

# Diario di lavoro

Luogo	Scuola Arti e Mestieri Trevano
Data	mercoledì, 30 gennaio 2019

## Lavori svolti

Oggi ho ripreso la pagina per la gestione dei clienti, l'ultima volta la pagina oltre a modificare i vari dati dei clienti poteva anche soft eliminarli. Ho voluto fare in modo che solamente all'interno di quella pagina il responsabile possa vedere anche i clienti soft eliminati ed avere la possibilità di riabilitarli.

GESTIONE PALESTRA			Home Responsabile	Logout
Aaliyah	Howe	ebony.o'conner@example.com	Modifica dati	Reimposta password
Aaron	Greenfelder	verona.o'hara@example.org	Modifica dati	Reimposta password
Abbey	Miller	melba08@example.net	Modifica dati	Reimposta password
Abby	Zemlak	uhomenick@example.net	Modifica dati	Reimposta password
Abdullah	Quigley	kenyatta.zieme@example.net	Modifica dati	Reimposta password
Abelardo	Harris	santa.bailey@example.com	Modifica dati	Reimposta password
Abigayle	Kassulke	antwon03@example.net	Modifica dati	Reimposta password
Abigayle	Hilpert	sierra.luetngen@example.com	Modifica dati	Reimposta password
Adah	Wehner	odell.bauch@example.com	Modifica dati	Reimposta password
Adalberto	Rodriguez	carley.huels@example.org	Modifica dati	Reimposta password

Dopo aver fatto ciò come detto ieri del diario ho creato ma pagina per modificare i dati del cliente, graficamente si presenta così:

GESTIONE PALESTRA		Home Responsabile	Logout
<div> <div>Email cliente</div> <div>davide.paradiso@samtrevano.ch</div> <div>Massa cliente (Kg)</div> <div>80</div> <div>Altezza cliente (m)</div> <div>1.87</div> <div>Modifica</div> </div>			
DAVIDE PARADISO SAMT I4AA 2018/2019			

La pagina fa visualizzati i dati correnti del cliente e dà la possibilità di modificarli.

Per far rappresentare i dati (quindi la grafica della pagina in generale) ho fatto una funzione nel controller:

```
function modificaDati($id) {
    // Riporto l'id del cliente nella pagina
    // Richiamo la classe del responsabile
    require_once 'application/models/responsabile.php';
    // Creo l'oggetto responsabile
    $r = new Responsabile();
    // Mi faccio restituire la mail del cliente dal responsabile
    $mailCliente = $r->getClientMail($id);
    // Richiamo la classe cliente
    require_once 'application/models/cliente.php';
    // Creo l'oggetto cliente
    $c = new Cliente($mailCliente);
    // Prendo i dati del cliente
    $pesoCliente = $c->getMassa();
    $altezzaCliente = $c->getAltezza();
    // View
    require 'application/views/_template/header.php';
    require 'application/views/responsabile/modificaDati.php';
    require 'application/views/_template/footer.php';
}
```

Per fare la modifica dei dati del cliente invece ho creato un'altra funzione sempre nel controller:

```
function changeDataCliente($id) {
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        // Controllo che i campi non siano vuoti
        $mail = isset($_POST['mail']) ? $_POST['mail'] : null;
        $peso = isset($_POST['peso']) ? $_POST['peso'] : null;
        $altezza = isset($_POST['altezza']) ? $_POST['altezza'] : null;
        // Se anche un campo è vuoto ritorno alla home del responsabile
        if ($mail == null || $peso == null || $altezza == null)
            header("location: " . URL . "clienteController/responsabile");

        // Richiamo la classe cliente
        require_once 'application/models/cliente.php';
        // Calcolo il BMI
        $bmi = Cliente::calcBMI($altezza, $peso);
        // Richiamo la classe del responsabile
        require_once 'application/models/responsabile.php';
        // Creo l'oggetto responsabile
        $r = new Responsabile();
        // Aggiorno i nuovi dati del cliente
        $r->updateDataCliente($id, $altezza, $peso, $bmi, $mail);
        // Ritorno alla pagina di modifica
        header("location: " . URL . "responsabileController/listaModifiche");
    }
}
```

La funzione che effettivamente modifica i dati del cliente è **updateDataCliente()** che viene richiamata tramite un oggetto di tipo **Responsabile**:

```
public function updateDataCliente($id, $altezza, $peso, $BMI, $mail) {
    $conn = $this->db->getConnection();
    // Faccio le query
    $query = "UPDATE cliente SET Altezza = $altezza WHERE id = $id";
    $conn->query($query);
    $query = "UPDATE cliente SET Massa = $peso WHERE id = $id";
    $conn->query($query);
    $query = "UPDATE cliente SET BMI = $BMI WHERE id = $id";
    $conn->query($query);
    $query = "UPDATE cliente SET Mail = '$mail' WHERE id = $id";
    $conn->query($query);
    $conn->close();
}
```

Dopo aver fatto questa parte di sito ho cominciato a riempire il database collegando i vari attrezzi ai parametri.

 id	 Parametro	 id_attrezzo
18	Ripetizioni	6
19	Serie	6
20	Carico	7
21	Ripetizioni	7
22	Serie	7
23	Carico	8
24	Ripetizioni	8
25	Serie	8
26	Carico	9
27	Ripetizioni	9
28	Serie	9

#### Problemi riscontrati e soluzioni adottate

-

#### Punto della situazione rispetto alla pianificazione

In orario con la mia pianificazione.

Programma di massima per la prossima giornata di lavoro

Iniziare l'implementazione della pagina per creare gli esercizi.