

Diario di lavoro

Luogo	SAMT
Data	05.02.2019

Lavori svolti

La prima cosa che ho fatto questa mattina è stata correggere degli errori trovati ho segnalati dal docente. Il primo stava nel database, nella tabella dei prodotti mancava l'immagine, allora ho inserito un campo che andrebbe a essere riempito con il percorso in cui trovare l'immagine per quel determinato prodotto, al posto di inserire le immagini direttamente nel database e appesantirlo.

Quello qui sotto è il nuovo schema E-R modificato.

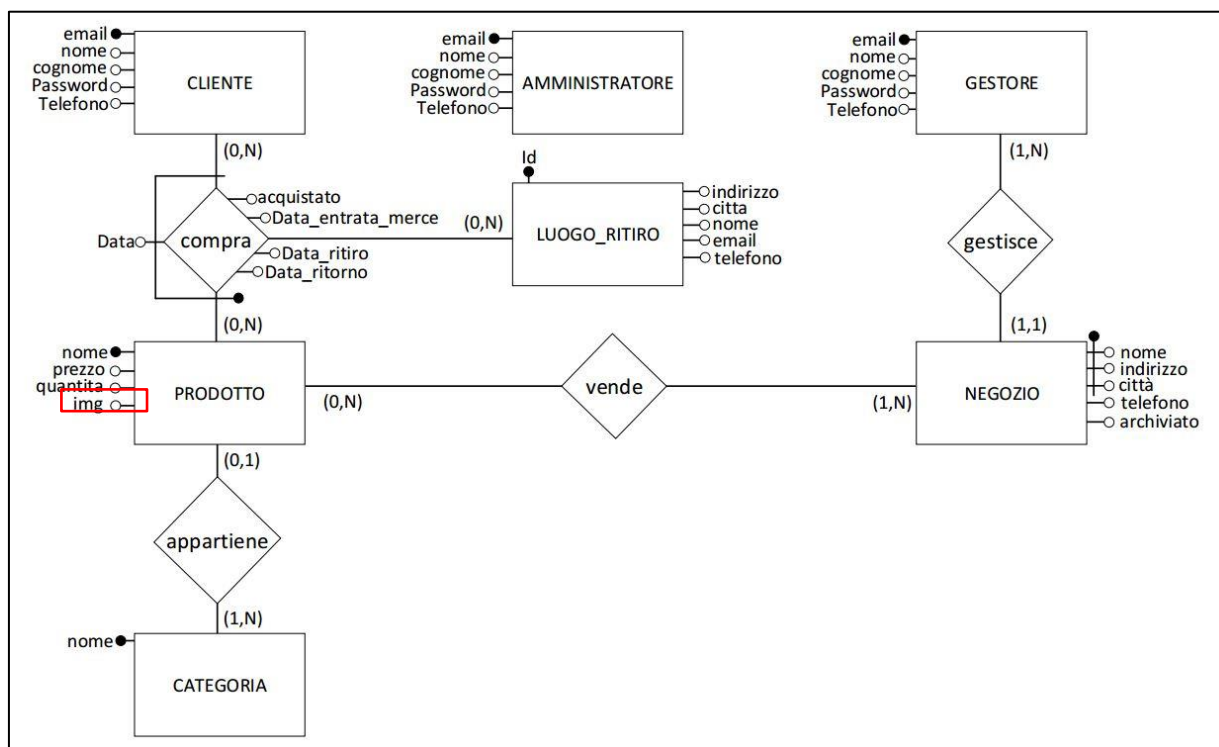


Figura 1 schema E-R con immagine

L'altro errore da mettere a posto mi è stato segnalato dal mio docente, all'interno della funzione della registrazione c'era un'incongruenza fuori dall'if.

```

11 public function createC($name, $surname, $mail, $phone, $pass)
12 {
13     require_once 'application/models/customer.php';
14     $customer = new Customer();
15
16     if (strcmp($name, str2: "") || strcmp($surname, str2: "") || strcmp($mail, str2: ""))
17     {
18         $customer->addUser($name, $surname, $mail, $phone, $pass);
19     }
20
21     $this->sendMail($name, $surname, $mail);
22
23     header( string: "location: ". URL ."login/loginPageC");
24 }

```

Figura 2 vecchia funzione per la creazione utente

La funzione inizialmente faceva il controllo dei campi e si assicurava che avessero un valore e se non l'avevano non creava l'utente, ma comunque sia inviava la mail, e questo è errato dal momento che non viene creato l'utente ma viene inviata l'email che è stato creato, quindi la semplice correzione che ho applicato è stata inserire il richiamo del metodo "sendMail" all'interno del controllo dei campi.

Una volta fatti questi cambiamenti ho esposto alcuni dubbi che mi erano venuti al docente sull'amministratore e i suoi compiti riguardo alla risposta "le tabelle dei parametri / configurazioni (tempi di consegna, luoghi di consegna, prodotti da mostrare pagina web, percentuali che il sito trattiene dal prezzo per il servizio)" che mi aveva dato e abbiamo risolto che i tempi di consegna e i luoghi vengono anche gestiti da lui, ma principalmente è il gestore a segnalarlo, come è giusto che sia, e possono essere modificati, e l'amministratore può vedere se vengono cambiati troppe volte e può andare a discuterne con il negoziante. I prodotti che vengono mostrati nelle pagine invece vengono decisi dall'utente invece che dall'amministratore, mentre le percentuali che il sito trattiene dall'amministratore in una pagina a parte. Dato ciò ho dovuto fare un'ulteriore modifica al database, per introdurre la percentuale che il sito trattiene, quindi l'E-R per ora definitivo è quello mostrato sotto.

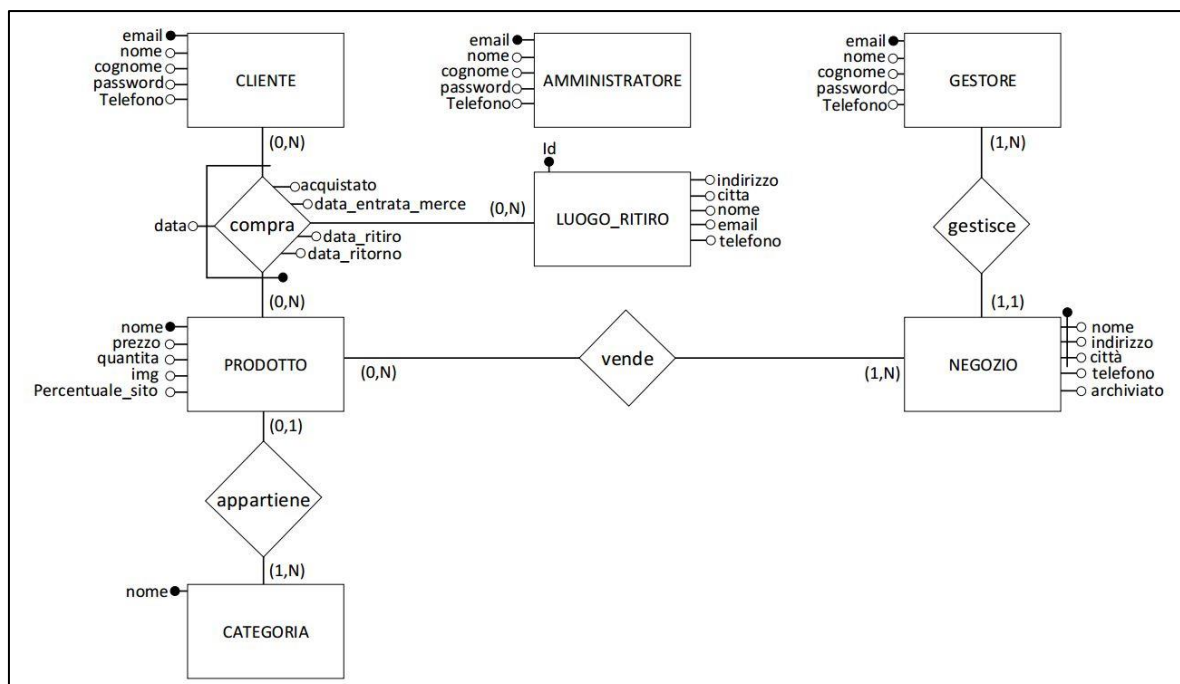
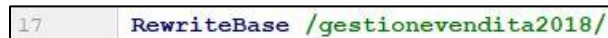


Figura 3 schema E-R con percentuale sito

Una volta completato ho messo il sito online, per farlo ho utilizzato il software “total commander” che è poi lo stesso software che uso per lavorare sui file online. Semplicemente ho fatto l’accesso all’ftp con le credenziali datemi dal professor Barchi e ho spostato al suo interno tutte le cartelle e i file della struttura, ovvero “application” con il contenuto e i file “.htaccess” e “index.php”. Una volta fatto ho dovuto riadattare le informazioni, prima di tutto ho dovuto cambiare il nome della cartella del progetto nel file “.htaccess” come mostrato nell’immagine che segue.

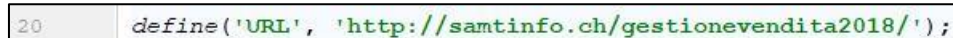


```
17 RewriteBase /gestionevendita2018/
```

Figura 4 cartella progetto online “.htaccess”

Dopo aver cambiato questo ci sono altri 2 cambiamenti da fare, il primo appartiene al database, essendo che faccio riferimento ad un database non più locale ho dovuto cambiare le variabili di base con i criteri di accesso al database con quelli datimi dal professore.

L’ultima cosa, ma non meno importante, da modificare è la variabile globale “URL” che si trova nel file “config.php” e bisogna dargli il valore del sito base a cui accedere, nel mio caso “<http://saminfo.ch/gestionevendita2018/>”, come mostrato nello screenshot che segue.



```
20 define('URL', 'http://saminfo.ch/gestionevendita2018/');
```

Figura 5 variabile URL config.php

Dopo aver fatto questo ho iniziato a lavorare alla pagina di mostra dei prodotti per i venditori. La prima cosa che ho fatto è stata modificare la lista delle categorie a sinistra, facendogli inserire automaticamente le categorie presenti nel database, per fare questo ho utilizzato Ajax, in una funzione faccio la richiesta e poi con la risposta che mi arriva compongo la lista delle categorie, la funzione è quella mostrata qui sotto.

```

8  function getCategories() {
9
10     xhttp.onreadystatechange = function () {
11         if (this.readyState === 4 && this.status === 200) {
12
13             //Prendo i valori passati dal server e li metto in un array
14             var obj = JSON.parse(xhttp.responseText);
15
16             //Prendo il corpo della tabella in cui inserire le righe
17             var tbody = document.getElementById( 'menu-content2' );
18
19             //Inserisco tutti le righe con i relativi corsi.
20             for (var i = 0; i < obj.length; i++) {
21                 //Creo la riga i cui inserire i campi e gli metto le informazioni
22                 var li = document.createElement( tagName: "li" );
23                 li.setAttribute( qualifiedName: "data-toggle", value: "collapse" );
24                 //li.setAttribute("class", "collapsed");
25
26                 //Creo una nuova colonna e inserisco il dato, per ogni informazioni richiesta
27                 var a = document.createElement( tagName: "a" );
28                 a.appendChild( document.createTextNode( obj[i][ "nome" ] ) );
29                 a.setAttribute( qualifiedName: "href", value: "#" );
30                 li.appendChild(a);
31                 console.log(tbody);
32                 //Inserisco la riga nella tabella
33                 tbody.appendChild(li);
34             }
35         }
36     }
37     xhttp.open( method: "POST", url: "/gestionevendita2018/home/getCategories", async: true );
38     xhttp.send();
39 }

```

Figura 6 funzione JavaScript inserimento categorie

Questa funzione va a fare la richiesta al controller Home, che contiene le funzioni utilizzabili da tutti, e prende quella per le categorie.

Questa funzione va a richiamare dalla classe “Category”, che è quella che si occupa di lavorare nel database con la tabella delle categorie, e prende la funzione che le ritorna tutte.

```

71  public function getCategories(){
72
73      if ($_SERVER["REQUEST_METHOD"] == "POST") {
74
75          //Prendo la classe model
76          require_once 'application/models/category.php';
77          $category = new Category();
78
79          $ccategories = $category->getCategories();
80
81          //Stampo con json i valori dei coach
82          header( string: 'Content-Type: application/json' );
83          echo json_encode($ccategories);
84      }else{
85          header( string: "location: javascript://history.back()" );
86      }
87  }

```

Figura 7 Controller getCategories

La funzione model invece va a prendere tutti i dati contenuti nella tabella, ovvero i nomi di tutte le categorie essendo quello l'unico campo esistente, e le ritorna in un array.

```

17 public function getCategories() {
18     //Connetto al database
19     $conn = $this->connection->sqlConnection();
20
21     //Prendo i dati dell'utente in base alla mail
22     $sql = $conn->prepare( statement: "SELECT * FROM categoria");
23
24     $dataArray = array();
25     //Eseguo la query
26     $sql->execute();
27
28     // Ciclo tutti i valori
29     while ($row = $sql->fetch()) {
30         array_push( &array: $dataArray, $row);
31     }
32     $conn = null;
33     return $dataArray;
34 }

```

Figura 8 model getCategories

Alla fine le categorie vengono inserite a destra di tutto e sono cliccabili, ma ancora non sono state implementate per cui non modificano la lista dei prodotti.

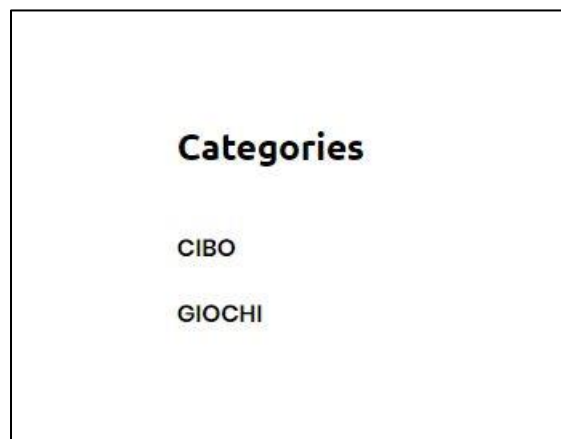


Figura 9 visualizzazione categorie

Dopo questo ho fatto la visualizzazione dei prodotti, più o meno il sistema è lo stesso, la differenza sostanziale, oltre alla tabella e di conseguenza la tabella, è la visualizzazione, perché è una struttura di "Div", ben definiti, quindi il model e il controller sono praticamente uguali a quelli della categoria con le differenze delle classi che richiamano e la tabella a cui fa riferimento, come mostrato nelle immagini qui sotto.


```

92     public function getProducts() {
93
94         if ($_SERVER["REQUEST_METHOD"] == "POST") {
95
96             //Prendo la classe model
97             require_once 'application/models/product.php';
98             $product = new Product();
99
100             $products = $product->getProducts();
101
102             //Stampo con json i valori dei coach
103             header( string: 'Content-Type: application/json');
104             echo json_encode($products);
105         }else{
106             header( string: "location: javascript://history.back()");
107         }
108     }

```

Figura 10 controller getProducts

```

17     public function getProducts(){
18         //Connetto al database
19         $conn = $this->connection->sqlConnection();
20
21         //Prendo i dati dell'utente in base alla mail
22         $sql = $conn->prepare( statement: "SELECT * FROM prodotto");
23
24         $dataArray = array();
25         //Eseguo la query
26         $sql->execute();
27
28         // Ciclo tutti i valori
29         while ($row = $sql->fetch()) {
30             array_push( &array: $dataArray, $row);
31         }
32         $conn = null;
33         return $dataArray;
34     }

```

Figura 11 model getProducts

La funzione in JavaScript invece è un po' più complicata ed è spiegata qui sotto.

```

50     var divbody = document.getElementById( elementId: 'prodContainer');

```

Figura 12 variabile div iniziale

```

53     for (var i = 0; i < obj.length; i++) {
54         //Prendo i div principale per i prodotti
55         var divContainer = document.createElement( tagName: "div");
56         divContainer.setAttribute( qualifiedName: "class", value: "col-12 col-sm-6 col-lg-4");
57         var divWrapper = document.createElement( tagName: "div");
58         divWrapper.setAttribute( qualifiedName: "class", value: "single-product-wrapper");

```

Figura 13 variabili div

La prima immagine mostra una variabile che contiene il div più grande con tutti i prodotti al suo interno. Dopodiché creo 2 div con delle classi specifiche che andranno a contenere le informazioni del prodotto.

```

60 //Creo la riga i cui inserire i campi e gli metto imposto la classe
61 var div = document.createElement( tagName: "div");
62 div.setAttribute( qualifiedName: "class", value: "product-img");
63
64 //Creo un immagine imposto il source con il dato preso ee inserisco il div nel contenitore padre
65 var img = document.createElement( tagName: "img");
66 var src = "http://saminfo.ch/gestionevendita2018/"+obj[i]["img"] ;
67 img.setAttribute( qualifiedName: "src", src);
68 div.appendChild(img);
69 divWrapper.appendChild(div);

```

Figura 14 inserimento immagine

Quest'immagine mostra la parte in cui creo un div con la classe per contenere le immagini, poi creo l'immagine imposto il percorso, che va a prendere nell'FTP e lo inserisco nel div.

```

71 //Creo un nuovo div che contiene le informazioni e lo metto nel padre
72 var div = document.createElement( tagName: "div");
73 var a = document.createElement( tagName: "a");
74 a.setAttribute( qualifiedName: "href", value: "#");
75 var h6 = document.createElement( tagName: "h6");
76 h6.appendChild(document.createTextNode(obj[i]['nome']));
77 a.appendChild(h6);
78 div.appendChild(a);
79 divWrapper.appendChild(div);

```

Figura 15 inserimento titolo

Con la parte di codice mostrata sopra vado a creare un ulteriore di con un link, che in futuro potrà alla pagina che mostra i dettagli del prodotto, e il titolo di esso al suo interno, e anche questo viene aggiunto al div padre.

```

82 var p = document.createElement( tagName: "p");
83 var prezzo = obj[i]['prezzo'] + " Fr."
84 p.appendChild(document.createTextNode(prezzo));
85 divWrapper.appendChild(p);

```

Figura 16 inserimento prezzo

Con questo creo un paragrafo da inserire e ci metto il prezzo, formattato prime (variabile "prezzo").

```

88 divContainer.appendChild(divWrapper);
89 divbody.appendChild(divContainer);
90 console.log(divbody);

```

Figura 17 unione finale

Una volta inserito tutto dentro il div "divWrapper" inserisco quest'ultimo nel div padre e il tutto in quello preso all'inizio. Questo percorso viene fatto in un for, come si è visto nella seconda riga mostrata, che passa tutti i prodotti in modo che tutti vengano inseriti con il formato corretto.



corda

10 Fr.

Figura 18 mostra prodotto

Questa è la visualizzazione di un prodotto d'esempio inserito hard coded da me.

Problemi riscontrati e soluzioni adottate

Non ho avuto particolari problemi

Punto della situazione rispetto alla pianificazione

Oggi sono arrivato a buon punto con la pagina della messa in vendita dei prodotti.

Programma di massima per la prossima giornata di lavoro

Completare la pagina di messa in vendita dei prodotti.