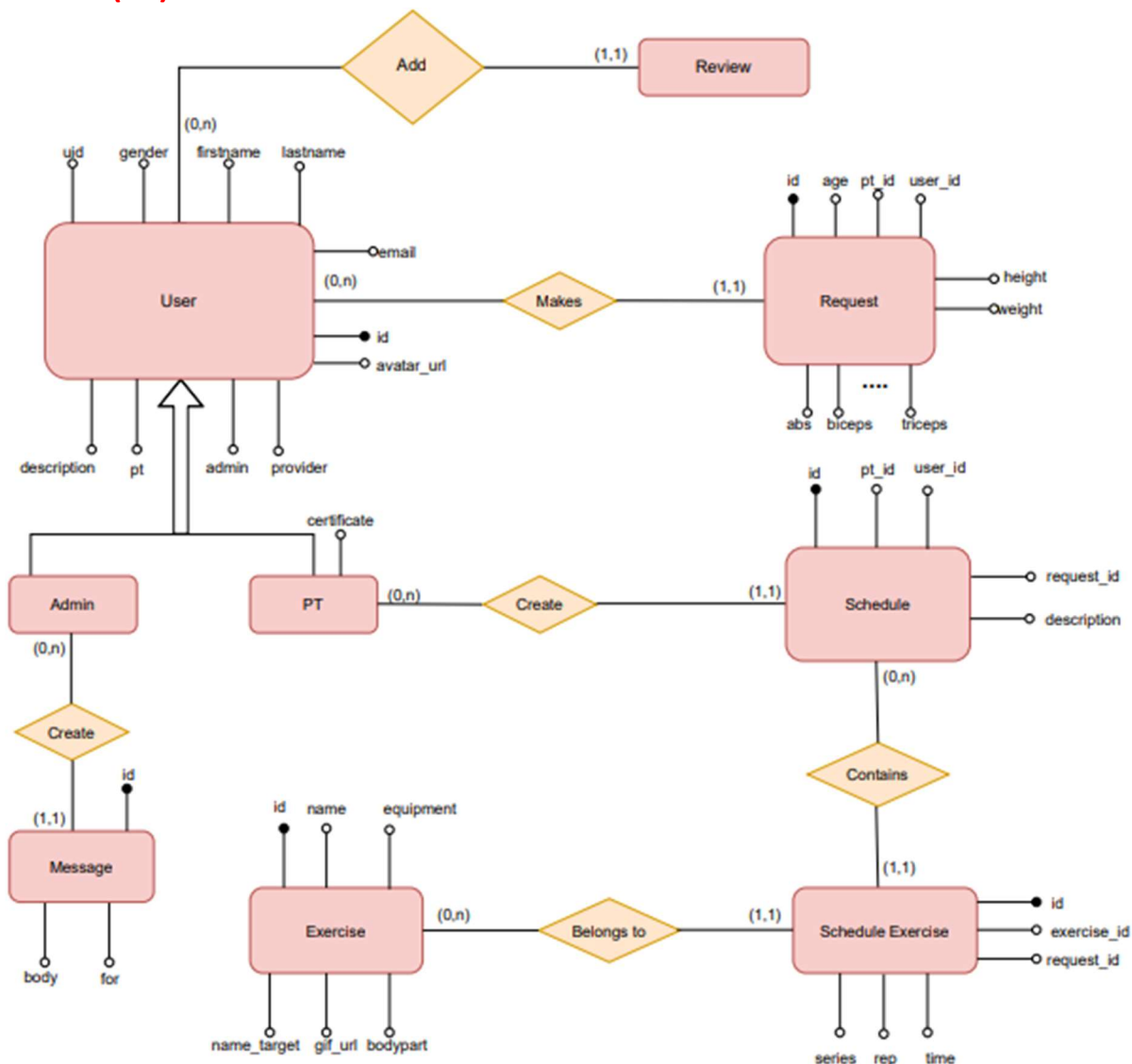
# Personal Trainer



## ADMIN



## Schema DB (ER):



## Struttura controllo degli accessi alle funzionalità:

La registrazione e il login degli utenti sono stati implementati tramite l'uso della gemma "Devise", mentre per quanto riguarda l'autenticazione esterna abbiamo utilizzato la gemma "omniauth-google-oauth2".

Due modalità di accesso:

- locale: l'utente si registra all'interno dell'applicazione
- OAuth: l'utente accede tramite Google - in questo modo è possibile compilare il form di registrazione e di accesso usando i dati del proprio account Google.

### **GUEST**

Un utente GUEST che non ha effettuato l'accesso può solamente accedere all'homepage, leggere le recensioni e vedere i personal trainer registrati nell'applicazione.

### **USER**

Un utente normale può:

- accedere all'area riservata
- leggere, modificare e cancellare il proprio account
- leggere e aggiungere una recensione
- creare una richiesta per una scheda di allenamento, leggere ed eliminare le proprie richieste
- leggere i messaggi ad esso indirizzati
- leggere ed eliminare le proprie schede
- leggere gli esercizi delle sue schede

### **PT**

Un utente registrato come personal trainer può:

- accedere all'area riservata
- leggere, modificare e cancellare il proprio account
- leggere gli altri personal trainer
- leggere le recensioni
- leggere le richieste di allenamento rivolte a lui o a tutti i personal trainer
- eliminare le richieste di allenamento rivolte a lui
- creare schede di allenamento, leggere ed eliminare le schede di allenamento da lui create
- leggere i messaggi ad esso indirizzati
- leggere gli esercizi

### **ADMIN**

Un amministratore può:

- accedere all'area riservata
- leggere, modificare e cancellare il proprio account
- leggere e cancellare qualsiasi account (sia di un utente normale che di un pt)

- leggere le recensioni
- leggere ed eliminare qualsiasi richiesta
- leggere ed eliminare qualsiasi scheda di allenamento
- leggere e creare esercizi tramite chiamate API
- leggere, creare ed eliminare qualsiasi messaggio

## Piano dei test:

Per effettuare i test abbiamo usato tre tools:

- Cucumber, con cui abbiamo testato alcune funzionalità presenti nelle User Stories sfruttando i metodi della gemma Capybara. Nella cartella features abbiamo eseguito i test di accettazione per il login, la registrazione, l'aggiunta di una recensione e l'invio di una richiesta.
- Database-cleaner, che pulisce ogni record di dati che è stato generato nei test.
- RSpec, con cui abbiamo effettuato i test di unità su alcuni models presenti all'interno dell'applicazione, come per esempio User, Review, Request e su alcuni controllers per testare la correttezza delle operazioni che si possono eseguire sui modelli Review e Request in base ai permessi.

Di seguito riportiamo un esempio completo di test sulla creazione e invio di una richiesta

- Feature relativa alla creazione di una nuova richiesta, con due possibili scenari (invio corretto della richiesta/ errore nell'invio della richiesta)

```
Scenario: Create a new request: success
  Given I am on the home page
  Given I am a registered user

  When I follow "Log In"
  Then I should be on the login page
  When I fill in "Email" with "lorenzo@gmail.com"
  And I fill in "user_password" with "lorenzo2000"
  And I press "Login"
  Then I should be on the home page

  When I follow "Personal Area"
  Then I should be on the dashboard page

  When I follow "My Requests"
  Then I should be on the requests page

  When I follow "Add a request"
  Then I should be on the create a new request page

  When I fill in "age" with "22"
  And I fill in "weight" with "83"
  And I fill in "height" with "185"

  And I check "glutes"
  Then the "glutes" checkbox should be checked
  And I check "cardio"
  Then the "cardio" checkbox should be checked
  And I check "pectorals"
  Then the "pectorals" checkbox should be checked

  And I press "Add request"
  Then I should be on the requests page
  And I should see "Request successfully sent."
```

```
Scenario: Create a new request:failure
  Given I am on the home page
  Given I am a registered user

  When I follow "Log In"
  Then I should be on the login page
  When I fill in "Email" with "lorenzo@gmail.com"
  And I fill in "user_password" with "lorenzo2000"
  And I press "Login"
  Then I should be on the home page

  When I follow "Personal Area"
  Then I should be on the dashboard page

  When I follow "My Requests"
  Then I should be on the requests page

  When I follow "Add a request"
  Then I should be on the create a new request page

  When I fill in "age" with "22"
  And I fill in "height" with "185"

  And I check "glutes"
  Then the "glutes" checkbox should be checked
  And I check "cardio"
  Then the "cardio" checkbox should be checked
  And I check "pectorals"
  Then the "pectorals" checkbox should be checked

  And I press "Add request"
  Then I should be on the requests page
  And I should see "Weight can't be blank"
```



- Test RSpec relativo al model Request

```
spec > models > request_spec.rb
3 RSpec.describe Review, type: :model do
4
5   before do
6     @user = FactoryBot.create(:user)
7     @pt = FactoryBot.create(:pt)
8   end
9
10
11   describe "Create a Request without a user " do
12     it "shouldn't be permitted" do
13       request = Request.create(age: '26', height: '185', weight: '85', abs: true, cardio: true, quads: true, pt_id: @pt)
14       expect(request).to_not be_valid
15     end
16   end
17
18   describe "Create a Request without age " do
19     it "shouldn't be permitted" do
20       request = Request.create(user_id: @user, height: '185', weight: '85', abs: true, cardio: true, quads: true, pt_id: @pt)
21       expect(request).to_not be_valid
22     end
23   end
24
25   describe "Create a Request without height " do
26     it "shouldn't be permitted" do
27       request = Request.create(user_id: @user, age: '26', weight: '85', abs: true, cardio: true, quads: true, pt_id: @pt)
28       expect(request).to_not be_valid
29     end
30   end
31
32   describe "Create a Request without weight " do
33     it "shouldn't be permitted" do
34       request = Request.create(user_id: @user, age: '26', height: '185', abs: true, cardio: true, quads: true, pt_id: @pt)
35       expect(request).to_not be_valid
36     end
37   end
38
39   describe "Create a valid Request " do
40     it "should be permitted" do
41       request = Request.create(user_id: @user, age: '26', height: '185', weight: '85', abs: true, cardio: true, quads: true, pt_id: @pt)
42       expect(request).to_not be_valid
43     end
44
45     it "should be permitted also not to choose a pt" do
46       request = Request.create(user_id: @user, age: '26', height: '185', weight: '85', abs: true, cardio: true, quads: true)
47       expect(request).to_not be_valid
48     end
49   end
50 end
```

- Test RSpec relativo al Request Controller

```
RSpec.describe RequestsController, type: :controller do
# visualizzazione schede
describe "GET #index" do
#utente loggato che vede le sue richieste
context "when user logged in" do
login_user
it 'renders "my requests" page' do
get :index
expect(response).to render_template("index")
end
end

#admin loggato che vede le richieste
context "when user logged in" do
login_admin
it 'renders "my requests" page' do
get :index
expect(response).to render_template("index")
end
end

#utente non loggato
context "when user not logged in" do
it 'renders a redirect to index' do
get :index
expect(response).to redirect_to("/")
end
end

# accesso alla pagina request/new
describe "GET #new" do
#utente loggato ha l'accesso
context "when user logged in" do
login_user
it 'renders "new review" page' do
get :new
expect(response).to render_template(:new)
end
end
end
end
```

```
#utente non loggato non ha l'accesso
context "with no user logged in" do
it "returns a redirect to index" do
allow(controller).to receive(:authenticate_user!).and_return(false)
get :new
expect(response).to redirect_to("/")
end
end

#creazione nuova richiesta
describe "POST #create" do
#utente crea una nuova richiesta per una scheda di allenamento
context "as an user logged in" do
it "create a new request successfully" do
user = FactoryBot.create(:user)
sign_in(user)
pt = FactoryBot.create(:pt)
attributes = { :user_id => user.id,
               :age => '24',
               :height => '180',
               :weight => '76',
               :abs => true,
               :biceps => false,
               :cardio => true,
               :glutes => true,
               :hamstrings => false,
               :pectorals => false,
               :quads => true,
               :triceps => false,
               :pt_id => pt.id }
post :create, params: { request: attributes }

request = Request.find_by(user_id: user.id)
expect(assigns(:request)).to eq(request)
expect(response).to redirect_to(requests_path())
end
end
end
```



```

# un admin non ha il permesso di creare una richiesta per una scheda di allenamento
context "as admin logged in" do
  it "doesn't have permission to create a new request" do
    admin = FactoryBot.create(:admin)
    sign_in(admin)
    pt = FactoryBot.create(:pt)
    attributes = { :user_id => admin.id,
                  :age => '24',
                  :height => '180',
                  :weight => '76',
                  :abs => true,
                  :biceps => false,
                  :cardio => true,
                  :glutes => true,
                  :hamstrings => false,
                  :pectorals => false,
                  :quads => true,
                  :triceps => false,
                  :pt_id => pt.id }

    post :create, params: { request: attributes }
    expect(response).not_to be_successful
  end
end

# un utente non loggato non può creare una richiesta
context "with no user logged in" do
  it "returns a unsuccess response" do
    user = FactoryBot.create(:user)
    pt = FactoryBot.create(:pt)
    attributes = { :user_id => user.id,
                  :age => '24',
                  :height => '180',
                  :weight => '76',
                  :abs => true,
                  :biceps => false,
                  :cardio => true,
                  :glutes => true,
                  :hamstrings => false,
                  :pectorals => false,
                  :quads => true,
                  :triceps => false,
                  :pt_id => pt.id }

    post :create, params: { request: attributes }
    expect(controller.user_signed_in?).to be false
    expect(response).not_to be_successful
  end
end

```

```

#eliminazione richiesta
describe "DELETE #destroy" do
  # utente non loggato
  context "with no user logged in" do
    it "returns no successfull response" do
      user = FactoryBot.create(:user)
      request = FactoryBot.create(:request)

      allow(controller).to receive(:authenticate_user!).and_return(false)
      delete :destroy, params: { :id => request.id }
      expect{ delete :destroy, params: { :id => request.id } }.not_to change(Request, :count)
      expect(response).not_to be_successful
    end
  end

  # utente loggato
  context "with user logged in" do
    it "returns successfull delete" do
      user = FactoryBot.create(:user)
      sign_in(user)
      request = FactoryBot.create(:request, :user_id => user.id)

      expect{ delete :destroy, params: { :id => request.id } }.to change(Request, :count).by(-1)
      expect(response).to redirect_to(requests_path)
    end
  end

  # utente loggato che cerca di eliminare una richiesta non sua
  context "with user logged in" do
    login_user
    it "returns successfull delete" do
      request = FactoryBot.create(:request)

      expect{ delete :destroy, params: { :id => request.id } }.not_to change(Request, :count)
      expect(response).not_to be_successful
    end
  end
end

```