PROJECT-2

Design and implement an application using JSF and CDI beans to sell the products defined in the table below. The customer defines the quantities of each of the items specified in the table she will purchase. The number of items is the only input field in the table. The total item price, the number of units in stock, and the total order price should be updated as soon as the field that defines the number of units to purchase loses focus.

Obviously, the number of units you have in stock decrease as people purchase your products. Assume you never restock, and make sure you keep up with the current number of each item in your inventory. You never let that number become negative. If the customer defines a number of items to purchase greater than the number of items in stock, upon loosing focus in the number of units to purchase field, give the customer an error message which informs her the maximum number of items she is allowed to purchase. Also, once the purchase order is submitted, and if some products have reached a level of 0 units in the inventory, you remove the rows associated with those products from the product table.

Given that the table has many rows, the table should use pagination. There should be 5 rows per page, and the last page should have 5 or less rows. The user should be able to sort the rows of this table based on each of its columns both ascending and descending. Also, the user should be able to filter items in the table in terms of greater or less than a defined price or in terms of the names of a product using the "*" as a wildcard. For example, "green*". That would output all products whose name start with "green".

You will not use a database. You can initially read the table product details from a file to start your selling session. Once you have read from the file initially, you may not use the file or any other form of persistent storage. Like Project-1, you will just use RAM memory for all your transactions.

Make sure that your application works with more than one user, preferably with at least four or more users simultaneously buying products from your site. Each user should see the correct number of items in the column associated with the number of units that are left in stock. Of course, until any user finalizes the purchase transaction the other users do not see the decrease of units purchased by that user.

If a given buyer is in the middle of defining a purchase order, and another buyer just finished ordering all of the remaining units of a product that the first buyer is about to order, then your application should give that buyer an error message, as well as update the product table she sees removing the rows that have no stock.

In general, after the buyer successfully submits an order, the next page rendered should be a table with all the items that were purchased, with the same columns as the product details table. Also, the total order price should be included with the same presentation format as defined in the product details table.

The default number for the quantities purchased of each item is 0 before the user inputs any quantity.

Conversiones, Validaciones y Mensajes de Notificaciones de Errores Generales

- Si el usuario no define números si no por ejemplo letras en el campo de cantidades de ítems, la aplicación genera un error, devolviendo la misma página con la entrada que genero el error, y mandando un mensaje explicando cual fue el error y como corregirlo.
- Si el usuario pone cantidades de ítems que son más grandes que las cantidades que están en el inventario, la aplicación genera un error devolviendo la página con la entrada que genero el error, y mandando un mensaje explicando cual fue el error y como corregirlo.
- Si el usuario pone cantidades negativas como numero de ítems para comprar, la aplicación genera un error devolviendo la página con la entrada que genero el error, y mandando un mensaje explicando cual fue el error y como corregirlo.

Table of Products

Number of Units to Purchase	Total Item Price	Serial Number	Product Name	Price per Unit	Number of Units Currently In Stock
0		00000100	brown shirts	$34.43	300
0		10000011	green shorts	$32.41	231
0		02234344	shaving cream	$14.23	443
0		12341233	green beer	$4.43	29
0		12100222	black shoes	$67.41	94
0		12231222	coffee	$10.23	99
0		10001222	black combs	$1.43	23
0		14311222	printer red ink	$43.43	745
0		02111222	scissors	$24.55	12
0		13111222	red tape	$4.03	45
0		14111222	black pens	$2.43	32
0		15111222	red pens	$1.43	134
0		16111222	pencils	$0.43	231
0		17111222	hard drives	$50.54	222
0		18111222	sofas	$400.43	3
0		19111222	chairs	$314.43	32
0		12001222	mp3 players	$20.43	67
0		12111222	white shoes	$44.53	62
0		12121222	pen drives	$19.43	33
0		12131222	tooth paste	$4.43	54
0		12141222	dental floss	$4.53	12
0		12151222	hair brushes	$36.43	44
0		12161222	alarm clocks	$9.43	56
0		12171222	watches	$87.43	11
0		12181222	tires	$78.98	78
0		12191222	car oil	$34.32	21
0		12111223	cooking oil	$19.88	225
0		12111224	note books	$6.55	21
0		12111225	vitamin A	$18.99	4
0		12111222	vitamin C	$21.87	55
0		12119222	vitamin B	$17.83	3
0		12111222	vitamin d	$12.32	12
0		12118222	zinc	$20.56	56
0		12711222	coke	$1.23	11
0		12111292	pepsi	$1.23	76
0		12511222	scotch	$60.43	33
0		12311222	bourbon	$40.44	10
0		12191222	rum	$32.43	30
0		12181222	dark beer	$10.43	67
0		12171222	light beer	$7.43	22
0		12161222	hats	$21.44	12
0		12151222	capes	$34.78	78
0		12131222	umbrellas	$4.21	22

The total order price is: $0.00