

# Diario di lavoro

|       |            |
|-------|------------|
| Luogo | SAMT       |
| Data  | 20.11.2018 |

## Lavori svolti

Questa mattina mi sono accorto che mancava un controllo, ovvero quando nella registrazione veniva creato un utente che già esisteva, tutto andava come se niente fosse, vengono inviate le mail per attivare e quelle di conferma che l'utente è stato attivato, l'unica cosa che non funziona è che non viene veramente inserito l'utente nel database, perché giustamente esiste già e la chiave sarebbe doppia. Per ovviare a questo problema ho fatto in modo che con un controllo in JavaScript, tramite un bottone a fianco al campo in cui bisogna inserire l'email per registrarsi, venga controllato che la mail scritta non esista già nel database.

Indirizzo Mail

Controlla

Questa è la visualizzazione del bottone che va a controllare se la mail esiste già o meno, e deve essere premuto in qualunque caso per fare in modo che il bottone di registrazione venga attivato o meno.

```

68 function checkMail(){
69     var mailExists = document.getElementById('mailExists');
70     xhttp.onreadystatechange = function() {
71         if (this.readyState === 4 && this.status === 200) {
72             //Variabile che prende l'input della mail
73             var mail = document.getElementById('mail');
74
75             //Prendo tutti i valori passati dalla pagina
76             var results = JSON.parse(xhttp.responseText);
77
78             //Controllo che l'email inserita non esista già nel database
79             for(var i = 0; i < results.length; i++){
80                 //Se la mail esiste già cambio la variabile di controllo e smetto di controllare segnalando il messaggio d'errore
81                 if(results[i]['email'] == mail.value){
82                     mailExists.hidden = false;
83                     checkMailE = false;
84                     break;
85                 }
86                 //Se la mail non esiste nascondo il messaggio d'errore e cambio la variabile di controllo.
87                 else{
88                     checkMailE = true;
89                     mailExists.hidden = true;
90                 }
91             }
92             confirm();
93         }
94     };
95     xhttp.open("POST", "/gestrip/registration/checkMail/", true);
96     xhttp.send();

```

Questa è la funzione di JavaScript che viene richiamata al click del bottone di controllo.

Tutto ciò che fa è prendere la mail che l'utente scrive, poi mandare una richiesta a una funzione che prende tutte le mail di tutti gli utenti, con la funzione "getUser" in model, e ritorna tutto in json, una volta che la funzione nel model stampa tutto in JavaScript prendo ciò che è stato stampando lo ritrasformo in modo da avere tutto in un array, riga 75, e poi passo tutti i dati e controllo se la mail scritta è una di quelle già esistenti nel database, in quel caso stampo un messaggio d'errore nella pagina e smetto di controllare, se invece nessuna delle mail nel database è come quella scritta trasformo una variabile di controllo in true, che significa che la mail è valida, e richiamo la funzione "confirm()" che si occupa di controllare se attivare o meno il bottone di registrazione.

Dopodiché ho completato la pagina dei corsi dell'amministratore, in cui mancava la possibilità di cercare i corsi tramite la materia, per farlo ho utilizzato le stesse funzioni JavaScript delle pagine di ricerca dell'allievo cambiando le informazioni da cercare, infatti ho dovuto modificare delle impostazioni delle funzioni che prendono i corsi nella classe "Connection" perché prima era utilizzabile solo dall'allievo, essendo che ho cambiato la variabile di sessione che andava a prendere con quella dell'allievo, allora ho fatto in modo che a dipendenza di quale tipo di utente andasse a richiamarlo passasse la variabile giusta.

```
if(isset($_SESSION['admin']) && !isset($_SESSION['allievo'])) {
    $courses = $connection->searchCourses($subject, $_SESSION['admin']);
} else {
```

In questo modo vado a controllare che son entrato con l'amministratore e non con l'allievo con un amministratore, in modo che vada a cercare i corsi che deve cercare l'amministratore, ovvero tutti quanti come mostrato anche nell'immagine seguente.

```
585 if($user == 'amministratore') {
586     //Prendo un array con tutti i corsi
587     $courses = $this->getCourses();
588 } else if($user == 'allievo') {
589     //Prendo un array con tutti i corsi che impara l'allievo
590     $courses = $this->getLearnedCourses($mail);
591 } else if($user == 'coach') {
592     //Prendo un array con tutti i corsi insegnati dal coach
593     $courses = $this->getLearnedCourses($mail);
594 }
```

Nel file model viene controllato quale utente fa la ricerca dei corsi, così per ogni tipo di utente vengono cercati corsi a cui può accedere. Per fare questa selezione ho preso il tipo con la funzione "getUser" perché utilizzare la variabile di sessione sarebbe stato più complicato, ma nel farlo ho avuto alcuni errori che ho dovuto risistemare, come spiegato nella sezione "Problemi riscontrati e soluzioni adottate".

Alla fine ho pensato di inserire i dati, nella view dei corsi dell'amministratore, in una tabella, essendo che non deve svolgerci delle azioni ma solamente vederle e vedere le ricerche.

Materia:

Categorie

| Materia             | giorno  | ora | crediti | numero massimo allievi | Coach responsabile           |
|---------------------|---------|-----|---------|------------------------|------------------------------|
| Economia e diritto  | lunedì  | 8   | 3       | 2                      | dyuman.bulloni@samtreveno.ch |
| Italiano MP         | lunedì  | 17  | 0       | 1                      | null                         |
| Italiano MP         | lunedì  | 17  | 5       | 1                      | dyuman.bulloni@samtreveno.ch |
| Educazione fisica   | lunedì  | 10  | 0       | 3                      | null                         |
| Inglese MP          | lunedì  | 9   | 0       | 2                      | null                         |
| Italiano MP         | lunedì  | 9   | 0       | 2                      | null                         |
| Etica professionale | giovedì | 12  | 0       | 1                      | null                         |

Questa è la pagina che viene mostrata, come si può notare sono mostrati inizialmente tutti i corsi esistenti con tutte le relative informazioni, ma c'è la possibilità di mostrare solo i corsi di una determinata materia. Magari se riuscirò ad aggiungere la possibilità di concludere un corso allora inserirò una colonna in più in cui si potrà vedere se il corso è ancora attivo oppure no. Dove c'è scritto "null" sotto la colonna del "coach responsabile" vuol dire che quel corso non ha ancora un docente responsabile.

La struttura dei dati è fatta in modo, alla fine, molto semplice, la base per prendere i dati è la stessa, cambia solo il modo in cui vengono mostrati, di base c'è all'interno della pagina una tabella contenente l'intestazione con il nome delle colonne, e un corpo, inizialmente vuoto, poi viene richiamata una funzione JavaScript che inserisce tutti i corsi e alla ricerca viene svuotata e riempita con i corsi della materia richiesta.

```

184     var tbody = document.getElementById('courses');
185     tbody.innerHTML = "";
186
187     //Inserisco tutti le righe con i relativi corsi.
188     for(var i = 0; i < results.length; i++){
189         //Creo la riga in cui inserire i campi
190         var tr = document.createElement("tr");
191
192         //Creo una nuova colonna e inserisco i dati, per ogni informazione richiesta
193         var td = document.createElement("td");
194         td.appendChild(document.createTextNode(results[i]['nome_materia']));
195         tr.appendChild(td);
196         var td = document.createElement("td");
197         td.appendChild(document.createTextNode(results[i]['giorno']));
198         tr.appendChild(td);
199         var td = document.createElement("td");
200         td.appendChild(document.createTextNode(results[i]['ora']));
201         tr.appendChild(td);
202         var td = document.createElement("td");
203         td.appendChild(document.createTextNode(results[i]['crediti']));
204         tr.appendChild(td);
205         var td = document.createElement("td");
206         td.appendChild(document.createTextNode(results[i]['num_allievi']));
207         tr.appendChild(td);
208         var td = document.createElement("td");
209         td.appendChild(document.createTextNode(results[i]['mail_insegnante']));
210         tr.appendChild(td);
211
212         //Inserisco la riga nella tabella
213         tbody.appendChild(tr);
214     }

```

Quella mostrata sopra è la parte, aggiornata, di inserimento dei dati, inizialmente il corpo viene svuotato da ciò che c'è già dopodiché viene riempito nuovamente con le nuove informazioni, creando l'elemento della riga da inserire nel corpo e, per ogni riga, le colonne che vanno inserite al suo interno, alla fine la riga viene inserita nel corpo e si passa alla prossima, se ce ne dovesse essere una.

Anche nel fare questo ho avuto alcuni problemi che mi hanno preso del tempo per risolverli.

Mentre facevo dei test, per poter completare anche la doc in modo completo, ho notato che, nel cambiamento che avevo fatto alla funzione "getUser" avevo scatenato dei problemi, ma è stato molto semplice sistemare, mi è bastato cambiare una variabile che ritornava un valore nella formattazione errata, e sono riuscito a sistemare.

#### Problemi riscontrati e soluzioni adottate

Un problema che ho avuto è stato quando facevo la richiesta di un dato specifico con la funzione getUser, perché nella prepared statement facevo inserire direttamente la variabile nella stringa

con il select, ma in questo caso rendevo inutile tutto, perché era comunque possibile fare delle SQLinjection, allora ho cercato un modo per ovviare a questo problema ma non potevo farlo normalmente come se fosse una stringa con il metodo "bind\_param" perché veniva inserita con gli apici e quindi mi ritornava una stringa con scritto il nome del dato richiesto invece che il dato. La ricerca di una soluzione mi ha occupato diverso tempo, alla fine, sotto anche il consiglio del docente, ho fatto con uno switch che in base all' tipo d dato richiesto fa una select diversa, come mostrato nell'immagine che segue.

```

264 switch($data){
265     case "nome":
266         //Prendo il dato richiesto dell'utente in base alla mail
267         $sql = $conn->prepare("SELECT nome FROM utente WHERE nome_tipo = ?");
268         break;
269     case "cognome":
270         //Prendo il dato richiesto dell'utente in base alla mail
271         $sql = $conn->prepare("SELECT cognome FROM utente WHERE nome_tipo = ?");
272         break;
273     case "telefono":
274         //Prendo il dato richiesto dell'utente in base alla mail
275         $sql = $conn->prepare("SELECT telefono FROM utente WHERE nome_tipo = ?");
276         break;
277     case "email":
278         //Prendo il dato richiesto dell'utente in base alla mail
279         $sql = $conn->prepare("SELECT email FROM utente WHERE nome_tipo = ?");
280         break;
281     case "crediti":
282         //Prendo il dato richiesto dell'utente in base alla mail
283         $sql = $conn->prepare("SELECT crediti FROM utente WHERE nome_tipo = ?");
284         break;
285     case "minimo_crediti":
286         //Prendo il dato richiesto dell'utente in base alla mail
287         $sql = $conn->prepare("SELECT minimo_crediti FROM utente WHERE nome_tipo = ?");
288         break;
289     case "via":
290         //Prendo il dato richiesto dell'utente in base alla mail
291         $sql = $conn->prepare("SELECT via FROM utente WHERE nome_tipo = ?");
292         break;
293     case "CAP":
294         //Prendo il dato richiesto dell'utente in base alla mail
295         $sql = $conn->prepare("SELECT CAP FROM utente WHERE nome_tipo = ?");
296         break;
297     case "citta":
298         //Prendo il dato richiesto dell'utente in base alla mail
299         $sql = $conn->prepare("SELECT citta FROM utente WHERE nome_tipo = ?");
300         break;
301     case "nome_tipo":
302         //Prendo il dato richiesto dell'utente in base alla mail
303         $sql = $conn->prepare("SELECT nome_tipo FROM utente WHERE nome_tipo = ?");
304         break;

```

Questa è solo una parte dello switch che gestisce la richiesta, e ne ho dovuti creare 2 perché uno fa la ricerca in base al tipo di utente l'altro in base alla mail.

Essendo che non avevo mai veramente usato i nodi di JavaScript e i metodi, "appendChild" ho dovuto informarmi bene prima di poterli utilizzare nel modo corretto per questo progetto, quindi inizialmente ho avuto dei problemi, perché non riuscivo ad inserire nel modo corretto le informazioni nella tabella, non sapendo in che modo lavoravano gli appendChild, ma una volta capito sono riuscito a mettere "in ordine".

#### Punto della situazione rispetto alla pianificazione

Sono in anticipo rispetto a quanto scritto sul gantt, essendo che ho finito tutte le pagine indispensabili e sul gantt c'è scritto che avrei ultimato le pagine dell'amministratore la settimana prossima.

Programma di massima per la prossima giornata di lavoro

Portare avanti la documentazione, Test e implementazione.