

1. Design and Functional Specification

ในส่วนนี้จะป็นรายละเอียดการออกแบบ Component และ Function บางส่วนของโครงการ โดยมีเป้าหมายที่จะพัฒนาระบบเฉพาะส่วนที่ตอบโจทยการเพิ่มรายการสินค้าที่ต้องการเลือกซื้อลงในรถเข็น (ในที่นี้คือ addProductToCart()) แต่ไม่รวมถึงการลดจำนวนสินค้าหรือการยืนยันรายการสินค้า ทั้งนี้มี 2 Component ที่เกี่ยวข้อง คือ Product Manager และ Shopping ซึ่งมีรายละเอียดดังต่อไปนี้

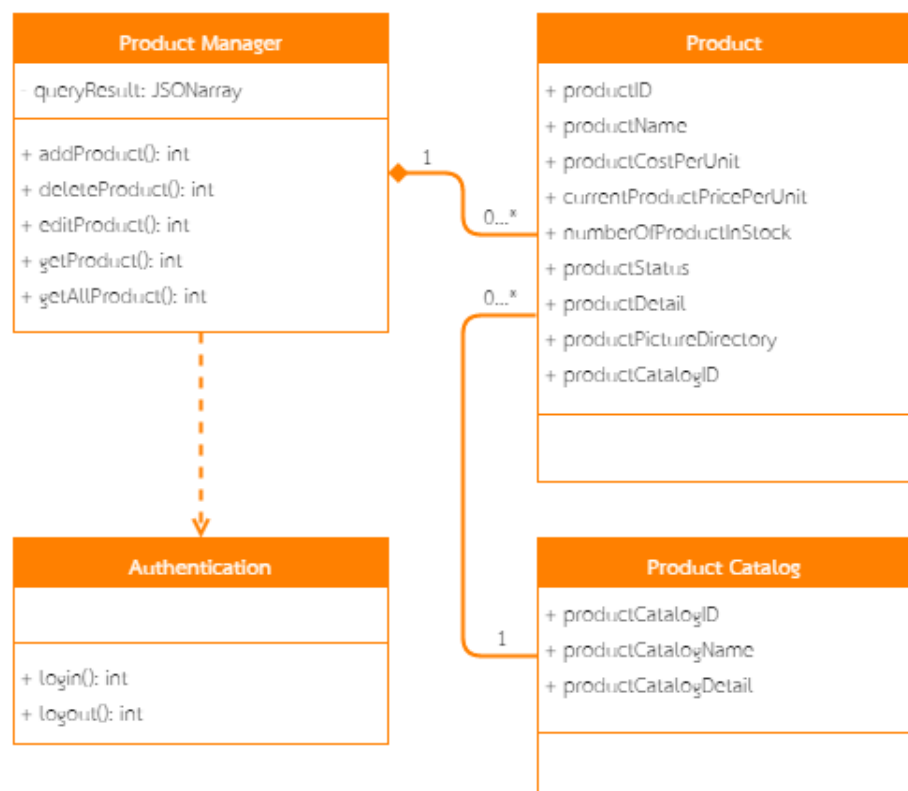
1) Component Product Manager

1.1) Process Narrative for Component Product Manager

Component Product Manager เหมือนกับคลังสินค้าที่เก็บรายละเอียดสินค้าทุกชนิดในร้านค้า ซึ่งจะถูกเก็บไว้ในฐานข้อมูล โดยจะต้องมีการเชื่อมต่อกับฐานข้อมูลเพื่อดึงข้อมูลสินค้ามาแสดงให้กับเจ้าของร้าน การเพิ่มสินค้า และการแก้ไขข้อมูลสินค้าผ่านทาง GUI บน Web Application

1.2) Process Detail for Component Product Manager

1.2.1) Design Class Hierarchy for Component Product Manager



ภาพที่ 1 Class Hierarchy for Component Product Manager

1.2.2) Restrictions/Limitations for Component Product Manager

- มีเพียงบัญชีผู้ใช้เจ้าของร้านเท่านั้นที่สามารถจัดการกับข้อมูลภายใน Component นี้ ซึ่งผู้ใช้อื่นๆ จะสามารถทำได้เพียงดูข้อมูลของ Component เท่านั้น

1.2.3) Performance Issues for Component Product Manager

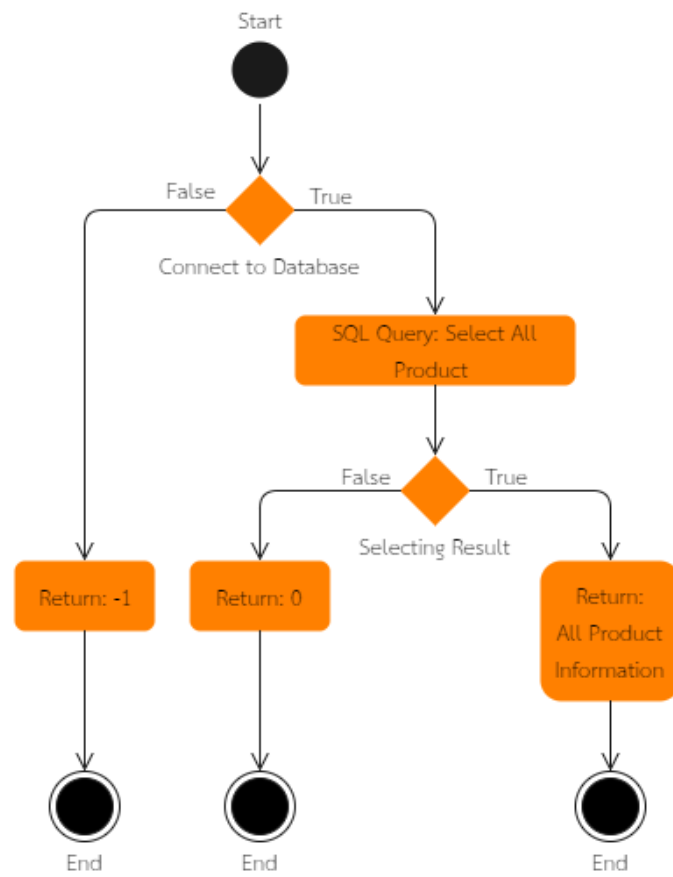
- การตอบสนองของทุกๆ คำสั่งบน GUI ควรทำได้ภายในเวลา 3 วินาที

1.2.4) Design Constraints for Component Product Manager

- เชื่อมต่อกับระบบฐานข้อมูล mysql ด้วย API ของภาษา PHP ดังนั้นชนิดของข้อมูลในฐานข้อมูลกับชนิดข้อมูลในภาษา PHP จะต้องสอดคล้องกันในการรับส่งหรือ Cast ข้อมูล
- การรับส่งข้อมูลระหว่าง Client Site กับ Server Site จะอยู่ในรูปแบบของ JSON

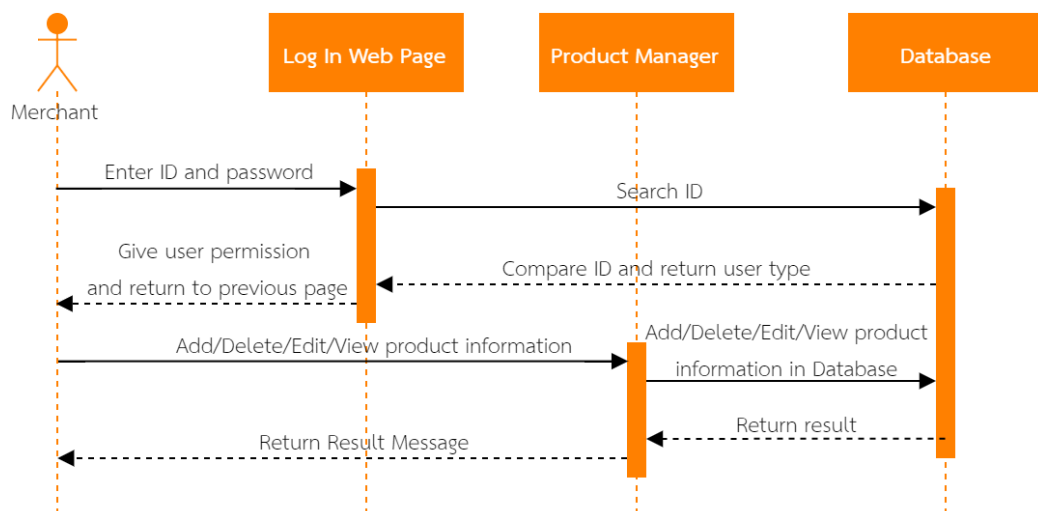
1.2.5) Processing Detail (Processing Narrative and Algorithmic Model) for Each Operation of Component Product Manager

- **addProduct** - ฟังก์ชันในการเพิ่ม Product ใหม่ลงในฐานข้อมูล
- **deleteProduct** - ฟังก์ชันสำหรับลบ Product ที่ต้องการออกจากฐานข้อมูล โดยในการใช้งานจะต้องระบุ ProductID ของ Product ที่ต้องการลบ
- **editProduct** - ฟังก์ชันในการแก้ไขค่าใน Attribute ซึ่งเป็นข้อมูลของ Product ทั้งนี้จะต้องระบุ Attribute ที่ต้องการแก้ไขและค่าใหม่ที่จะแทนที่
- **getProduct** - ฟังก์ชันในการดึงข้อมูล Product จากระบบฐานข้อมูล โดยในการใช้งานจะต้องระบุ ProductID ของ Product นั้น
- **getAllProduct** - ฟังก์ชันในการดึงข้อมูล Product ทั้งหมดจากฐานข้อมูล
(หมายเหตุ: เป็น Support Function ที่จะถูก Implement ในงานนี้)



ภาพที่ 2 Get All Product

1.3) Dynamic Behavior for Component Product Manager



ภาพที่ 3 Add / Edit / Delete / View Product

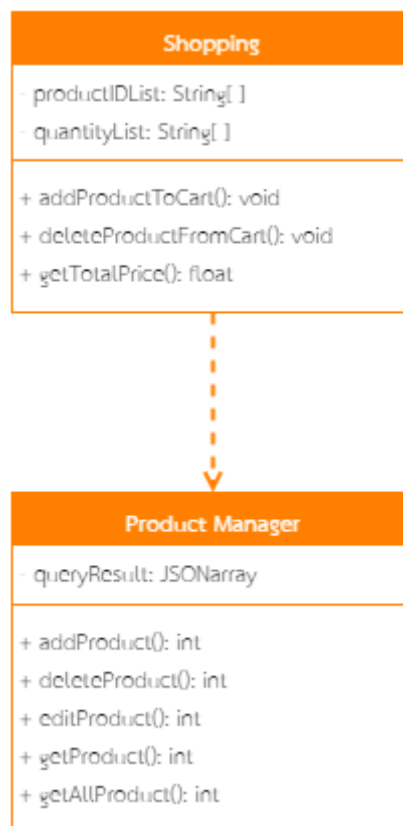
2) Component Shopping

2.1) Process Narrative for Component Shopping

Component Shopping มีหน้าที่ในการบริการหรือช่องทางในการเลือกซื้อสินค้าบน Web Application สำหรับลูกค้า โดยจะดึงข้อมูลสินค้าจากฐานข้อมูลมาแสดง และมีรถเข็น (Cart) ให้กับลูกค้าเพื่อเลือกซื้อสินค้าได้คราวละหลายๆ ชิ้นต่อการสั่งซื้อ โดยเก็บข้อมูลรายการและจำนวนสินค้าอยู่บน Client Site ตลอดระยะเวลาในการทำกิจกรรมต่างๆ บน Web Application จนกว่าลูกค้าจะยืนยันการสั่งซื้อรายการสินค้านั้นๆ แล้วทำการเพิ่มการสั่งซื้อนี้เข้าสู่ฐานข้อมูล

2.2) Process Detail for Component Shopping

2.2.1) Design Class Hierarchy for Component Shopping



ภาพที่ 4 Class Hierarchy for Component Shopping

2.2.2) Restrictions/Limitations for Component Shopping

- Component จะถูกเรียกใช้ตั้งแต่ลูกค้าเข้าสู่ Web Application และทำการเลือกสินค้าในหน้าแสดงรายการสินค้าที่เปิดขาย ดังนั้นจึงถูกใช้งานกับเฉพาะผู้ใช้ทั่วไปที่ไม่มีบัญชี (ผู้เข้าชม Website) หรือผู้ใช้ที่มีบัญชีเป็นลูกค้าเท่านั้น

2.2.3) Performance Issues for Component Shopping

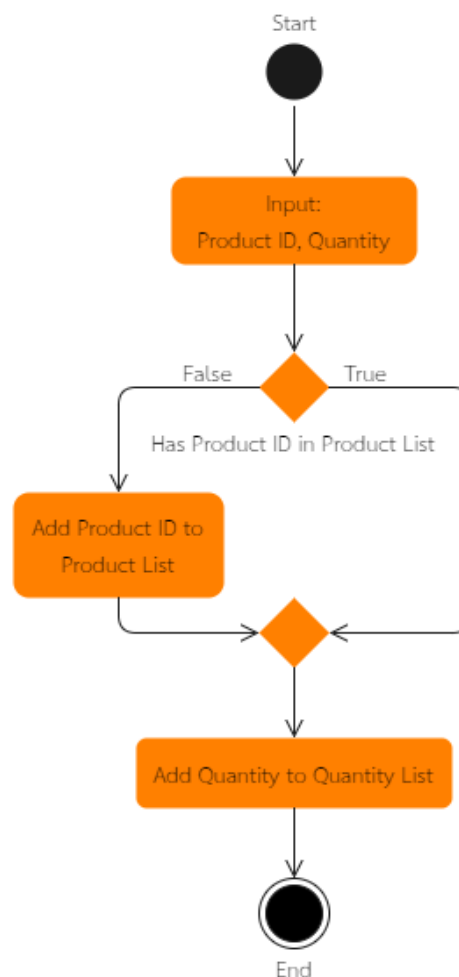
- การตอบสนองของทุกๆ คำสั่งบน GUI ควรทำได้ภายในเวลา 3 วินาที

2.2.4) Design Constraints for Component Shopping

- เนื่องจากจะต้องเก็บข้อมูลรถเข็น (Cart) จากการเลือกซื้อสินค้าของลูกค้าเพื่อใช้ในการสร้าง Order ต่อไป ดังนั้นจึงต้องมีการสร้าง Shopping Cart Session โดยใช้ภาษา PHP
- เชื่อมต่อกับระบบฐานข้อมูล mysql ด้วย API ของภาษา PHP ดังนั้นชนิดของข้อมูลในฐานข้อมูลกับชนิดข้อมูลในภาษา PHP จะต้องสอดคล้องกันในการรับส่งหรือ Cast ข้อมูล
- การรับส่งข้อมูลระหว่าง Client Site กับ Server Site จะอยู่ในรูปแบบของ JSON

2.2.5) Processing Detail (Processing Narrative and Algorithmic Model) for Each Operation of Component Shopping

- **addProductToCart** - ฟังก์ชันในการเพิ่ม Product ลงในรถเข็นสินค้า (หมายเหตุ: เป็นฟังก์ชันหลักที่จะ Implement ในงานนี้)



ภาพที่ 5 Add Product To Cart

- **deleteProductFromCart** - ฟังก์ชันในการลบ Product ออกจากรถเข็นสินค้า โดยในการใช้งานจะต้องระบุ ProductID ของ Product ที่ต้องการลบ

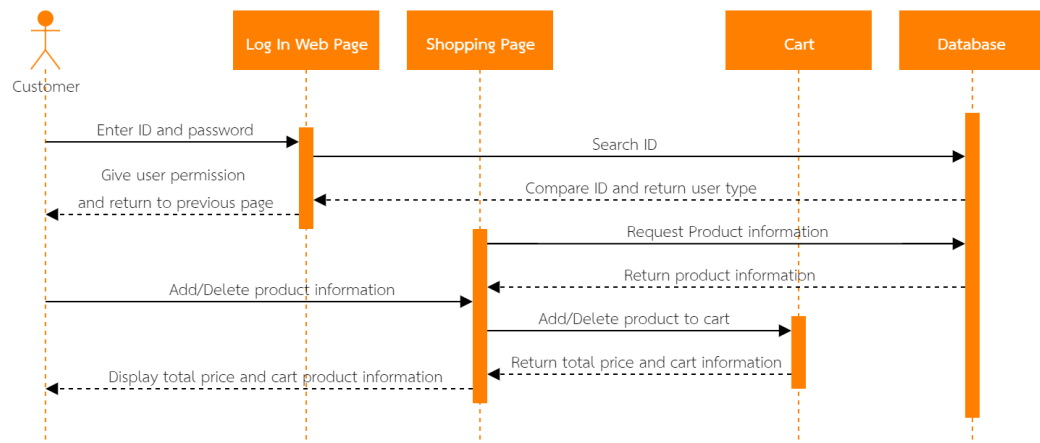
- getTotalPrice – ฟังก์ชันคำนวณหาราคารวมของสินค้าในรถเข็น

2.3) Test Points List and Description for Component Shopping

Test ID	Test Description	Test Data	Pre-Condition	Expected Result	Actual Result	Status
1	Customer ทำการเพิ่มสินค้าลงในรถเข็น	สินค้าที่เลือก และจำนวนสินค้า	-	มีรายการสินค้าเพิ่มขึ้นในรถเข็น และราคารวมเปลี่ยนแปลงไป		
2	Customer ทำการลบสินค้าลงในรถเข็น	สินค้าที่เลือก	-	มีรายการสินค้าลดลงในรถเข็น และราคารวมเปลี่ยนแปลงไป		

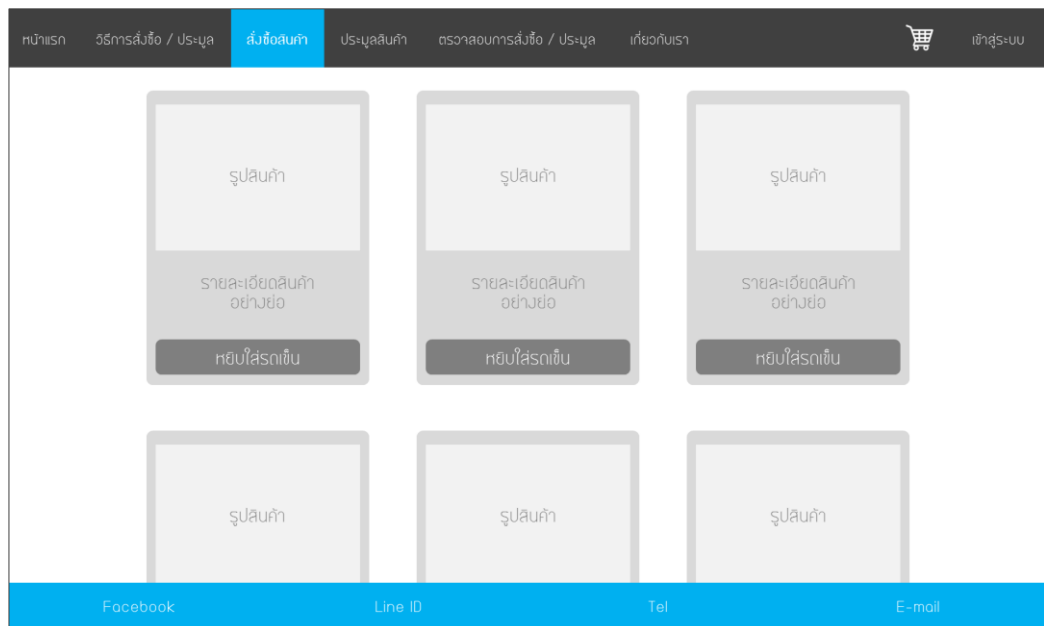
ตารางที่ 1 Test Points List for Component Shopping

2.4) Dynamic Behavior for Component Shopping



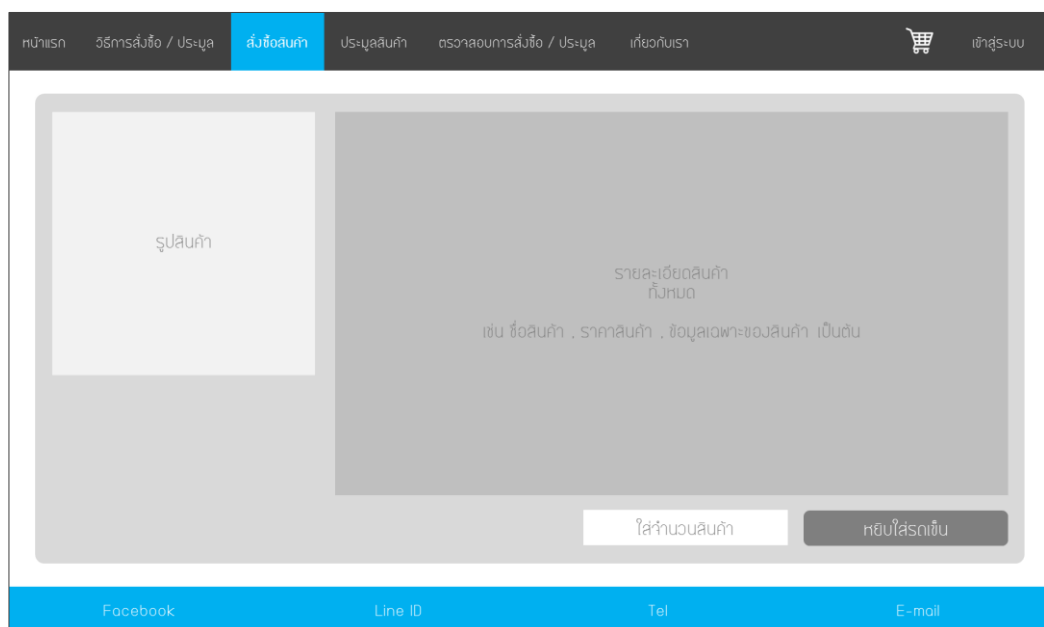
ภาพที่ 5 Add / Delete Product in Cart

2.5) Interfaces for Component Shopping



ภาพที่ 6 Product List

เป็นหน้าแสดงสินค้าทั้งหมดที่มีอยู่ในร้าน ทุกหมวด ทุกโปรโมชั่น โดยแสดงข้อมูลสินค้าแบบย่อ โดยมีรูปสินค้ากับราคา สามารถคลิกที่ขอบของแต่ละสินค้าเพื่อเข้าไปหน้าของสินค้าแต่ละตัวได้ (หมายเหตุ: เป็น GUI หลักที่จะ Implement ในงานนี้)



ภาพที่ 7 Product

เป็นหน้าแสดงสินค้าแต่ละตัว โดยมีรายละเอียดทั้งหมดของสินค้า มีการแสดงรูปของสินค้า ทางซ้ายมือของหน้าต่าง UI และสามารถแก้ไขจำนวนสินค้าที่ต้องการสั่งในช่องในช่อง

จำนวนสินค้า และมีปุ่มหยิบใส่รถเข็นเพื่อเป็นการสั่งซื้อสินค้านั้น ตามจำนวนที่กรอกไว้ในช่องใส่จำนวนสินค้า

หน้าแรก


วิธีการสั่งซื้อ / ประมวล

สั่งซื้อสินค้า

ประมวลสินค้า

ตรวจสอบการสั่งซื้อ / ประมวล

เกี่ยวกับเรา



เข้าสู่ระบบ

เลขที่ใบสั่งซื้อ

รายการที่	รายการสินค้า	จำนวน	ราคาต่อหน่วย	รวม	ลบสินค้า
รวมราคาสุทธิ					

ยืนยันการสั่งซื้อ

Facebook

Line ID

Tel

E-mail

ภาพที่ 8 Confirm Order

เป็นหน้ายืนยันข้อมูลการสั่งซื้อสินค้าในรถเข็นของลูกค้า โดยมีข้อมูลของสินค้าที่อยู่ในรถเข็น ทั้งชื่อ รหัส จำนวน ราคา และราคารวมสุทธิ และมีปุ่มสำหรับให้ลูกค้า ลบสินค้าที่ไม่ต้องการออกได้ และในช่องจำนวน สามารถแก้ไขจำนวนสินค้าที่ต้องการสั่งได้สินค้า

2. Source Code with Code Comments

1) ShoppingAndBiddingWebApplication.sql

```
25  --
26  -- Table structure for table `product`
27  --
28
29  CREATE TABLE IF NOT EXISTS `product` (
30    `Product ID` int(11) NOT NULL,
31    `Product Name` varchar(64) COLLATE utf8_unicode_ci NOT NULL,
32    `Product Cost Per Unit` int(64) NOT NULL,
33    `Current Product Price Per Unit` int(64) NOT NULL,
34    `Number Of Product In Stock` int(64) NOT NULL,
35    `Product Status` varchar(64) COLLATE utf8_unicode_ci NOT NULL,
36    `Product Detail` text COLLATE utf8_unicode_ci NOT NULL,
37    `Product Picture Directory` varchar(64) COLLATE utf8_unicode_ci NOT NULL,
38    `Product Catalog ID` int(64) NOT NULL
39  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
40
41  --
42  -- Indexes for table `product`
43  --
44  ALTER TABLE `product`
45  ADD PRIMARY KEY (`Product ID`);
46
```

ภาพที่ 9 ใช้สำหรับสร้างตาราง Product ใน Database โดยมี Attribute และ Primary Key ตาม Source Code ข้างต้น

2) Connent.php

```
1  <?php
2      header("Access-Control-Allow-Origin: *");
3      header("Content-Type: application/json; charset=UTF8");
4
5      // Parameter for access to database
6      $host = "localhost";
7      $username = "root";
8      $password = "tuntuntun";
9      $databaseName = "ShoppingAndBiddingWebApplication";
10 ?>
```

ภาพที่ 10 ใช้สำหรับกำหนดค่า Parameter ต่างๆ ที่จำเป็นต่อการเปิดการเชื่อมต่อกับฐานข้อมูล

3) GetAllProduct.php

```
1 <?php
2     include("Connect.php");
3
4     // Connect to database
5     $con = @mysqli_connect($host, $username, $password, $databaseName);
6
7     if (mysqli_connect_errno())
8     {
9         echo "-1"; // Cannot connect to database
10    }
11    else
12    {
13        // Successfully connect to database and ready to query all product
14        $sql = "SELECT * FROM product";
15        $result = @mysqli_query($con, $sql);
16        if ($result)
17        {
18            // Successfully query all product and ready to fetch result and form to JSON array
19            $out = "[";
20            while($rs = $result->fetch_array(MYSQLI_ASSOC))
21            {
22                if ($out != "[")
23                {
24                    $out .= ",";
25                }
26                $out .= '{"Product ID":"' . $rs["Product ID"] . "',';
27                $out .= "Product Name":"' . $rs["Product Name"] . "',';
28                $out .= "Product Cost Per Unit":"' . $rs["Product Cost Per Unit"] . "',';
29                $out .= "Current Product Price Per Unit":"' . $rs["Current Product Price Per Unit"] . "',';
30                $out .= "Number Of Product In Stock":"' . $rs["Number Of Product In Stock"] . "',';
31                $out .= "Product Status":"' . $rs["Product Status"] . "',';
32                $out .= "Product Detail":"' . $rs["Product Detail"] . "',';
33                $out .= "Product Picture Directory":"' . $rs["Product Picture Directory"] . "',';
34                $out .= "Product Catalog ID":"' . $rs["Product Catalog ID"] . "'}';
35            }
36            $out .= "]";
37            echo $out; // Show JAON array result
38        }
39        else
40        {
41            echo "0"; // Cannot query
42        }
43        mysqli_close($con); // Close connention
44    }
45    ?>
```

ภาพที่ 11 ใช้สำหรับติดต่อกับฐานข้อมูลตาราง Product เพื่อดึงข้อมูลสินค้ามาแสดงโดยจัดให้อยู่ในรูปแบบ JSON Array

4) Shop.html

```
1 <!DOCTYPE html>
2 <html>
3     <head>
4         <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
5         <meta name="viewport" content="width=device-width, initial-scale=1.0">
6         <title>Shopping</title>
7         <link rel="stylesheet" type="text/css" href="CSS.css">
8     </head>
```

ภาพที่ 12 Header Tag ใน html

```
9      <body>
10      <!-- Menu GUI -->
11      <div class="menu">
12      <!-- Menu GUI for Device Width <= 876px -->
13      <div class="menubarForMaxWidth876">
14      <ul>
15      <li><a id="clickMenubarDropDown" onclick="showHideDropDown('menubarDropDown')">
16      
17      </a></li>
18
19      <li style="float: right;"><a href="#">เข้าสู่ระบบ</a></li>
20      <li style="float: right;"><a id="clickCartDropDown1" onclick="showHideDropDown('cartDropDown')">
21      
22      <div id="total1" class="total"></div>
23      รายการสั่งซื้อ
24      </a></li>
25      </ul>
26      </div>
27      <!-- Menu Dropdown GUI for Device Width <= 876px -->
28      <div id="menubarDropDown" class="menubarDropDown" style="visibility: hidden;">
29      <li><a href="#">หน้าแรก</a></li>
30      <li><a href="#">วิธีการสั่งซื้อ</a></li>
31      <li><a class="active" href="Shop.html">สั่งซื้อสินค้า</a></li>
32      <li><a href="#">ตรวจสอบการสั่งซื้อ</a></li>
33      <li><a href="#">เกี่ยวกับเรา</a></li>
34      </div>
```

ภาพที่ 13 GUI ส่วนแสดงผล Menubar

```
36      <!-- Menu GUI for Device Width >= 877px -->
37      <div class="menubarForMinWidth877">
38      <ul>
39      <li><a href="#">หน้าแรก</a></li>
40      <li><a href="#">วิธีการสั่งซื้อ</a></li>
41      <li><a class="active" href="Shop.html">สั่งซื้อสินค้า</a></li>
42      <li><a href="#">ตรวจสอบการสั่งซื้อ</a></li>
43      <li><a href="#">เกี่ยวกับเรา</a></li>
44
45      <li style="float: right;"><a href="#">เข้าสู่ระบบ</a></li>
46      <li style="float: right;"><a id="clickCartDropDown2" onclick="showHideDropDown('cartDropDown')">
47      
48      <div id="total2" class="total"></div>
49      รายการสั่งซื้อ
50      </a></li>
51      </ul>
52      </div>
53
54      <!-- Cart Dropdown GUI -->
55      <div id="cartDropDown" class="cartDropDown" style="visibility: hidden;">
56      <li><a id="clickShowCart" style="width: 180px;" onclick="showCartList()">ดูรายการสั่งซื้อ</a></li>
57      <li><a id="clickClearCart" style="width: 180px;" onclick="clearCookie()">ล้างรถเข็น</a></li>
58      </div>
59      </div>
```

ภาพที่ 14 GUI ส่วนแสดงผล Menubar

```
61      <!-- Content GUI : Show Product List -->
62      <div class="content">
63      <center>
64      <br><br><br><br>
65      <div id="ProductListTemplate"></div>
66      <br><br><br>
67      </center>
68      </div>
```

ภาพที่ 15 GUI ส่วนแสดงผล Content (Product List)

```
70      <!-- Footer GUI -->
71      <div class="footer">
72      <center>
73      <div class="contact">
74          
75          <br>Facebook
76      </div>
77      <div class="contact">
78          
79          <br>Line ID
80      </div>
81      <div class="contact">
82          
83          <br>(+66) XX-XXX-XXXX
84      </div>
85      <div class="contact">
86          
87          <br>E-mail
88      </div>
89      </center>
90  </div>
```

ภาพที่ 16 GUI ส่วนแสดงผล Footer

```
92      <!-- Part of Javascript -->
93      <script type="text/javascript">
94          loadProduct(); // Load Product List from Database by Call GetAllProduct.php Service
95          initialCookie(); // Initial Cookie for Cart Data (When Open This page)
```

ภาพที่ 17 ส่วนของ Javascript ที่จะ Run ทุกครั้งเมื่อมีการเปิดหน้า html นี้

```
97      // Load Product List from Database by Call GetAllProduct.php Service
98      function loadProduct()
99      {
100          var xmlhttp = new XMLHttpRequest();
101          var url = "GetAllProduct.php";
102          xmlhttp.onreadystatechange=function()
103          {
104              if (xmlhttp.readyState == 4 && xmlhttp.status == 200)
105              {
106                  // Map JSON Array Data to GUI
107                  displayResponseProduct(xmlhttp.responseText);
108              }
109          }
110          xmlhttp.open("GET", url, true);
111          xmlhttp.send();
112      }
```

ภาพที่ 18 Javascript Function สำหรับดึงข้อมูล Product ทั้งหมดจากฐานข้อมูล

```
114      // Map JSON Array Data to GUI
115      function displayResponseProduct(response)
116      {
117          var arr = JSON.parse(response);
118          var i=0;
119          var out = " ";
120          var name = " ";
121
122          // Loop: Map Product Data to Product Card List
123          for(i = 0 ; i < arr.length ; i++)
124          {
125              if(parseInt(arr[i]["Number Of Product In Stock"]) > 0)
126              {
127                  out += "<div class='shopList'>";
128                  out += "<center>";
129                  // Contain Product Image to Product Card
130                  out += "<img id='shopImg' + arr[i]['Product ID'] + \"'\" class='shopImg' src='\" + arr[i]['Product Picture Directory'] + \"'>";
131                  // Contain Product ID to Product Card
132                  out += "<center><div id='shopID' + arr[i]['Product ID'] + \"'\" class='shopID'>หมายเลข + arr[i]['Product ID'] + \"</div></center>";
133                  out += "<hr class='style-shop'>";
134                  // Product Name Abbreviation for containing to Product Card
135                  name = (arr[i]["Product Name"].length > 12)? ((arr[i]["Product Name"].substring(0,9)) + "...") : arr[i]["Product Name"];
136                  // Contain Product Name to Product Card
137                  out += "<span title='\" + arr[i]['Product Name'] + \"'><div id='shopName' + arr[i]['Product ID'] + \"'\" class='shopName'> + name + \"</div></span>";
138                  // Contain Product Price to Product Card
139                  out += "<div id='shopPrice' + arr[i]['Product ID'] + \"'\" class='shopPrice'> + commaFormatted(arr[i]['Current Product Price Per Unit'] + \"00\") + \" $</div>";
140                  // Button for Adding 1-Unit Product to Cart
141                  out += "<input id='shopAdd' + arr[i]['Product ID'] + \"'\" class='addtocart' type='button' value='เพิ่มสินค้า' onclick='writeCookie(\" + arr[i]['Product ID'] + \",1)'>";
142                  out += "</center>";
143                  out += "</div>";
144              }
145          }
146          document.getElementById("ProductListTemplate").innerHTML = out;
147      }
```

ภาพที่ 19 Javascript Function สำหรับนำข้อมูลที่ได้จากฐานข้อมูลมาแสดงบน GUI

```
192 // Initial Cookie for Cart Data
193 function initialCookie()
194 {
195     // Set Count (# of Type Product)
196     if(getCookie("count") == "") // Never Opened This Page
197     {
198         document.cookie = "count=0;";
199     }
200     // Set Total (Summation of Quantity Of All Product)
201     if(getCookie("total") == "") // Never Opened This Page
202     {
203         document.cookie = "total=0;";
204         document.getElementById('total1').innerHTML = "0";
205         document.getElementById('total2').innerHTML = "0";
206     }
207     else // Opened This Page
208     {
209         document.getElementById('total1').innerHTML = getCookie("total");
210         document.getElementById('total2').innerHTML = getCookie("total");
211     }
212 }
```

ภาพที่ 20 Javascript Function สำหรับกำหนดค่าเริ่มต้นของ Cookie

```
214 // ***Focus On This Function*** addProductToCart by Write Data to Cookie
215 function writeCookie(productID, quantity)
216 {
217     // Does Cookie Have This Product?
218     var found = searchProductIDInCookie(productID);
219
220     if(found == -1) // Not Found that This Product was in Cookie
221     {
222         // Add This Product as New Product in Cart (Cookie)
223         var new_count = parseInt(getCookie("count")) + 1;
224         document.cookie = "productID" + new_count + "=" + productID;
225         document.cookie = "quantity" + new_count + "=" + quantity;
226         document.cookie = "count" + "=" + new_count;
227     }
228     else // Found that This Product was in Cookie
229     {
230         // Only Update Quantity
231         var new_quantity = parseInt(getCookie("quantity"+found)) + quantity;
232         document.cookie = "quantity" + found + "=" + new_quantity;
233     }
234     // Update Total Quantity
235     var new_total = parseInt(getCookie("total")) + quantity;
236     document.cookie = "total=" + new_total;
237     document.getElementById('total1').innerHTML = new_total.toString();
238     document.getElementById('total2').innerHTML = new_total.toString();
239 }
```

ภาพที่ 21 Javascript Function สำหรับเพิ่มสินค้าลงในรถเข็น เมื่อกดปุ่ม “หยิบใส่รถเข็น”
(หมายเหตุ: เป็นฟังก์ชันหลักที่จะ Implement ในงานนี้)

```
241 // Search Product ID in Cookie
242 function searchProductIDInCookie(productID)
243 {
244     for(var i = 1 ; i <= parseInt(getCookie("count")) ; i++)
245     {
246         if(parseInt(getCookie("productID" + i)) == productID )
247         {
248             return i; // Found and Return Index
249         }
250     }
251     return -1; // Not Found and Return -1
252 }
```

ภาพที่ 22 Javascript Function สำหรับตรวจสอบว่ามีสินค้าชนิดนี้ในรถเข็นอยู่แล้วหรือไม่

```
254 // Get Cookie by Key Value
255 // Reference: http://www.w3schools.com/js/js_cookies.asp
256 function getCookie(cname)
257 {
258     var name = cname + "=";
259     var ca = document.cookie.split(';');
260     for(var i = 0 ; i < ca.length ; i++)
261     {
262         var c = ca[i];
263         while (c.charAt(0) == ' ')
264         {
265             c = c.substring(1);
266         }
267         if (c.indexOf(name) == 0)
268         {
269             return c.substring(name.length, c.length);
270         }
271     }
272     return "";
273 }
```

ภาพที่ 23 Javascript Function สำหรับดึงค่าจาก Cookie

```
275 // Show Cart Data on Popup (Alert by Javascript)
276 function showCartList()
277 {
278     var show = "Product ID (Quantity)\n";
279     for(var i = 1 ; i <= parseInt(getCookie("count")) ; i++)
280     {
281         show += getCookie("productID" + i) + " (" + getCookie("quantity" + i) + ")\n";
282     }
283     show += "\n\n";
284     alert(show);
285 }
```

หมายเหตุ:
ในกรณีนี้ Implementation และ addProductToCart Function ตั้งขึ้นจึงตรวจสอบเพื่อข้อมูลที่ถูกต้อง Add ใ้ใน Cart (Cookie)\n
โดยในการพัฒนาต้องนำข้อมูล Product ID และ Quantity (ซึ่งเก็บใน Cookie) ไปแสดงในหน้ารายการสั่งซื้อ ซึ่งสามารถ
เพิ่ม กด และยืนยันการสั่งซื้อสินค้าได้

ภาพที่ 24 Javascript Function สำหรับแสดงข้อมูลสินค้าในรถเข็น

```
287 // Clear Cookie (Clear Cart)
288 function clearCookie()
289 {
290     document.cookie = "count=0;";
291     document.cookie = "total=0;";
292     document.getElementById('total1').innerHTML = "0";
293     document.getElementById('total2').innerHTML = "0";
294 }
```

ภาพที่ 25 Javascript Function สำหรับล้างข้อมูลในรถเข็น

```
296 // Dropdown Controlling
297 function showHideDropDown(id)
298 {
299     if(document.getElementById(id).style.visibility == 'hidden')
300     {
301         document.getElementById(id).style.visibility = 'visible';
302     }
303     else
304     {
305         document.getElementById(id).style.visibility = 'hidden';
306     }
307 }
308 </script>
309 </body>
310 </html>
```

ภาพที่ 26 Javascript Function สำหรับเปิด/ปิด Dropdown

5) CSS.css

ใช้สำหรับกำหนดและออกแบบรูปแบบ Element ต่างๆ ของ GUI โดยสร้างเป็น Class ต่างๆ (สามารถดู Source Code ได้จาก CSS.css)

3. Test Cases and Results

ในขั้นตอนนี้เป็นการทดสอบ Function ที่ได้ Implement ไว้ โดยจะทดสอบผ่าน PHP Codeception ซึ่งเป็นเครื่องมือที่ช่วยให้การทดสอบสะดวกรวดเร็วขึ้นเมื่อต้องการทดสอบ Website Application ที่ต้องมีการกระทำที่หลากหลายและทำจำนวนหลายกรณี

1) Adding Products to Cart Scenarios

Test ID	Test Description	Test Data	Step	Pre-Condition	Post-Condition	Expected Result	Actual Result	Status
1	เลือกซื้อสินค้าอย่างละ 1 ชิ้น	-	1. เลือกสินค้ารหัส 1 โดยกดปุ่ม 'หยิบใส่รถเข็น' จำนวน 1 ครั้ง 2. เลือกสินค้ารหัส 3 โดยกดปุ่ม 'หยิบใส่รถเข็น' จำนวน 1 ครั้ง 3. เลือกสินค้ารหัส 5 โดยกดปุ่ม 'หยิบใส่รถเข็น' จำนวน 1 ครั้ง 4. เลือกสินค้ารหัส 7 โดยกดปุ่ม 'หยิบใส่รถเข็น' จำนวน 1 ครั้ง 5. เลือกสินค้ารหัส 9 โดยกดปุ่ม 'หยิบใส่รถเข็น' จำนวน 1 ครั้ง 6. กดปุ่ม 'รายการสั่งซื้อ' 7. กดปุ่ม 'ดูรายการสั่งซื้อ'	อยู่ในหน้า 'สั่งซื้อสินค้า'	แสดงข้อมูลผ่านทาง pop-up	- รหัสสินค้า 1 จำนวน 1 ชิ้น - รหัสสินค้า 3 จำนวน 1 ชิ้น - รหัสสินค้า 5 จำนวน 1 ชิ้น - รหัสสินค้า 7 จำนวน 1 ชิ้น - รหัสสินค้า 9 จำนวน 1 ชิ้น	- รหัสสินค้า 1 จำนวน 1 ชิ้น - รหัสสินค้า 3 จำนวน 1 ชิ้น - รหัสสินค้า 5 จำนวน 1 ชิ้น - รหัสสินค้า 7 จำนวน 1 ชิ้น - รหัสสินค้า 9 จำนวน 1 ชิ้น	Pass
2	เลือกซื้อสินค้าบางรายการมากกว่า 1 ชิ้น โดยมีการสลับไปกดปุ่ม 'หยิบใส่รถเข็น' ของสินค้าอื่น	-	1. เลือกสินค้ารหัส 2 โดยกดปุ่ม 'หยิบใส่รถเข็น' จำนวน 1 ครั้ง 2. เลือกสินค้ารหัส 6 โดยกดปุ่ม 'หยิบใส่รถเข็น' จำนวน 1 ครั้ง 3. เลือกสินค้ารหัส 4 โดยกดปุ่ม 'หยิบใส่รถเข็น' จำนวน 1 ครั้ง 4. เลือกสินค้ารหัส 2 โดยกดปุ่ม 'หยิบใส่รถเข็น' จำนวน 2 ครั้ง 5. เลือกสินค้ารหัส 6 โดยกดปุ่ม 'หยิบใส่รถเข็น' จำนวน 4 ครั้ง 6. กดปุ่ม 'รายการสั่งซื้อ' 7. กดปุ่ม 'ดูรายการสั่งซื้อ'	อยู่ในหน้า 'สั่งซื้อสินค้า'	แสดงข้อมูลผ่านทาง pop-up	- รหัสสินค้า 2 จำนวน 3 ชิ้น - รหัสสินค้า 6 จำนวน 5 ชิ้น - รหัสสินค้า 4 จำนวน 1 ชิ้น	- รหัสสินค้า 2 จำนวน 3 ชิ้น - รหัสสินค้า 6 จำนวน 5 ชิ้น - รหัสสินค้า 4 จำนวน 1 ชิ้น	Pass
3	เลือกหยิบสินค้าใส่รถเข็น จากนั้นทำการล้างข้อมูลในรถเข็น	-	1. เลือกสินค้ารหัส 3 โดยกดปุ่ม 'หยิบใส่รถเข็น' จำนวน 1 ครั้ง 2. เลือกสินค้ารหัส 9 โดยกดปุ่ม 'หยิบใส่รถเข็น' จำนวน 1 ครั้ง 3. เลือกสินค้ารหัส 3 โดยกดปุ่ม 'หยิบใส่รถเข็น' จำนวน 2 ครั้ง 4. กดปุ่ม 'รายการสั่งซื้อ' 5. กดปุ่ม 'ล้างรถเข็น'	อยู่ในหน้า 'สั่งซื้อสินค้า'	แสดงข้อมูลผ่านทาง pop-up	ไม่มีรายการสินค้าในรถเข็น	ไม่มีรายการสินค้าในรถเข็น	Pass

ตารางที่ 2 Adding Products to Cart Scenarios

หมายเหตุ ในการพัฒนาต่อไป จะปรับจากการแสดงข้อมูลผ่านทาง pop-up ไปสู่หน้ารายการสั่งซื้อ ซึ่งมีรายการสินค้าในรถเข็น จำนวนสินค้า ราคาสินค้า รวมไปถึงการเพิ่ม/ลด จำนวนสินค้า เพื่อให้ลูกค้าได้ตรวจสอบความถูกต้องก่อนยืนยันการสั่งซื้อ (ข้อมูลสินค้าในรถเข็นจะถูกส่งจากหน้าสั่งซื้อสินค้า ไปยังหน้ารายการสั่งซื้อ ผ่านทาง Cookie Session)

2) Test Case (PHP Codeception)

```
1  <?php
2  $I = new AcceptanceTester($scenario);
3  $I->wantTo('Test ID : 1');
4  $I->amOnPage('/Shop.html');
5  $I->seeCurrentUrlEquals('/HaHaHa/Shop.html');
6
7  $I->wait(1);
8  $I->click('input[id="shopAdd1"]');
9  $I->wait(1);
10 $I->click('input[id="shopAdd3"]');
11 $I->wait(1);
12 $I->click('input[id="shopAdd5"]');
13 $I->wait(1);
14 $I->click('input[id="shopAdd7"]');
15 $I->wait(1);
16 $I->click('input[id="shopAdd9"]');
17 $I->wait(1);
18 $I->click('a[id="clickCartDropDown2"]');
19 $I->wait(1);
20 $I->click('a[id="clickShowCart"]');
21 $I->wait(1);
22
23 //pop-up display cart list
24 $I->wait(1);
25 $I->seeInPopup('1 (1)');
26 $I->seeInPopup('3 (1)');
27 $I->seeInPopup('5 (1)');
28 $I->seeInPopup('7 (1)');
29 $I->seeInPopup('9 (1)');
30 $I->wait(1);
31 $I->acceptPopup();
32 $I->wait(1);
```

ภาพที่ 27 PHP Codeception สำหรับ Test ID 1


```
1 <?php
2 $I = new AcceptanceTester($scenario);
3 $I->wantTo('Test ID : 2');
4 $I->amOnPage('/Shop.html');
5 $I->seeCurrentUrlEquals('/HaHaHa/Shop.html');
6
7 $I->wait(1);
8 $I->click('input[id="shopAdd2"]');
9 $I->wait(1);
10 $I->click('input[id="shopAdd6"]');
11 $I->wait(1);
12 $I->click('input[id="shopAdd4"]');
13 $I->wait(1);
14 $I->click('input[id="shopAdd2"]');
15 $I->wait(1);
16 $I->click('input[id="shopAdd2"]');
17 $I->wait(1);
18 $I->click('input[id="shopAdd6"]');
19 $I->wait(1);
20 $I->click('input[id="shopAdd6"]');
21 $I->wait(1);
22 $I->click('input[id="shopAdd6"]');
23 $I->wait(1);
24 $I->click('input[id="shopAdd6"]');
25 $I->wait(1);
26 $I->click('a[id="clickCartDropDown2"]');
27 $I->wait(1);
28 $I->click('a[id="clickShowCart"]');
29 $I->wait(1);
30
31 //pop-up display cart list
32 $I->wait(1);
33 $I->seeInPopup('2 (3)');
34 $I->seeInPopup('6 (5)');
35 $I->seeInPopup('4 (1)');
36 $I->wait(1);
37 $I->acceptPopup();
38 $I->wait(1);
```

ภาพที่ 28 PHP Codeception สำหรับ Test ID 2

```
1 <?php
2 $I = new AcceptanceTester($scenario);
3 $I->wantTo('Test ID : 3');
4 $I->amOnPage('/Shop.html');
5 $I->seeCurrentUrlEquals('/HaHaHa/Shop.html');
6
7 $I->wait(1);
8 $I->click('input[id="shopAdd3"]');
9 $I->wait(1);
10 $I->click('input[id="shopAdd9"]');
11 $I->wait(1);
12 $I->click('input[id="shopAdd3"]');
13 $I->wait(1);
14 $I->click('input[id="shopAdd3"]');
15 $I->wait(1);
16 $I->click('a[id="clickCartDropDown2"]');
17 $I->wait(1);
18 $I->click('a[id="clickClearCart"]');
19 $I->wait(1);
20 $I->click('a[id="clickShowCart"]');
21 $I->wait(1);
22 $I->acceptPopup();
23 $I->wait(1);
```

ภาพที่ 29 PHP Codeception สำหรับ Test ID 3

3) Result

```
D:\HaHaHaCompany>php codecept.phar run
Codeception PHP Testing Framework v2.1.7
Powered by PHPUnit 4.8.23 by Sebastian Bergmann and contributors.

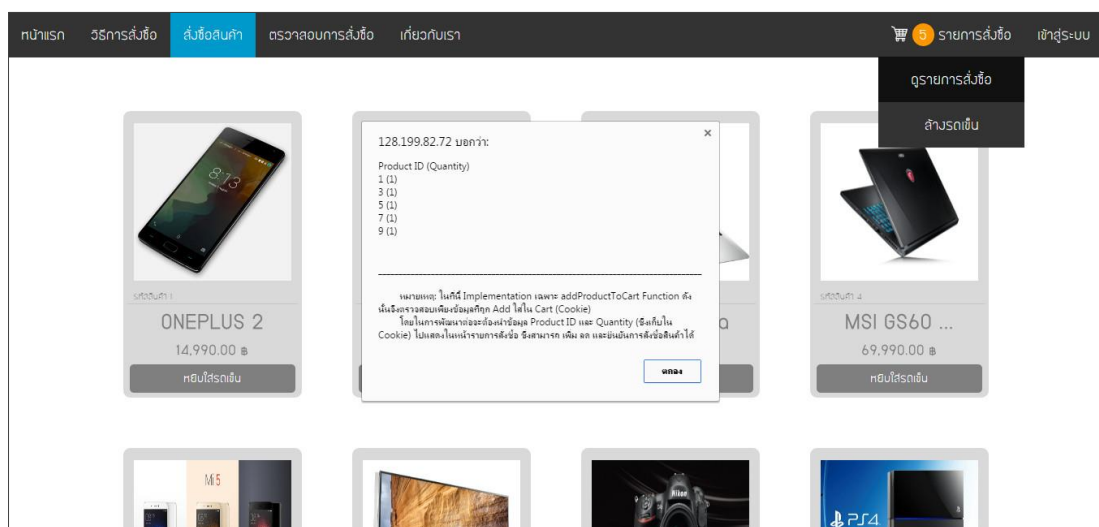
Acceptance Tests (3)
Test ID : 2 <HaHaHaTest1Cept>                                Ok
Test ID : 3 <HaHaHaTest2Cept>                                Ok
Test ID : 1 <HaHaHaTestCept>                                  Ok

Functional Tests (0)
-----

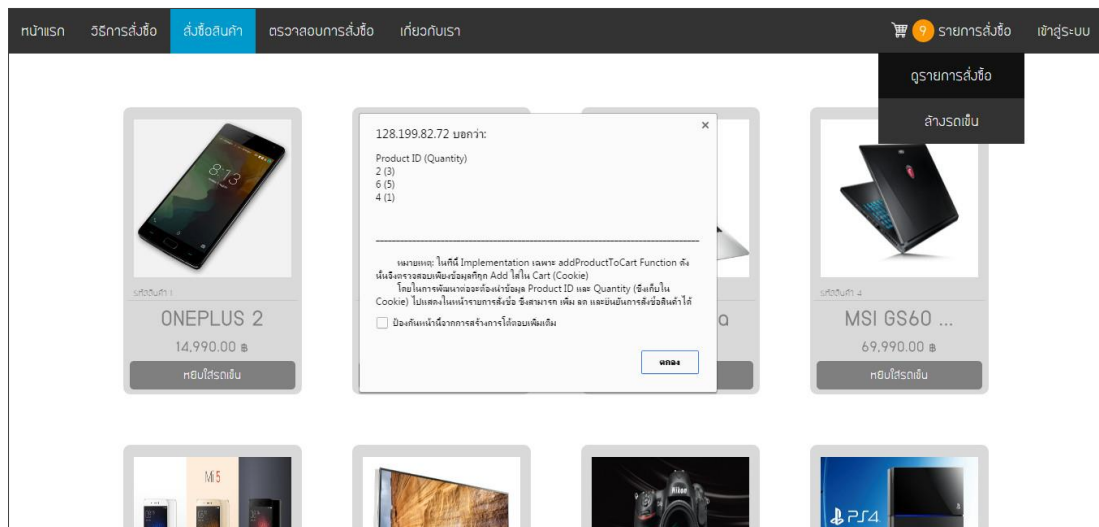
Unit Tests (0)
-----

Time: 52.34 seconds, Memory: 7.75Mb
OK (3 tests, 11 assertions)
```

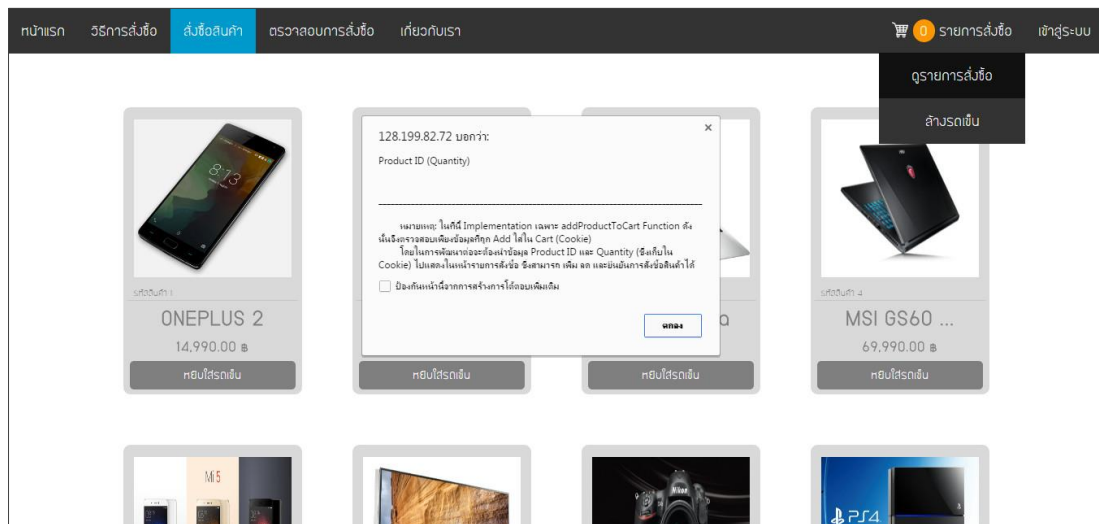
ภาพที่ 30 ผลลัพธ์จากการทดสอบด้วย PHP Codeception



ภาพที่ 31 ผลลัพธ์จากการทดสอบ Test ID 1



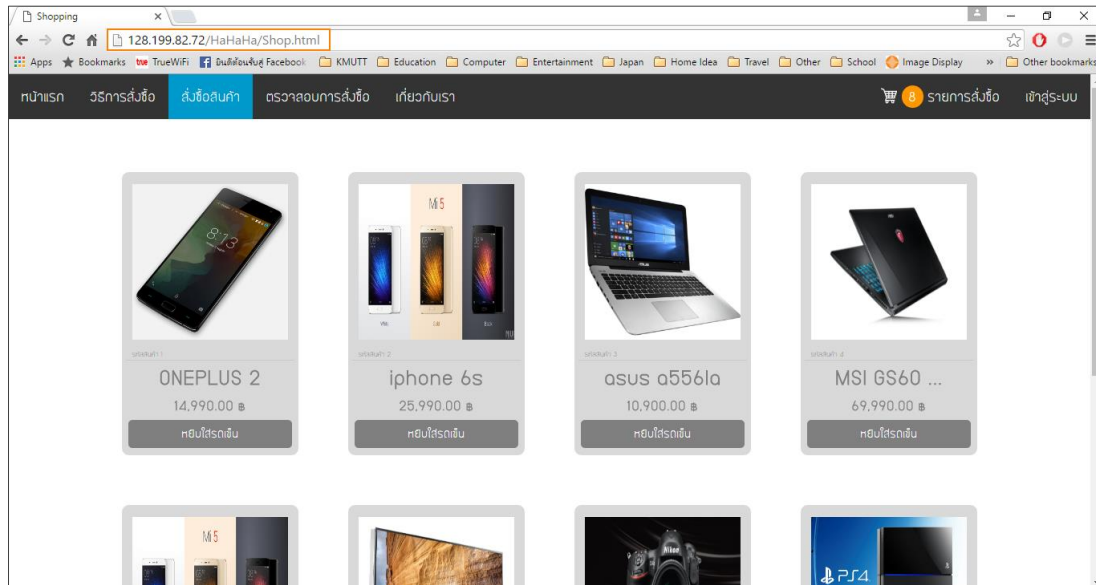
ภาพที่ 32 ผลลัพธ์จากการทดสอบ Test ID 2



ภาพที่ 33 ผลลัพธ์จากการทดสอบ Test ID 3

4. Instructions to Run Code

1. สามารถทำการ Run หรือเข้าสู่เว็บไซต์ได้ที่ 128.199.82.72/HaHaHa/Shop.html



2. สิ่งที่สามารถทำได้บนเว็บไซต์หรือข้อจำกัดที่เกิดจากการพัฒนาเพียง Function เดียว
 - 2.1 หน้า “สั่งซื้อสินค้า” (Shop.html) เป็นเพียงหน้าเดียวที่สามารถเข้าถึงได้ในตอนนี้
 - 2.2 สามารถเลือกสินค้าใส่รถเข็นได้ โดยกดที่ปุ่ม “หยิบใส่รถเข็น” ของสินค้านั้นๆ
 - 2.3 เมื่อกดปุ่มรูปรถเข็นที่แถบเมนูด้านบน จะแสดง Dropdown 2 รายการให้เลือกคือ รายการสั่งซื้อ และ ล้างรถเข็น
 - 2.4 กรณีปรับขนาดจอให้มีความกว้างน้อยกว่า 876px ปุ่มเมนูทางซ้ายทั้งหมดบนแถบเมนูจะถูกหุบเหลือเพียง Icon เมนูเพียงปุ่มเดียว ซึ่งเมื่อคลิกปุ่ม Icon ดังกล่าวจะแสดง Dropdown รายการเมนูทั้งหมดที่ถูกหุบ
3. หากต้องการดูฐานข้อมูล (Database) สามารถเข้าถึงได้ที่ 128.199.82.72/phpmyadmin/ โดยสามารถดู Username และ Password ได้จาก Connent.php
4. กรณีที่ไม่สามารถเข้าถึงเว็บไซต์ข้างต้นได้ และต้องการดำเนินการ Run Code โดย Remote Site ผ่านโปรแกรม Team Viewer สามารถติดต่อได้ที่
Facebook: Tun Positive Plus
E-mail: Tun.positive@gmail.com

5. GitHub Repository Link

<https://github.com/S-Suraphat/CPE333-Final-Project---HaHaHa-Company>

	ชื่อ	นามสกุล	รหัสนักศึกษา	ชื่อใน GitHub	หน้าที่หลัก
1	นายทีปภัทร	ศิริสมบุญเวช	56070501018	NackyTS	Database Development
2	นางสาวภัทราพร	มีมาก	56070501035	nimp68	Tester
3	นายสุภณัฐ	หัตตดิโนพานิช	56070501053	Nagisachan	Controlling Data on Client (javascript, xmlhttp)
4	นายสุรภัทร	แสงศิริ	56070501062	S-Suraphat	GUI (html, css, javascript)
5	นายศรสรณ์	รัตนนพดลชัย	56070501090	pingac127	PHP API Service (Intermediary between Client and Server)

ตารางที่ 3 รายชื่อสมาชิกและหน้าที่ที่รับผิดชอบ