

# Diario di lavoro

Luogo	SAMT
Data	16.10.2018

## Lavori svolti

A casa:

A casa ho fatto ciò che, a causa della mancanza del mio computer, non sono riuscito a completare in classe, per non perdere troppe ore.

Ho finito la presa dei corsi che non hanno un docente dal database, in modo che potessi mostrarli tutti nella tabella.

```

282 public function getPendingCourses() {
283     //Connetto al database
284     $conn = mysqli_connect($this->servername, $this->username, $this->password, $this->dbName);
285
286     //Prendo i dati dell'utente in base alla mail
287     $sql = $conn->prepare("SELECT * FROM corso WHERE mail_insegnante IS NULL");
288
289     $courses = array();
290     if ($sql->execute()) {
291
292         if ($sql->execute()) {
293             $result = $sql->get_result();
294             while ($row = $result->fetch_assoc()) {
295                 array_push($courses, $row);
296             }
297         }
298     } else {
299         return "Error: <br>" . $conn->error . " . $sql . "
300     }
301     return $courses;
302 }

```

La funzione soprastante è quella nella classe "Connection" che viene richiamata dal controller e si occupa di comunicare con il database, in questo caso specifico per prendere i valori dei corsi che ancora non hanno un insegnante assegnato e inserire tutti i dati in un array che poi ritornerà a chi la richiama.

```

24 public function getPendingCourses() {
25     if (!isset($_SESSION['mail'])) {
26         header("location: ". URL);
27     }
28
29     //Setto le variabili del database a cui faro la richiesta
30     require_once 'application/models/connection.php';
31     $connection = new Connection("efof.myd.infomaniak.com", "efof_gestrip18", "GestRip_Admin2018", "efof_gestrip_2018");
32
33     return $connection->getPendingCourses();
34 }

```

Questa è la funzione che viene richiamata prima di creare il select nella pagina, si occupa di dare i dati del database a cui fare riferimento per prendere i dati e richiamare la funzione che si occupa di prendere i dati dal database e metterli in un database prima di ritornarli. Con questa funzione ritorno direttamente il ritorno di quella nella classe "Connection".

Una volta finita la logica mi sono occupato di gestire la grafica, ovvero come e cosa veniva visualizzato nel select. Per provarlo ho dovuto creare manualmente nel database un corso senza settare nessun insegnante.

## Corsi pendenti

Materia	giorno	ora	crediti	numero massimo allievi
Etica Professionale	Lunedì	3	3	2

## Insegna

Questa è la visualizzazione finale del select, che forse verrà allungata in futuro.

Come si può vedere non sono allineate le colonne, questo perché le option hanno un problema, ovvero che non ci si può inserire niente dentro oltre a del semplice testo, questo mi ha tagliato le mani e impedito di poter fare una visualizzazione allineata, di conseguenza questo è il metodo migliore che ho trovato.

Una volta finito anche con la grafica ho implementato la parte di codice che inserisce l'utente nel database come nuovo coach di un corso una volta che selezione e preme il bottone.

```

69 public function setTeacher(){
70     //Controllo che la sessione sia attiva e esista la variabile
71     if (!isset($_SESSION['mail'])) {
72         header("location: ". URL);
73     }
74
75     //Prendo i dati in entrata dal form
76     if ($_SERVER["REQUEST_METHOD"] == "POST") {
77         $pendingCourse = $_POST["pendingCourse"];
78     }
79
80     //Setto le variabili del database a cui farò la richiesta
81     require_once 'application/models/connection.php';
82     $connection = new Connection("efof.myd.infomaniak.com", "efof_gestrip18", "GestRip_Admin2018", "efof_gestrip_2018");
83
84     if($pendingCourse != ''){
85         $setTeacher = $connection->setCourseTeacher($pendingCourse);
86     }
87
88     $this->phpAlert($setTeacher);
89     $this->pendingCourse();
90 }

```

Questa è la funzione che viene richiamata dal form della pagina una volta selezionato un corso e premuto il bottone di submit, e prende il valore dell'id del corso selezionato per poi passarlo alla funzione di connessione con il database che inserirà l'informazione in quest'ultimo.

```

301 public function setCourseTeacher($pendingCourse){
302     //Connetto al database
303     $conn = mysqli_connect($this->servername, $this->username, $this->password, $this->dbName);
304
305     //Setto la variabile del corso richiesto con la mail del docente corrente
306     $sql = $conn->prepare("UPDATE corso SET mail_insegnante = ? WHERE id_corso = ?");
307     $sql->bind_param("si", $_SESSION['mail'], $pendingCourse);
308
309     //Ritorno un messaggio se la query va a buon fine o un errore in caso contrario
310     if ($sql->execute()) {
311         return "Sei stato impostato come docente per questo corso";
312     }else{
313         return "Error: <br>" . $conn->error."/". $sql . "
314     }
315
316     $conn->close();
317 }

```

Questa funzione viene richiamata dal controller e va ad inserire la mail dell'utente loggato come "mail\_insegnante" del corso con l'id che gli viene passato come argomento.

Come si può notare alla fine della funzione nel controller, quella che viene richiamata subito dopo la premuta del bottone di submit, prima di ricaricare la pagina dei corsi pendenti ci viene richiamata la funzione "phpAlert".

```

97 public function phpAlert($msg) {
98     echo '<script type="text/javascript">alert("'" . $msg . '"</script>';
99 }

```

Questa funzione è molto semplice, si occupa semplicemente di aprire un alert nella pagina per segnalare un dato messaggio. In questo caso è richiamata per segnalare che il corso è stato segnalato come insegnato dal docente, o un errore se dovesse esserci.

A scuola:

Appena aperto il sito mi sono accorto che la pagina iniziale, di login/registrazione non funzionava correttamente, allora ho guardato che problemi aveva e ho risolto.

Dopodiché ho notato che, a causa del malfunzionamento di bootstrap, nella parte di registrazione il select che controllava il tipo era tornato normale senza più la formattazione adattata, allora ho guardato un po' su internet per vedere come metterlo a posto e l'ho fatto. Una volta risolti tutti i problemi ho deciso di iniziare a lavorare alla prima pagina dell'utente, ovvero "chiedi corso", perché in questo mod potevo provarle contemporaneamente, essendo che sono "colegate", allora l'ho creata, la grafica simile a quella della pagina di creazione dei corsi dei coach ma senza la scelta dei crediti, e l'header è diverso perché fa riferimento a link diversi.

Progetto Esercizario - Richiedi corso

Crediti: 0 Giairo Mauro ▼

Richiedi corso  
Ricerca corso  
Mostra statistiche

Questa è un'immagine dell'header degli allievi, molto simile a quello dei coach ma con link e titoli diversi

Materia:

Scegli materia...

Giorno:

Orario:

Scegli giorno...

Scegli orario...

Numero Allievi

0

Submit

Questa pagina è la prima che appare alla vista dell'allievo una volta fatto il login, la pagina di richiesta di un corso, in cui mancano alcune opzioni rispetto a quella di inserimento dei corsi dei coach, perché sono opzioni che gli allievi non possono decidere. Può selezionare tutte le impostazioni che vuole per il corso e poi, premendo il bottone submit, richiederlo e inserirlo nei corsi pendenti dei coach.

Per fare questa pagina ho utilizzato la stessa funzione di creazione del corso nella classe "Connection" cambiando alcune opzioni.

```

255 public function createCourse($subject, $day, $hour, $learners, $credits = 0, $mail_insegnante = null){

```

Inizialmente cambiano gli argomenti, i crediti e la mail le ho settate con un valore di default perché se la richiesta arriva da un allievo quei valori potrebbero non esserci.

```

265     if($mail_insegnante != null){
266         //Query per inserire i valori con il docente
267         $sql = $conn->prepare("INSERT INTO corso (mail_insegnante, giorno, ora, crediti, num_allievi, nome_materia)
268         VALUES (?, ?, ?, ?, ?)");
269         $sql->bind_param("ssiiis", $mail_insegnante, $day, $hour, $credits, $learners, $subject);
270     }else{
271         //Query per inserire i valori con il docente
272         $sql = $conn->prepare("INSERT INTO corso (giorno, ora, num_allievi, nome_materia)
273         VALUES (?, ?, ?, ?)");
274         $sql->bind_param("ssis", $day, $hour, $learners, $subject);
275     }
276 }
277

```

Questa è parte principale che, come si può vedere se si confronta con l'ultima mostrata, è cambiata abbastanza. C'è un controllo che si assicura che ci sia una mail, se c'è fa ciò che faceva prima, in caso contrario fa l'insert senza crediti e mail dell'insegnante, che verranno settati nella pagina dei corsi pendenti dai coach.

Alla fine della funzione viene gestito il ritorno come per i corsi pendenti.

```

9 public function askForCourse() {
10     //Setto le variabili per accedere al database.
11     require_once 'application/models/connection.php';
12     $connection = new Connection("efof.myd.infomaniak.com", "efof_gestrip18", "GestRip_Admin2018", "efof_gestrip_2018");
13
14     //Prendo i dati in entrata dal form
15     if ($_SERVER["REQUEST_METHOD"] == "POST") {
16         $subject = $_POST["subject"];
17         $day = $_POST["day"];
18         $hour = $_POST["hour"];
19         $learners = $_POST["learners"];
20     }
21
22     //Prendo tutti i valori dell'utente
23     $userData = $connection->getUser($_SESSION['mail']);
24
25     //Controllo che i campi non siano vuoti e richiamo la funzione che inserisce i valori nel database creando il corso.
26     if (strcmp($subject, "") || strcmp($day, "") || strcmp($hour, "") || strcmp($learners, "")) {
27         $createCourse = $connection->createCourse($subject, $day, $hour, $learners, null);
28     }
29
30     //Stampo un messaggio in un alert e poi riapro la pagina base
31     $this->phpAlert($createCourse);
32     header("location: ". URL . "home/user/");
33 }

```

Questa è la funzione che viene richiamata dal form della pagina, in cui viene richiamata la classe Connection e presi i dati del form, poi, se tutti i dati son validi, richiama la funzione "createCourse" così da aggiungere il corso richiesto dall'allievo.

Alla fine si vede che viene richiamato un alert che stampa il messaggio di ritorno e poi riportato alla pagina principale.

Per provare se funzionava tutto ciò che ho fatto fino ad ora, oltre a controllare il database, ho fatto login con l'utente coach di test per vedere se venivano mostrati tutti i corsi che l'utente richiedeva, ma mi sono accorto di non aver gestito un'opzione, che 2 utenti diversi non possono fare login nello stesso browser, perché la variabile di sessione viene cambiata con l'ultimo utente loggato, e quindi dovevo gestire quell'"errore", ma prima di farlo ho voluto provare il sito allora ho loggato in un altro browser e tutto funzionava correttamente su entrambi i tipi di utente, dopo aver risolto alcuni "errori".

Dopodiché ho gestito il fatto che non possono accedere 2 utenti diversi al sito dallo stesso browser. L'ho gestito mettendo un controllo nella funzione principale prima di aprire le pagine del login.

```

19     if (isset($_SESSION['mail'])) {
20         //Prendo tutti i valori dell'utente
21         $userData = $connection->getUser($_SESSION['mail'], 'nome_tipo');
22
23         //Se c'è già dentro un coach o un allievo riporti alla pagina principale
24         if(!strcmp($userData, "coach")){
25             header("location: ". URL ."home/coach");
26         }elseif(!strcmp($userData, "allievo")){
27             header("location: ". URL ."home/user");
28         }
29         //Se si tratta di un amministratore chiudo la connessione e torno alla pagina di login
30     }else{
31         session_destroy();
32     }

```

Questo è l'if che, inizialmente controlla se esiste già la una sessione aperta, in caso di risposta positiva prende il tipo dell'utente in sessione e in base a quale è lo manda alla sua pagina di home, a parte se si tratta dell'amministratore, in quel caso chiude la sessione e va alla pagina di login, questo per evitare che se l'amministratore si dimentica di fare il login il prossimo che entra possa entrare nella sua pagina.

Dopo diversi test mi sono accorto di una falla nello spostamento tra le pagine, ovvero che se faceva un login il un allievo, o rispettivamente un coach, semplicemente inserendo il link di una determinata pagina potevano entrarci, cosa che era sbagliata perché un allievo non può entrare nelle pagine di un coach e viceversa, così ho risolto il problema semplicemente aggiungendo un controllo prima di aprire le pagine, molto simile a quello per controllare l'entrata di più utenti diversi ma con una piccola differenza.

```

18     }else{
19         //Prendo tutti i valori dell'utente
20         $userData = $connection->getUser($_SESSION['mail'], 'nome_tipo');
21
22         //controlla se ha fatto il login un coach
23         if(!strcmp($userData, "allievo")){
24             header("location: ". URL ."home/user/");
25         }else if(!strcmp($userData, "amministratore")){
26             header("location: ". URL ."home/admin/");
27         }
28     }

```

In questo caso anche l'amministratore viene reindirizzato, perché, anche se può accedere alle pagine degli allievi e dei coach, avrà dei link predefiniti per farlo.

Infine ho aggiunto un input nella pagina dei corsi pendenti, in modo che il docente possa decidere quanti crediti costa il corso, essendo che l'allievo non può deciderlo. Ho anche aggiunto tutto ciò che ne concerneva a livello di inserimento nel database.



### Corsi pendenti

Materia	giorno	ora	crediti	numero massimo allievi
Educazione fisica	lunedì	9	0	2
Etica professionale	mercoledì	17	0	1
Etica professionale	mercoledì	16	0	1

#### Crediti per ora

Insegna

Questa è la visualizzazione della pagina definitiva.

#### Problemi riscontrati e soluzioni adottate

A casa:

Ho avuto alcuni problemi nel prendere i dati dei corsi che non avevano un insegnante, in cui la colonna "mail\_insegnante" è settata a "NULL", perché la richiamavo nel modo sbagliato, al posto di scrivere nella select "WHERE mail\_insegnante IS NULL" scrivevo "WHERE mail\_insegnante = NULL" e con l'uguale al posto di "IS" non poteva controllare i valori nulli, quindi una volta modificato quello tutto funzionava correttamente.

A scuola:

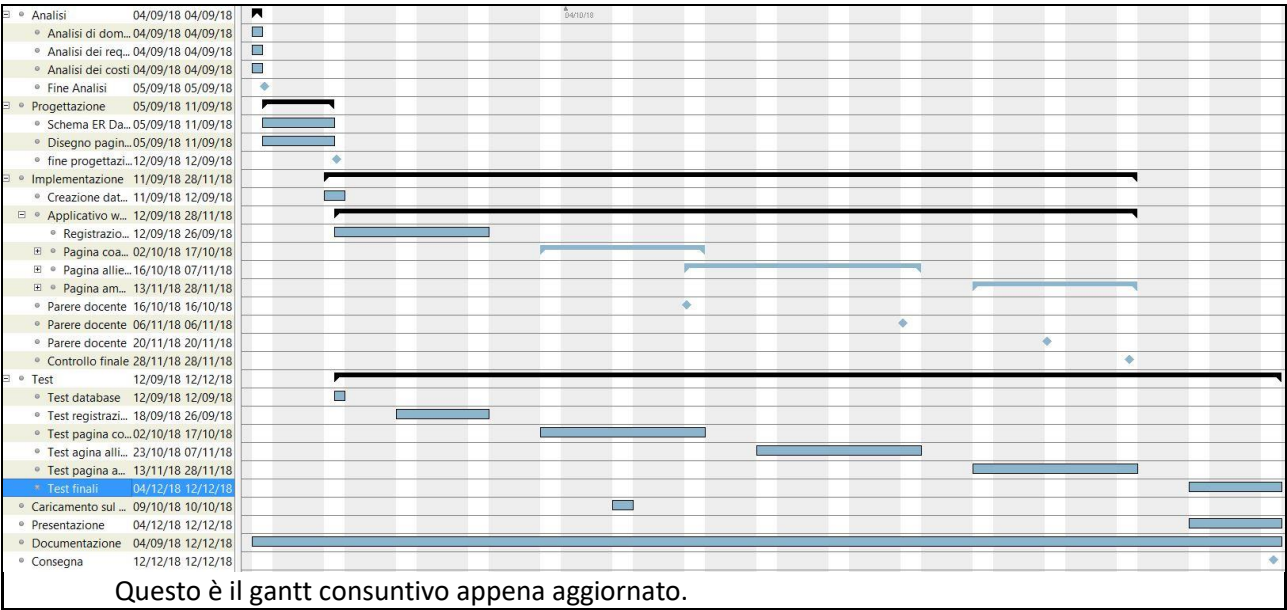
La pagina di login/registrazione non funzionava più, ovvero non cambiava più la tab, e ho notato, aprendo la console di chrome che non caricava correttamente il file JavaScript perché prendeva il link come errato, allora l'ho cambiato da "<http://saminfo.ch/gestrip/views/login/javascript/login.js>" a "<http://saminfo.ch/gestrip/application/views/login/javascript/login.js>" nonostante il primo link avrebbe dovuto prenderlo giusto.

Un altro problema era la grafica del select nella tab di registrazione, che non era corretta a causa del malfunzionamento di bootstrap, per motivi ancora sconosciuti. Per risolvere ho guardato su internet e ho visto che dovevo aggiungere la classe nella select e non solo nel div che la conteneva, quindi l'ho fatto, questa soluzione l'ho trovata a questo sito "<https://bootsnipp.com/snippets/E1BM>".

Ho avuto alcuni errori nella creazione della richiesta dei dati, ma sempre gli stessi, funzioni utilizzate o richiamate nel modo sbagliato, o variabili errate, poi con diverse prove nel database e diversi test sono riuscito a risolvere sempre nello stesso modo.

#### Punto della situazione rispetto alla pianificazione

Non ho seguito la progettazione del gantt perché era meglio se iniziavo già la pagina dell'utente per testare alcune cose in quella del coach, quindi non sono indietro rispetto alla progettazione iniziale ma sono ad un livello diverso.



Programma di massima per la prossima giornata di lavoro
Fare la pagina di statistiche.
Iniziare la pagina di ricerca del corso dell'allievo.