

# Diario di lavoro

Luogo	SAMT
Data	27.03.2019

## Lavori svolti

La prima cosa che ho fatto questa mattina è stata portare avanti il gantt consuntivo in modo da avere una corretta prospettiva della situazione, come mostrato nella sezione "Punto della situazione rispetto alla pianificazione".

Dopodiché ho fatto la mostra dei dettagli nella pagina dei clienti.

Come prima cosa ho messo il "div" che conterrà dati e apparirà quando si preme sul nome del prodotto di cui si vogliono vedere i dettagli, come mostrato nella foto.

```

49 <div id="detailsModal" class="modal" style="...">
50
51 <!-- Modal content -->
52 <div class="modal-content">
53   <span class="close">x</span>
54   <!-- ##### Single Product Details Area Start ##### -->
55   <section class="single_product_details_area d-flex align-items-center SBody">
56
57     <!-- Single Product Thumb -->
58     <div class="single_product_thumb clearfix">
59       <label for="file-input">
60         <img id="image">
61       </label>
62     </div>
63
64     <!-- Single Product Description -->
65     <div class="single_product_desc clearfix">
66       <h5>Inserimento prodotto</h5>
67       <br><br>
68       <div class="widget-desc">
69         <ul id="detailsList">
70           <li id="category"></li><br>
71           <li id="title"></li><br>
72           <li id="price"></li><br>
73           <li id="quantity"></li><br>
74         </ul>
75       </div>
76     </div>
77   </section>
78   <!-- ##### Single Product Details Area End ##### -->
79 </div>
80 </div>

```

*Figura 1 contenitore mostra dettagli prodotto*

Come si può notare nella lista manca il campo per il negozio, questo perché quel campo viene inserito da JavaScript direttamente, essendo che può essere sia singolo che multiplo, perché ci possono essere più prodotti che vendono lo stesso prodotto.

La funzione che viene richiamata al click sul nome del prodotto è mostrata nell'immagine qui sotto.

```

522 function detailsProduct(obj) {
523     var prodKey = obj.id.split(".");
524     var data;
525
526     //Prendo il corpo del div in cui inserire i campi
527     var divbody = document.getElementById( elementId: 'modSBody');
528
529     //Richiedo i dati del prodotto richiesto
530     xhttp.onreadystatechange = function () {
531         if (this.readyState === 4 && this.status === 200) {
532
533             //Prendo i valori passati dal server e li metto in un array
534             var obj = JSON.parse(xhttp.responseText);
535             data = obj;
536
537             //Creo la riga i cui inserire i campi e gli metto imposto la classe
538             var div = document.createElement( tagName: "div");
539             div.setAttribute( qualifiedName: "class", value: "single_product_thumb clearfix");
540
541             //Inserisco l'immagine del prodotto
542             var img = document.getElementById( elementId: "image");
543             img.src = "http://saminfo.ch/gestionevendita2018/" + data[0]['img'];
544
545             //Inserisco il prezzo del prodotto
546             var category = document.getElementById( elementId: "category");
547
548             //Creo il nuovo campo e lo sostituisco
549             var newCategory = document.createElement( tagName: "li");
550             newCategory.setAttribute( qualifiedName: "id", value: "category");
551             newCategory.appendChild(document.createTextNode( data: "Categoria: " + data[0]['categoria_prodotto']));
552             category.parentNode.replaceChild(newCategory, category);

```

Figura 2 JS detailsProduct per il cliente

Da questa prima parte si può notare che per inserire i dati nei campi e non fare in modo che vengano scritti più dati quando si chiudono e riaprono le pagine ma si sostituisca il contenuto già esistente prendo il campo e lo rimpiazzo con uno nuovo creato da 0 con il contenuto da inserire (righe 546-552).

```

555     var name = document.getElementById( elementId: "title");
556
557     //Creo il nuovo campo e lo sostituisco
558     var newName = document.createElement( tagName: "li");
559     newName.setAttribute( qualifiedName: "id", value: "category");
560     newName.appendChild(document.createTextNode( data: "Prodotto: " + data[0]['nome_prodotto']));
561     name.parentNode.replaceChild(newName, name);
562
563     //Se ci sono più di un negozio
564     if(data.length > 1) {
565         //Passo tutti i dati
566         for (var i = data.length; i > 0; i--) {
567             //Inserisco i nomi dei negozi che vendono il prodotto
568             var shop = document.createElement( tagName: "li");
569             shop.appendChild(document.createTextNode( data: "Negozio " + i + ": " + data[i - 1]['nome_negozio']));
570
571             //Prendo la lista e inserisco il nuovo elemento prima del nome del prodotto
572             var list = document.getElementById( elementId: "detailsList");
573             list.insertBefore(shop, list.childNodes[1]);
574             list.insertBefore(document.createElement( tagName: "br"), list.childNodes[0]);
575         }
576     }else{
577         //Inserisco i nomi dei negozi che vendono il prodotto
578         var shop = document.createElement( tagName: "li");
579         shop.appendChild(document.createTextNode( data: "Negozio: " + data[0]['nome_negozio']));
580
581         //Prendo la lista e inserisco il nuovo elemento prima del nome del prodotto
582         var list = document.getElementById( elementId: "detailsList");
583         list.insertBefore(shop, list.childNodes[1]);
584         list.insertBefore(document.createElement( tagName: "br"), list.childNodes[0]);
585     }

```

Figura 3 JS detailsProduct per il cliente

In questa seconda parte viene mostrato l'inserimento dei dati dei/l negozi/negozio.

Prima controllo se sono più di uno, perché in caso non lo sia inserisco solo quello esistente, se è così invece con un ciclo for passo tutti i dati e concateno uno sopra l'altro tutti i negozi esistenti.

```

588     var price = document.getElementById( elementId: "price");
589
590     //Creo il nuovo campo e elo sostituisco
591     var newPrice = document.createElement( tagName: "li");
592     newPrice.setAttribute( qualifiedName: "id", value: "price");
593     newPrice.appendChild(document.createTextNode( data: "Prezzo: "+ data[0]['prezzo_prodotto'] +" Fr."));
594     price.parentNode.replaceChild(newPrice, price);
595
596     //Inserisco il prezzo del prodotto
597     var quantity = document.getElementById( elementId: "quantity");
598
599     //Creo il nuovo campo e elo sostituisco
600     var newQuantity = document.createElement( tagName: "li");
601     newQuantity.setAttribute( qualifiedName: "id", value: "quantity");
602     newQuantity.appendChild(document.createTextNode( data: "Quantità: "+ data[0]['quantita_prodotto'] +" pezzi"));
603     quantity.parentNode.replaceChild(newQuantity, quantity);
604 }
605
606 xhttp.open( method: "POST", url: "/gestionevendita2018/product/getProduct", async: true);
607 xhttp.setRequestHeader( name: "Content-Type", value: "application/x-www-form-urlencoded");
608 xhttp.send( body: "&name=" + prodKey[0] + "&price=" + prodKey[1] + "&quantity=" + prodKey[2]);
609
610 document.getElementById( elementId: 'detailsModal').style.display = "block";
611 }

```

*Figura 4 JS detailsProduct per il cliente*

Nell'ultima parte vengono inseriti gli ultimi dati.

Come si può notare vado a prendere i dati dalla funzione "getProduct" già usata precedentemente per altri campi, in modo che mi prende anche i dati del negozio, avrei potuto fare una nuova funzione che prende solo i dati che utilizzo ma avendola già pronta, anche se prende più dati di quelli utili, non mi conveniva ricrearla da 0.

Oggi sono riuscito a completare solo questa parte, mi mancano ancora un paio di cose per la parte dei clienti dopodiché potrò lavorare all'amministratore, per lo meno per quanto riguarda la creazione dei venditori e i relativi negozi.

#### Problemi riscontrati e soluzioni adottate

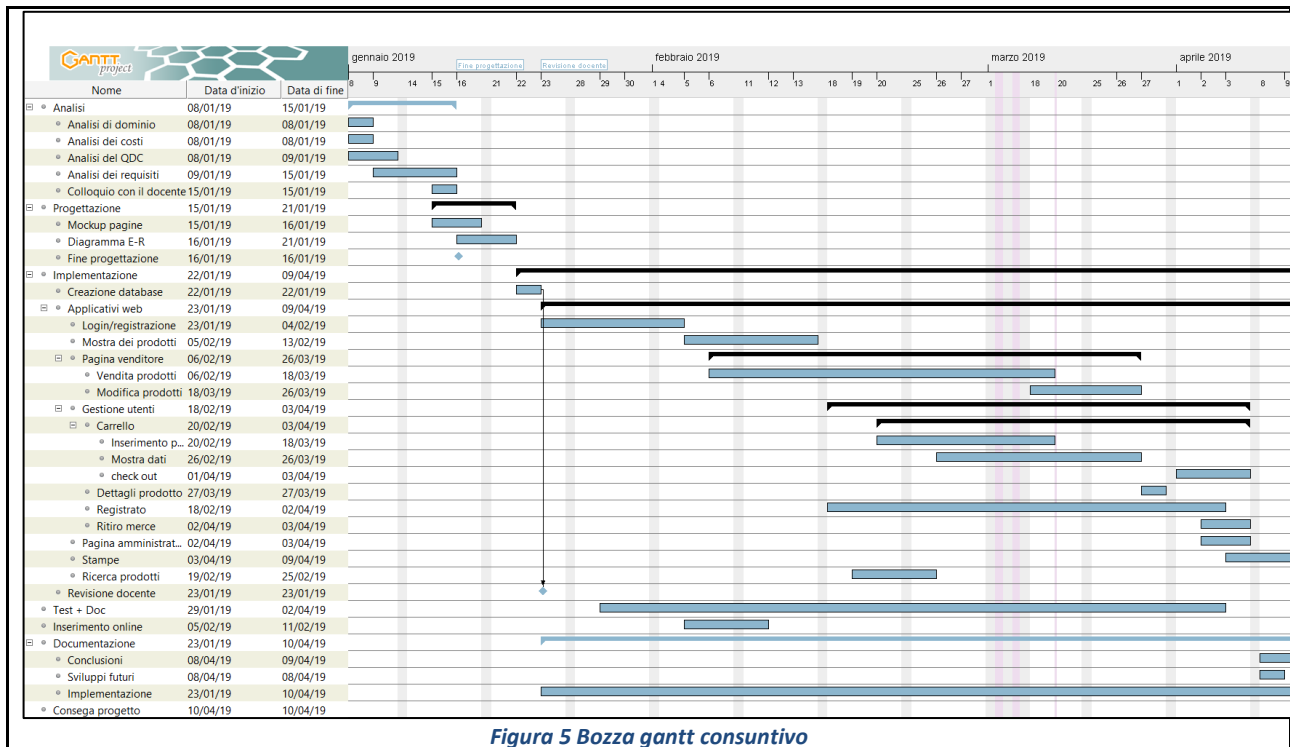
Non ho avuto particolari problemi.

#### Punto della situazione rispetto alla pianificazione

Devo completare la pagina dei clienti.

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

L'immagine sotto mostra una bozza iniziale del gantt consuntivo.



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

Completare pagina dei clienti, implementando la scelta del luogo di ritiro e la stampa dei dettagli finali.