

Diario di lavoro

Luogo	SAMT
Data	25.03.2019

Lavori svolti

Inizialmente ho completato la modifica dei dati del prodotto che può fare il venditore sulla sua pagina, una volta cliccato sul nome del prodotto si apre un popup che permette di modificarne i dati ed eventualmente salvare. Questa pagina dava un problema nell'inserimento dei dati all'interno dei select, come spiegato nella sezione che segue questa.

Una volta fatto ho creato una funzione che elimina i dati dei select, e viene richiamata quando si chiude il popup.

```

181 function deleteCategoriesShop(ulC, ulS){
182     //Faccio passare tutti i valori del select dall'ultimo e li cancello
183     // lasciando il primo
184     var ulClength = ulC.getElementsByTagName("li").length;
185     for (var i = ulClength-1; i > 0; i--){
186         ulC.removeChild(ulC.childNodes[i]);
187     }
188
189     var ulSlength = ulS.getElementsByTagName("li").length;
190     for (var i = ulSlength-1; i > 0; i--){
191         ulS.removeChild(ulS.childNodes[i]);
192     }
193 }

```

Figura 1 deleteCategoriesShop eliminazione dati select

La funzione riceve le 2 liste e in base a quanti dati ci sono dentro grazie ai child elimina tutti i dati tranne il primo, perché è quello che segnala il titolo del select e non viene inserito dinamicamente ma è scritto hardcoded essendo sempre uguale.

Qui sotto è mostrata la parte della funzione che va ad inserire tutti i dati nei campi del popup di modifica dei prodotti.

```

288 function detailsProduct(obj) {
289     var data = obj.id.split(".");
290
291     //Prendo il corpo del div in cui inserire i campi
292     var divbody = document.getElementById( elementId: 'modSBody');
293
294     //Creo la riga in cui inserire i campi e gli metto imposto la classe
295     var div = document.createElement( tagName: "div");
296     div.setAttribute( qualifiedName: "class", value: "single_product_thumb clearfix");
297
298     //Inserisco l'immagine del prodotto
299     var img = document.getElementById( elementId: "modImage");
300     img.src = "http://saminfo.ch/gestionevendita2018/" + data[0] + "." + data[1];
301
302     //Imposto le liste di categoria e negozi
303     getCategories( list: false, ul[2]);
304     setTimeout( handler: function(){
305         getShop(window.ul[3]);
306     }, timeout: 400);
307
308     //Prendo i div e inserisco i valori li imposto come selected
309     var divUl = document.getElementsByClassName( classNames: "nice-select");
310     divUl[2].childNodes[0].innerHTML = data[2];
311     divUl[3].childNodes[0].innerHTML = data[3];
312
313     //Inserisco il nome del prodotto
314     var name = document.getElementById( elementId: "modTitle");
315     name.setAttribute( qualifiedName: "value", data[4]);
316
317     //Inserisco il prezzo del prodotto
318     var price = document.getElementById( elementId: "modPrice");
319     price.setAttribute( qualifiedName: "value", data[5]);
320
321     //Inserisco il prezzo del prodotto
322     var quantity = document.getElementById( elementId: "modQuantity");
323     quantity.setAttribute( qualifiedName: "value", data[6]);
324
325     document.getElementById( elementId: 'modifyModal').style.display = "block";

```

Figura 2 detailsProduct, mostra dei dettagli del prodotto

Anche in questa funzione utilizzo il timeout (righe 304-306) per aprire le funzioni della categoria e dei negozi, dopodiché inserisco, grazie sempre ai child, i dati nei campi in modo da far capire quali sono quelli attuali.

La modifica non sono ancora riuscito ad ultimarla perché la correzione dei problemi mi ha preso diverso tempo.

Problemi riscontrati e soluzioni adottate

Un primo problema che ho avuto è stato con l'inserimento dei dati nella pagina di modifica del prodotto, per i select non riuscivo ad inserire le informazioni nella lista, ovvero, venivano inseriti i dati ma subito tolti. Dopo un po' ho capito il problema. Per modificare i select, vengono usate 2 funzioni, una che imposta le categorie e una i negozi, quella dei negozi è quella mostrata qui sotto.

```

141 function getShop(ul) {
142     xhttp.onreadystatechange = function () {
143         if (this.readyState === 4 && this.status === 200) {
144
145             //Prendo i valori passati dal server e li metto in un array
146             var obj = JSON.parse(xhttp.responseText);
147
148             var divUl = document.getElementsByClassName( className: "nice-select");
149             divUl[1].setAttribute( qualifiedName: "style", value: "z-index:0;");
150
151             //Inserisco tutte le categorie in optio e poi nel select.
152             for(var i = 0; i < obj.length; i++){
153                 var li = document.createElement( tagName: "li");
154                 li.setAttribute( qualifiedName: "class", value: "option selected focus");
155                 li.setAttribute( qualifiedName: "style", value: "z-index:3;");
156                 li.setAttribute( qualifiedName: "data-value", value: obj[i]["nome"] + "." + obj[i]["indirizzo"] + "." + obj[i]["citta"]);
157                 li.setAttribute( qualifiedName: "id", value: obj[i]["nome"] + "." + obj[i]["indirizzo"] + "." + obj[i]["citta"]);
158                 li.appendChild(document.createTextNode(obj[i]["nome"]));
159                 ul.appendChild(li);
160             }
161         }
162     }
163     xhttp.open( method: "POST", url: "/gestionevendita2018/dealer/getShopsByDealer", async: true);
164     xhttp.send();
165 }

```

Figura 3 funzione getShop per riempire il select di negozi

La funzione riceve tramite argomento il campo che deve riempire, poi con i dati presi dalla richiesta al database lo riempie.

Il problema era che la funzione veniva richiamata sia all'apertura del popup di modifica, per il campo in quella sezione, che alla fine della funzione che prende i dati delle categorie, di conseguenza andava in conflitto e riempiva la prima volta ma poi veniva richiamata la funzione ed essendo che i vari tag "ul" hanno gli stessi nomi svuotava quello riempito in precedenza e riempiva il nuovo. Per risolvere il problema mi è bastato inserire il richiamo della funzione per il select dell'inserimento, assieme a quello delle categorie, come mostrato qui sotto.

```

19 $('#switch').click(function() {
20     modal.style.display = "block";
21     getCategories( list: false);
22     getShop(ul[1]);
23 });

```

Figura 4 apertura inserimento prodotti

Dopo aver fatto questa modifica però è tornato il problema di Ajax, che è troppo veloce e di conseguenza funziona solo l'ultimo richiamo, ma sono riuscito a fare in modo che prima di far partire la seconda funzione attenda un po', con il metodo "setTimeout" mostrato nell'immagine sotto, che aspetta 500 millisecondi e dopo esegue ciò che c'è al suo interno.

```

19 $('#switch').click(function() {
20     modal.style.display = "block";
21     getCategories( list: false, ul[0]);
22     setTimeout( handler: function(){
23         getShop(ul[1]);
24     }, timeout: 500);
25 });

```

Figura 5 apertura inserimento prodotti

L'ultimo problema che ho dovuto risolvere è stata la chiusura del popup della modifica. Questo problema era dato dal fatto che in JQuery andavo a chiudere solo il modal dell'aggiunta quindi mi è bastato aggiungere il secondo modal come mostra l'immagine sotto e tutto funziona correttamente.

```

15 $(function() {
16     // Get the modal
17     var addmodal = document.getElementById( 'addModal');
18     var modifyModal = document.getElementById( 'modifyModal');
19
20     $('#switch').click(function() {
21         addmodal.style.display = "block";
22         getCategories( list: false, ul[0]);
23         setTimeout( handler: function() {
24             getShop(ul[1]);
25         }, timeout: 500);
26     });
27
28     // When the user clicks on <span> (x), close the modal
29     $('.close').click(function() {
30         addmodal.style.display = "none";
31         deleteCategoriesShop(ul[0], ul[1]);
32         modifyModal.style.display = "none";
33         deleteCategoriesShop(ul[2], ul[3]);
34     });
35
36     // When the user clicks anywhere outside of the modal, close it
37     window.onclick = function(event) {
38
39         if (event.target == addmodal || event.target == modifyModal) {
40             addmodal.style.display = "none";
41             deleteCategoriesShop(ul[0], ul[1]);
42             modifyModal.style.display = "none";
43             deleteCategoriesShop(ul[2], ul[3]);
44         }
45     }
46 }

```

Figura 6 JQuery apertura e chiusura modal

Punto della situazione rispetto alla pianificazione

Ho completato la parte del venditore.

Devo ancora gestire il caso in cui 2 negozi facciano riferimento allo stesso prodotto e in uno dei 2 diminuisca la quantità.

Sono ancora in ritardo di una settimana.

Programma di massima per la prossima giornata di lavoro

Iniziare la pagina dell'amministratore.