

# Diario di lavoro

Luogo	SAMT
Data	13.11.2018

## Lavori svolti

Questa mattina per prima cosa ho ricontrollato il lavoro fatto la settimana scorsa con la pagina dell'admin e dei coach e fatto ancora alcuni test, dopodiché mi sono assicurato che il fattore funzioni dei controller funzionava correttamente, ovvero se provavo ad accedere ma non potevo non me lo faceva fare ma mi riportava alla pagina di home, ed effettivamente la situazione non era cambiata, funzionava solo se dall'amministratore provavo a collegarmi a funzioni di home o dei coach, ma quelli degli allievi davano lo stesso problema della settimana scorsa, poi mi è venuto in mente che effettivamente il controllo se era arrivata una richiesta POST lo facevo solo per prendere la variabile, in questo modo.

```

63 //Controllo che esista la sessione
64 if ($_SERVER["REQUEST_METHOD"] == "POST") {
65     //Prendo i dati in entrata dal form
66     $subject = $_POST["subject"];
67     $day = $_POST["day"];
68     $hour = $_POST["hour"];
69     $learners = $_POST["learners"];
70 }

```

Ma mi sono reso conto che il metodo migliore era di controllare su tutto il codice della funzione perché altrimenti tutto il resto veniva eseguito ugualmente e non era quello che volevo, quindi ho modificato il codice di conseguenza.

```

47 //Controllo che esista la sessione
48 if ($_SERVER["REQUEST_METHOD"] == "POST") {
49     //Controllo se l'utente che vuole accedere alla pagina ne ha il permesso
50     //Se è un coach lo riporto alla sua pagina principale
51     if(isset($_SESSION['coach'])){
52         header("location: ". URL ."home/coach");
53     }else if(isset($_SESSION['admin']) && $_SESSION['allievo'] == null){
54         $print = isset($_SESSION['allievo']) ? " " : " " . isset($_SESSION['admin']);
55         $this->phpalert($print);
56         //header("location: ". URL ."admin/coach");
57     }else{
58         $this->phpalert("LOG");
59     }
60 }
61 //Setto le variabili per accedere al database.
62 require_once 'application/models/connection.php';
63 $connection = new Connection("efof.myd.infomaniak.com","efof_gestrip18","GestRip_Admin2018", "efof_gestrip_2018");
64
65 //Prendo i dati in entrata dal form
66 $subject = $_POST["subject"];
67 $day = $_POST["day"];
68 $hour = $_POST["hour"];
69 $learners = $_POST["learners"];
70
71 //Controllo che i campi non siano vuoti e richiamo la funzione che inserisce i valori nel database creando il corso.
72 if (strcmp($subject, "") || strcmp($day, "") || strcmp($hour, "") || strcmp($learners, "")) {
73     $requestCourse = $connection->requestCourse($subject, $day, $hour, $learners);
74 }
75
76 //Stampo un messaggio in un alert e poi riapro la pagina base
77 header("location: ". URL ."home/user/");
78 }else{
79     header("location: javascript://history.back()");
80 }

```

Oltre a racchiudere tutto il contenuto nell'if si può notare da questa immagine che ho aggiunto un else, righe 78-80, per fare in modo che se non era una richiesta corretta ritorna alla pagina precedente, questa possibilità l'ho trovata sulla pagina

["http://www.informaticando.net/post/redirect-pagina-precedente-in-php.aspx"](http://www.informaticando.net/post/redirect-pagina-precedente-in-php.aspx). Questa modifica l'ho apportata a tutte le funzioni dei controller di tutte le classi in cui c'era bisogno. Inoltre ho anche cambiato tutti i controlli per vedere se l'accesso alle funzioni era fatto dall'utente corretto come fatto la settimana scorsa.

Dopodiché ho chiesto al docente responsabile 2 informazioni, la prima era se rimandare tutti alla pagina di login se qualcuno prova ad accedere alla pagina di login/registrazione mentre c'è già una sessione in corso, e ha risposto di sì, quindi ho cambiato i controlli nella classe login di conseguenza.

```

40  if (isset($_SESSION['coach'])) {
41      session_destroy();
42  } else if (isset($_SESSION['allievo'])) {
43      session_destroy();
44  } else if (isset($_SESSION['admin'])) {
45      session_destroy();
46  }

```

La seconda domanda era come gestire il fatto che il coach e gli allievi decidono quando vogliono concluderlo, e pensavo di segnalarlo all'amministratore in qualche modo e mi è stato suggerito di creare un bottone sulla tabella dei corsi con cui un coach può concluderlo e inviare una mail all'amministratore di farlo, o che può semplicemente smettere di insegnare un determinato corso. Poi gli ho proposto di creare un bottone anche nella lista dei corsi degli allievi in modo che possano disiscriversi da un determinato corso, e mi ha detto di farlo ma aggiungendo un campo nel database che segnali se l'utente è attivo o meno nel corso. Allora ho aggiunto tutte queste informazioni nella documentazione sotto l'analisi dei requisiti con priorità 2.

Una volta finito il "colloquio" con il docente e finite le modifiche alla documentazione ho messo a posto alcuni problemi che c'erano con le pagine degli allievi.

Dopo aver sistemato i problemi ho cambiato le ultime cose per permettere all'amministratore di connettersi agli allievi e le relative azioni, per farlo ho dovuto cambiare alcune opzioni a causa dei cambiamenti che ho apportato al codice in queste ultime lezioni e poi ho iniziato la pagina per cambiare i crediti e segnalare gli errori.

La grafica della pagina è molto semplice, con le 2 possibilità.

Aggiungi crediti



Seleziona minimo di crediti prima di segnalare un problema



Questa è la pagina, ed è all'interno della pagina dell'admin ma accessibile solo dalla pagina di un utente. La parte dell'aggiunta crediti va a richiamare una funzione nel controller che prende i dati che servono e tramite una funzione in Connection va a modificare i crediti dell'utente attuale, se non è selezionato nessun utente non modifica i crediti di nessuno.

```

230 function changeCredits(){
231     //Controllo che esista la richiesta
232     if ($SERVER["REQUEST_METHOD"] == "POST") {
233         //Controllo se l'utente che vuole accedere alla pagina ne ha il permesso
234         //Se è settato un admin
235         if(!isset($_SESSION['admin'])){
236             //Riporto l'utente alla sua pagina principale
237             if(isset($_SESSION['allievo'])){
238                 header("location: ". URL . "home/user");
239             }else if(isset($_SESSION['coach'])){
240                 header("location: ". URL . "home/coach");
241             }
242         }
243
244         //Richiedo la connessione
245         require_once 'application/models/connection.php';
246         $connection = new Connection("efof.myd.infomaniak.com","efof_gestrip18","GestRip_Admin2018", "efof_gestrip_2018");
247
248         //Prendo i dati in entrata dal form
249         $credits = $_POST['credits'];
250
251         if(strcmp($credits, "")){
252             if(isset($_SESSION['allievo'])){
253                 $credits = $connection->changeCredits($credits, $_SESSION['allievo']);
254             }else if(isset($_SESSION['coach'])){
255                 $credits = $connection->changeCredits($credits, $_SESSION['coach']);
256             }
257         }
258
259         header("location: ". URL . "admin/credits");
260     }
261 }

```

Questa è la funzione nel controller dell'amministratore che prende il valore passato con il POST e poi controlla, prima di richiamare la funzione nel model, se deve modificare i crediti ad un allievo o ad un coach e passa i dati di conseguenza.

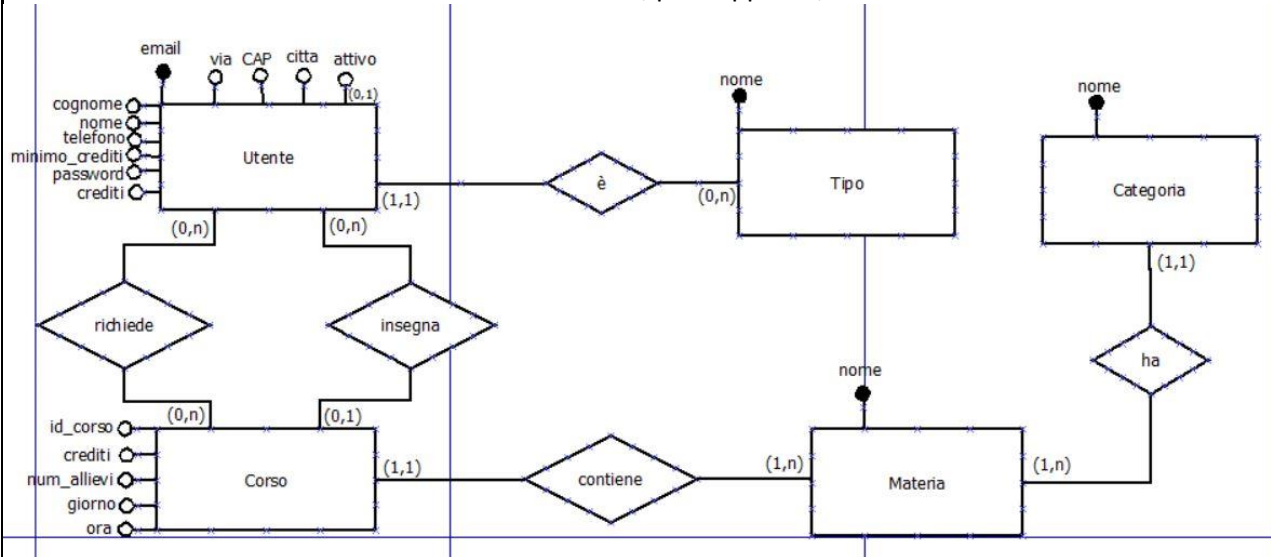
```

675 public function changeCredits($credits, $mail){
676
677     //Connect to database
678     $conn = mysqli_connect($this->servername, $this->username, $this->password, $this->dbName);
679     // Check connection
680     if (!$conn) {
681         die("Connection failed: " . mysqli_connect_error());
682     }
683
684     $sql = $conn->prepare("SELECT crediti FROM utente WHERE email LIKE ?");
685     $sql->bind_param("s", $mail);
686
687
688     if ($sql->execute()) {
689         $result = $sql->get_result();
690         $row = $result->fetch_assoc();
691     }
692     //Sommo i crediti già posseduti con quelli aggiunti
693     $credits += $row['crediti'];
694
695     //echo "Connected successfully";
696     $sql = $conn->prepare("UPDATE utente SET crediti = ? WHERE email LIKE ?");
697
698     $sql->bind_param("is", $credits, $mail);
699
700     if ($sql->execute()) {
701         // echo "New record created successfully";
702     } else {
703         //echo "Error: " . $sql . "<br>" . $conn->error;
704         $msg = "Registrazione utente non avvenuta, l'utente Ã© giÃ  esistente.";
705     }
706     $conn->close();
707 }

```

Per la modifica dei crediti prima prendo i crediti attuali dell'utente dopodich  ci sommo i crediti che si vogliono aggiungere e poi li metto nel database, in modo che vengano aggiunti, e poi ho fatto in modo che fosse possibile inserire anche i numeri negativi, se si vogliono togliere crediti.

Per la parte di impostare il minimo di crediti possibili prima della segnalazione invece ho aggiunto un attributo alla tabella utente che contiene, per l'appunto, il numero scelto.



Questa è lo schema ER con l'ultima modifica fatta al database, ovvero "minimo\_credits" aggiunto alla tabella "utente".

Dopodiché ho aggiunto alle pagine degli utenti a fianco di dove vengono mostrati i crediti posseduti 2 messaggi, che vengono mostrati solo in casi particolari.

```

13 <?php if($dataArray[0]['credits'] < $dataArray[0]['minimo_credits']): ?>
14 <label style="color: #ff0000;">ATTENZIONE: credito critico</label>
15 <?php elseif($dataArray[0]['credits'] == $dataArray[0]['minimo_credits']): ?>
16 <label style="color: #ffcc00;">ATTENZIONE: credito al limite</label>
17 <?php endif; ?>

```

Come mostrato nell'immagine, nel caso che ei crediti posseduti sono minori di quelli minimi allora segnala una scritta in rosso all'utente con scritto "ATTENZIONE: credito critico" in caso i due crediti siano uguali segnala un messaggio giallo per mettere all'allerta l'utente con scritto "ATTENZIONE: credito al limite".

Dopo aver creato la visualizzazione ho fatto una funzione nel controller dell'admin, molto simile a quella per aggiungere i crediti con 2 differenze, il nome della variabile da passare e il nome del metodo nel model.

Anche la funzione nella classe "Connection" era molto simile all'altra, la differenza era che inserivo il numero senza vedere qual è il precedente perché non devo sommarlo ma solo impostarlo.

ATTENZIONE: credito al limite Credits: 0 Dyuman Bulloni ▼

Questo è il messaggio che viene visualizzato se il numero di crediti è uguale a quello minimo impostato, in questo esempio il numero minimo è 0.

ATTENZIONE: credito critico Credits: 0

Questo è il messaggio che appare se il credito posseduto è minore di quello minimo possibile per l'utente, in questo esempio il credito minimo è 1.

#### Problemi riscontrati e soluzioni adottate

A causa dei cambiamenti e le modifiche apportate ai siti erano sorti alcuni problemi nelle pagine degli allievi, non riuscivano a prendere i dati correttamente per stamparli, o nella pagina delle statistiche non prendeva proprio niente. Nella pagina delle ricerche il problema era che non veniva per qualche errore avevo eliminato la creazione della variabile connection nel controller degli allievi e di conseguenza non poteva prendere i dati dal database. Nella pagina delle statistiche il

problema era che non avevo cambiato l'indice della variabile di sessione ed essendo ancora 'mail' non riusciva a trovare nessun corso.

Un altro problema che ho riscontrato era che quando si faceva la ricerca dei corsi su un browser diverso da Chrome dava errore 'accesso negato' e mi è stato spiegato dal docente che il problema era quando facevo la richiesta con Ajax, perché mettevo tutto il link intero ma il problema era che se un utente lo scriveva con 'www' non avrebbe funzionato perché io facevo la richiesta senza e di conseguenza non riusciva ad accedere, mi è bastato cambiare il link nel modo seguente.

```
94 | xhttp.open("POST", "/gestrip/home/searchCourses", true);
```

#### Punto della situazione rispetto alla pianificazione

Sono in linea con il gantt.

#### Programma di massima per la prossima giornata di lavoro

Portare avanti la documentazione, Test e implementazione.