

# Diario di lavoro

Luogo	SAMT
Data	26.02.2019

## Lavori svolti

La prima cosa che ho fatto questa mattina è stata modificare le funzioni della tabella "compra", per renderle più efficienti e ottimizzarle, in modo che sia anche più facile lavorarci. La funzione che ho modificato è stata quella che prende i dati in base al prodotto, innanzitutto gli mando la mail dell'utente tramite argomento, ma non obbligatoria, in modo che se dovesse servire prendere i dati solo dai prodotti sia possibile, dopodiché ho cambiato il modo in cui prende i prodotti, ovvero non fa un altro riferimento al database ma ad una funzione già esistente e poi ne controlla i dati.

```

144 public function getDataByproduct($prodName, $prodPrice, $prodQuantity, $custMail = null){
145
146     //Se richiesto prendo i dati dell'utente
147     $products = array();
148     if($custMail != null)
149         $products = $this->getDataByMail($custMail);
150     else
151         $products = $this->getData();
152
153     $dataArray = array();
154
155     //Se ci sono dei valori
156     if(array_key_exists( key: 0,$products) && is_array($products[0])) {
157
158         // Ciclo tutti i valori
159         foreach ($products as $value) {
160             //Se passa il prodotto richiesto
161             if ($value['nome_prodotto'] == $prodName && $value['prezzo_prodotto'] == $prodPrice
162                 && $value['quantita_prodotto'] == $prodQuantity) {
163
164                 //inserisco il valore nell'array
165                 array_push( &array: $dataArray, $value);
166             }
167         }
168         //Se c'è un solo valore lo inserisco
169     }else if(array_key_exists( key: 0,$products) && !is_array($products[0])) {
170         //Se passa il prodotto richiesto
171         if ($products['nome_prodotto'] == $prodName && $products['prezzo_prodotto'] == $prodPrice
172             && $products['quantita_prodotto'] == $prodQuantity) {
173
174             $dataArray = $products;
175         }
176     }
177     $conn = null;
178     return $dataArray;
179 }

```

**Figura 1** model `getDataByProduct` per la tabella "compra"

La funzione, come mostrato, prende i dati con in base alla mail o meno, a dipendenza del campo passato, e dopodiché se esistono già dei dati controlla quanti sono, se è il primo allora controllo se è del prodotto richiesto e passo o meno il valore, se sono più di uno li controllo e passo solo quelli del prodotto richiesto, se non ce ne sono passo un array vuoto. Un ulteriore modifica che ho fatto è stata nella funzione che inserisce i dati.

```

54      //Imposto la nuova quantità del prodotto
55      require_once 'application/models/product.php';
56      $product = new ProductModel();
57      $product->setQuantity($prodName, $prodPrice, $prodQuantity, $prodQuantity - 1);

```

**Figura 2** modifica quantità prodotto

Dopo aver inserito o modificato la riga nella tabella prendo il prodotto e tramite la funzione nella foto mostrata (riga 57) modifico il valore della quantità di quel prodotto, semplicemente prendendo il valore attuale e diminuendolo di 1.

La funzione che modifica la quantità è mostrata qui sotto.

```

140      public function setQuantity($prodName, $prodPrice, $prodQuantity, $quantity){
141          //Connetto al database
142          $conn = $this->connection->sqlConnection();
143
144          //Imposto la nuova quantità
145          $sql = $conn->prepare("UPDATE prodotto SET quantita = :quantity WHERE nome LIKE :prodName
146                                AND prezzo LIKE :prodPrice
147                                AND quantita LIKE :prodQuantity");
148          $sql->bindParam(':quantity', $quantity, PDO::PARAM_INT);
149          $sql->bindParam(':prodName', $prodName, PDO::PARAM_STR);
150          $sql->bindParam(':prodPrice', $prodPrice);
151          $sql->bindParam(':prodQuantity', $prodQuantity, PDO::PARAM_INT);
152
153          //Eseguo la query
154          if(!$sql->execute()){
155              $conn = null;
156              return false;
157          }
158
159          $conn = null;
160          return true;
161      }

```

**Figura 3** model setQuantity prodotto

La modifica viene effettuata con la query mostrata (righe 145-151) e grazie ai dati passati. Fare questa funzione mi ha dato un problema con la query quindi completarla mi ha preso del tempo.

Un'altra modifica che ho dovuto fare è stata nel file JavaScript, per fare in modo che anche se e non si aggiorna la pagina cambi comunque il valore passato della chiave di un determinato prodotto.

```

325      //Diminuisco di uno il valore della quantità del prodotto nel nome
326      keys[2]--;
327      link.name = encode(keys[0], keys[1], keys[2]);

```

**Figura 4** modifica nome link prodotto

La funzione dopo aver inserito i dati e modificato la quantità del prodotto va a decrementare il valore della quantità da inserire nel nome del link, h è quello che viene codificato e decodificato e contiene i dati della chiave del prodotto.

Una volta fatto questo ho aggiunto un controllo prima di inserire i prodotti nella lista.

```

220      for (var i = 0; i < obj.length; i++) {
221          //Se c'è ancora il prodotto
222          if(obj[i]['quantita'] > 0) {

```

**Figura 5** controllo disponibilità prodotto

Dopo aver preso i dati prima di inserire il prodotto controllo che ci sia, ovvero che la quantità sia maggiore di 0 (riga 222) e solo in quel caso iniziano a creare i componenti da inserire. La modifica viene attuata in modo provvisorio, ne discuterò con il docente per quando applicarla, perché potrebbe avere più senso farla solo all'acquisto dei prodotti, perché se un utente si mette tutti i prodotti nel carrello senza mai comprarli non saranno più disponibili agli altri, ma se vengono cambiati solo all'acquisto c'è il rischio che 2 utenti comprino l'ultimo prodotto nello stesso momento e uno dei 2 avrà un errore. Una volta completato questo ho iniziato a lavorare per fare in modo che vengano inseriti nella mostra del carrello i prodotti aggiunti. Per farlo ho utilizzato JavaScript e dopo aver inserito i dati nel database prendo i dati dal carrello e "codifico" i valori con la funzione apposita nel formato mostrato nello scorso diario.

```

343 function getCart(){
344     xhttp.onreadystatechange = function () {
345         if (this.readyState === 4 && this.status === 200) {
346
347             //Prendo i valori passati dal server e li metto in un array
348             var obj = JSON.parse(xhttp.responseText);
349
350             if(typeof (obj[0]) == "object"){
351                 for(var i = 0; i < obj.length; i++){
352                     var codedKeys = encode(obj[i]['nome_prodotto'], obj[i]['prezzo_prodotto'], obj[i]['quantita_prodotto']);
353
354                     //richiamo la funzione che inserisce i prodotti
355                     modifyCart(codedKeys);
356                 }
357             }else{
358                 var codedKeys = encode(obj['nome_prodotto'], obj['prezzo_prodotto'], obj['quantita_prodotto']);
359
360                 //richiamo la funzione che inserisce i prodotti
361                 modifyCart(codedKeys);
362             }
363         }
364     }
365     xhttp.open( method: "POST", url: "/gestionevendita2018/customer/getCart", async: true);
366     xhttp.send();
367 }

```

*Figura 6 JS per prendere i valori del carrello*

Dopo aver preso i valori controllo se ce ne sono più di uno o uno per prendere i valori nel modo corretto, e poi richiamo un'altra funzione che va a modificare i valori della mostra del carrello. L'ultima funzione non è ancora completata quindi non ho fatto una foto, la farò domani e che completerò la mostra dei prodotti nel carrello nella pagina.

#### Problemi riscontrati e soluzioni adottate

Un errore che ho avuto è stato quando cambiavo la quantità del prodotto dopo averlo messo nel carrello, perché dovevo impostare le Foreign keys delle tabelle "vende" e "compra" legate alla tabella "prodotto" le impostazioni degli "on update", impostandole come "cascade", e "on delete", mettendo "no action".

#### Punto della situazione rispetto alla pianificazione

Ho portato avanti il carrello ma non è ancora completo. Devo completare in modo ottimale la parte d'inserimento dei prodotti, ovvero inserendo anche quale negozio e venditore lo imposta.

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

Completare l'implementazione del carrello.