International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:10, 2014

The Conceptual Design Model of an Automated Supermarket

Sathya Narayanan V., Sidharth P., Sanal Kumar. V. R.

***Abstract*—**The success of any retail business is predisposed by its swift response and its knack in understanding the constraints and the requirements of customers. In this paper a conceptual design model of an automated customer-friendly supermarket has been proposed. In this model a 10-sided, space benefited, regular polygon shaped gravity shelves have been designed for goods storage and effective customer-specific algorithms have been built-in for quick automatic delivery of the randomly listed goods. The algorithm is developed with two main objectives, viz., delivery time and priority. For meeting these objectives the randomly listed items are reorganized according to the critical-path of the robotic arm specific to the identified shop and its layout and the items are categorized according to the demand, shape, size, similarity and nature of the product for an efficient pick-up, packing and delivery process. We conjectured that the proposed automated supermarket model reduces business operating costs with much customer satisfaction warranting a win-win situation.

***Keywords*—**Automated Supermarket, Electronic Shopping, Polygon-shaped Rack, Shortest Path Algorithm for Shopping.

I. Introduction

THE supermarket, a commercial retail business, plays an important role in the economic development of any nation. The shopping experience is changing persistently over the years and the new generation customers prefer an early delivery of their randomly listed goods. Although many models have been designed and developed for the past three decades for an automated supermarket there are still many constraints in the existing models for meeting the fullest satisfaction of the customers in terms of the order-delivery process and the time saving [1]-[17]. In the existing models, in an automated store, packaged goods will be picked robotically in response to electronic orders placed by customers via personal digital devices, either in-store or online. Customers can self-select their fresh goods; when finished, customers will make payment, typically via smartphone and without any other checkout procedure, then leave the store and receive their orders at drive-through pick-up stations. For an additional fee, a home-delivery option will be available for all orders, whether generated online or by a store visit, most likely provided by a third party.

Literature review and the hands on experiences reveal that if there is one thing that the observations of retail markets from the past years has taught us, it is that there is a lot of demand for various types of items in the day to day life, which are not scientifically processed for a better delivery to the customer. This demand has only increased over the years because of several reasons such as population growth and changing lifestyle. Owing to the change in lifestyle, there is an exponential increase in the number of brands of a particular product under each category. So, as the number of items increase, the size of the conventional supermarket naturally increases. Hence, the time taken for an average person to search for items also increases. This brings about the need for automation in this regard [1]. The heart of automation lies on the disentanglement of the sophistication in the routine activities.

According to the data collected by American Time Use Survey (ATUS), an average person spends 40.14 minutes for shopping in a week [2]. According to another survey, Canadians go on 37 general stock-up trips per year at an average of 44 minutes in the store – with an additional 13 minutes of travel time – for a total of 57 minutes on average [3]. Canadians also go on an average of 76 quick pick-up trips per year at an average of 18 minutes in the store – with an additional 9 minutes of travel time – for a total of 27 minutes on average [3]. Thus, from the two independent reports, we can conclude that the minimum average time taken for shopping is 27 minutes and the maximum comes to be 57 minutes on average. The layout of a supermarket has been found to significantly impact a retailer's overall performance. Layouts are not only concerned with improved utilization of buildings and land but are very much concerned with increasing sales [17]. The results of a survey conducted by Punjaisri and Wilson [18] proved that layout has a big influence on customers and that the customers want stores to spend whatever it takes to create a layout that minimizes wasted steps and motion in the shopping process. Note that the usual retail stores originally displayed their product categories in an industrial department approach, which have produced the store layouts based on fruits, vegetables, magazines, cds, and so on. Despite improvements, the store remains organized in product categories as defined by the manufacturers or category buyers. This approach is company oriented and it fails to respond to the needs of the time-pressured consumer. Most retailers nowadays face challenges such as how to respond

Sathya Narayanan is an Undergraduate Student of the department of Mechatronics Engineering, Kumaraguru College of Technology, Coimbatore – 641 049, Tamil Nadu, India (Phone: +91-9894520580; e-mail: sathya452@gmail.com)

Sidharth Prabhu is an Undergraduate Student of the department of Electronics and Communication Engineering, Kumaraguru College of Technology, Coimbatore – 641 049, Tamil Nadu, India (e-mail: sidharth.prabukumar@gmail.com).

V. R. Sanal Kumar is Professor and Aerospace Scientist, currently with the Aeronautical Engineering Department, Kumaraguru College of Technology, Coimbatore – 641 049, affiliated to Anna University, Tamil Nadu, India (Phone: +91-9388679565; e-mail: vr_sanalkumar@yahoo.co.in).

consumer's ever-changing demands and how to adapt themselves to keen competition in dynamic market. These are succinctly reported by Ibrahim Cil [17]. It is well known that in the dynamic retail market, understanding changes in customer behavior can help managers to establish effective store layouts. But in this era it is extremely difficult to understand the customer behavior *a priori.* Therefore in this paper we proposed an automated supermarket model without demanding the physical presence of the customer. The details of the aforesaid model are demonstrated in the subsequent sessions.

II. Design of an Automated Supermarket

In our approach, we revolutionize the entire design of the supermarket and introduce efficiently modified algorithms so as to make a completely automated supermarket such that the average time taken by the customer to shop for goods is reduced. This reduction in time is apparent because it involves no physical presence of the customer to pick up the required products and the time taken to move between aisles that contain different category of items is almost completely eliminated.

A. Rack Design

Rack design plays a major role in shopping scenario. The shape, size and location of each rack are sometimes based on the buying patterns of the customers. Conventional supermarket is closely related to manual order picking warehouses in terms of operation. Searching and picking accounts for nearly 35 % of overall timing of the manual order picking in a warehouse [11]. So time spent in these processes can be considerably reduced by applying certain algorithms and optimal rack design which complements the algorithms to the maximum.

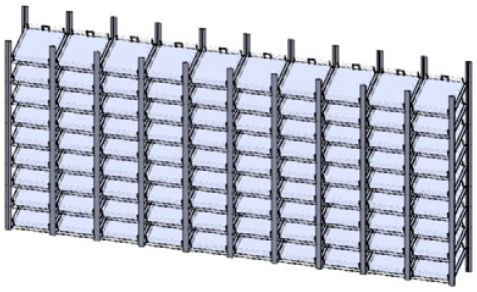

Fig. 1 Conventional Rack Design

A conventional rack consists of rows and columns arranged in a rectangular shape (see Fig. 1). Automated Storage and Retrieval System (AS/RS) is placed in front of the rack to retrieve items. The proposed design (see Fig. 2) is a 10-sided regular polygon [1] from the top-view, with arrangement of the shelves slightly tilted towards the origin making it similar to the gravity shelves depicted by Susan Van Zelst et al. [11]. Many criteria are available for designing objectives for shelf space allocation. The most frequently used objectives can be categorized to three classes: cost, sales or profit, productivity [12]. The store capacity constraint which designates that the aggregate shelf space allocated to all products cannot exceed total available shelf space in the store [12]. Owing to this, we found that the productivity of the market can be considerably enhanced with reduction in time.

Both designs have the same size in terms of number of rows and columns. Hence, the area remains the same. However, the volume occupied by our setup for the same dimensions is increased by 31.25%, when compared to the conventional rack. Nevertheless, this area can be properly utilized for storage place for the highest consumable items for quick automatic refilling. This difference is mainly due to the shape of the conventional rack. Our main objective here is to reduce the overall time taken for item selection and retrieval as much as possible when compared to the conventional style.

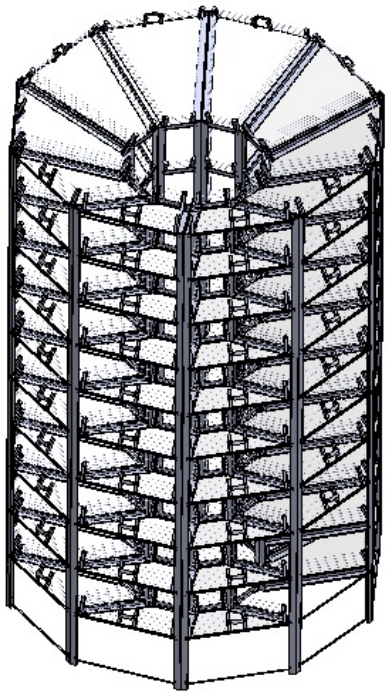

Fig. 2 Proposed Rack Design

B. Item Placement

Warehouse managers are interested in finding the most economical way of picking orders, which minimizes the costs involved in terms of travel distance or travel time. A batch is a group of orders that is simultaneously picked in a single tour. In the case of batch picking, orders are generally grouped into batches in an optimum manner under the criteria of minimum travel distance or minimum travel time [13]. Thus, the conveyer belt from the rack is placed in the most strategic location just below the most demand items, so that there is least time taken while picking up the items. Placing related departments or product categories close to each other have substantial impact on the retailers' profitability. Note that the empirical studies [14], stochastic models [15] and agent based modeling applications [16] support this fact [17]. The location of the items less in demand can be spread out in such a way

that the least priority item is at the direct opposite end of the location of the conveyer belt. For this purpose, the items are placed in such a way that the frequently, moderately and rarely bought items, are arranged in the shelves in such a way that it can be reached with relatively easier. This arrangement is shown in Fig. 3.

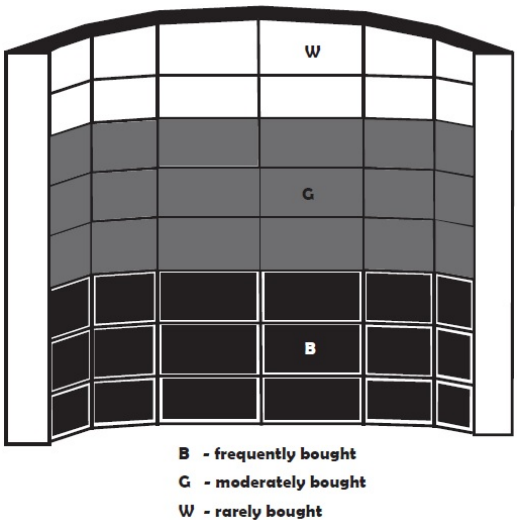

Fig. 3 Item Arrangement in Shelves

III. Selection of Frequent Items

Often times during purchases, there occurs instances where an item in a particular category is bought more frequently than others. When it comes to customer preferences, there is little to do in order to change it. So, instead of changing their preferences, we make the arrangement in favor of the customer. So, in order to make an arrangement such that the high priority items are more in stock and are easily accessible, we must first be able to identify what those items are. This is the sole reason for the development of this unique algorithm dedicated for the purpose of identifying the frequently preferred items from a list of items. This method is similar to the *a priori* algorithm [4]. The difference is elaborated as follows.

1. Unlike the *a priori* algorithm, we do not make use of candidate sets. This is mainly because our application is clearly specific and does not require the recognition of frequent item sets.
2. Like the *a priori* algorithm, the first pass of the algorithm simply counts item occurrences to determine the large item occurrences [4]. These are otherwise called as support.

The working of our algorithm is described below:

Input: List of items along with their IDs.
Output: Decreasing order of the most frequently preferred items.
Step 1: Start
Step 2: Scan the item list once and identify the unique items and their support. Store this value in the array A[I], where 'I' is the corresponding item ID.
Step 3: Assign a new array Q[k] and P[k] such that P[k]=A[I] and Q[k]=corresponding item ID of A[I], where k=0,1,2,…$(A[I])_{MAX}$
Step 4: Arrange the items in P[k] in descending order and map the Item ID to the corresponding P[k] value.
Step 5: Stop

For sorting in the descending order, we made use of the quick sort algorithm because according to [5], it is the fastest algorithm when compared to all the existing sorting algorithms. Since we are working on the time constraint, it only seems wise to go for this method. The quick sort algorithm developed by C.A.R. Hoare [9] is explained as follows.

Input: Unsorted array of items.
Output:Sorted array in the descending order
Step 1: Start
Step 2: Pick an element, called a pivot, from the array.
Step 3: Reorder the array so that all elements with values greater than the pivot come before the pivot, while all elements with values less than the pivot come after it (equal values can go either way)
Step 4: After this partitioning, the pivot is in its final position. This is called the partition operation.
Step 5: Recursively apply the above steps to the sub-array of elements with greater values and separately to the sub-array of elements with smaller values.
Step 6: Stop.

As an example, the following sample list was fed as input to the system:

No. of Items: 15
Item IDs: 28, 32, 46, 28, 46, 46, 65, 28, 32, 8, 4, 78, 32, 65, 28
At the end of the algorithm, the following was displayed:

Item 28 occurs 4 times.
Item 32 occurs 3 times.
Item 46 occurs 3 times.
Item 65 occurs 2 times.
Item 4 occurs 1 times.
Item 8 occurs 1 times.
Item 78 occurs 1 times.

This algorithm can further be improved by setting an arbitrary value as the threshold and a condition such that only the frequency of items exceeding the threshold is displayed after the sorting.

IV. Shortest Path for Item Retrieval

The problem when considering multiple racks where different items are present, there is always a time constraint while choosing the most efficient path for the product retrieval from the appropriate racks consuming the least time delay possible. In order to overcome this, we have come up with a novel method to efficiently choose the shortest path with the least time possible. This algorithm was chosen after acute appraisals of the various shortest path algorithms like Djikstra's algorithm [7], Floyd Warshall Algorithm [6] and with reference from [8]. We realized that these algorithms worked efficiently in cases such as graph traversals, but

weren't as efficient in our application. The algorithm is depicted as follows.

A. Formula for Relative Distance Calculation between Each Items

Due to the arrangement of each rack, every item on the rack behaves as a specific location on the Cartesian plane, where the co-ordinates of the x and y axis are the row and column address respectively. The relative distance between two nodes can be calculated using the formula:

$$D = \sqrt{[(x2 - x1)2 + (y2 - y1)2]} \rightarrow \quad (1)$$

where, (x_1, y_1)- Address of the location of item 1.
(x_2, y_2)- Address of the location of item 2.

B. Algorithm for Finding the Distance between the 2 Items

Parameters to consider:
Nc = number of columns
Nr = number of rows.
C_i = column address of item.
R_i = row address of item.
D = Distance between the 2 items.

Owing to the circular arrangement of the rack, there are some alterations to the original formula introduced by (1). Let the address of the two values whose distance is to be evaluated be (R_i,C_i) and (R_j,C_j)

Step 1: Start

Step 2: Check if $| Ci - Cj | > Nc/2$. If true, do steps 3. Else, go to step 7.

Step 3: Check if $C_j > C_i$. If true, do step 4. Else do step 5.

Step 4: Assign high<=C_j and low<=C_i.

Step 5: Assign high<=C_i and low<=C_j.

Step 6: Assign

$$D = \sqrt{[(Rj - Ri)2 + \{(Nc - high + 2) - low\}2]}$$

Step 7: Assign $D = \sqrt{[(Rj - Ri)2 + (Cj - Ci)2]}$

Step 8: Stop.

C. Shortest Distance Calculation between 'n' Number of Items

In order to calculate the most efficient path, we have made use of an algorithm which is the modified version of another algorithm which is used for re-ordering the test vectors for minimizing power consumption in VLSI circuits [10]. The modification done in our algorithm is the calculation of the relative distance between each of the items. In [10], the calculation was for the various hamming distance between the test vectors. In our implementation, we merely replaced the function of the hamming distance calculation with our relative distance calculation formula. The formation of the adjacency matrix and the reduction is explained as follows.

1) Formation of the Adjacency Matrix

In order to form the matrix, the distance between each of the individual items between others and itself are to be evaluated and represented in the form of a matrix. This method can be illustrated as follows;

Let Nc = 5
Nr = 5

D[i,j] = Distance between the i^{th} and the j^{th} item.

As an example, let us consider the fetching of 6 items located at the following addresses:

Item p1: (1,5)
Item p2: (2,1)
Item p3: (5,5)
Item p4: (1,1)
Item p5: (3,5)
Item p6: (4,2)

Then, the adjacency matrix D[ ][ ] can be represented as follows:

TABLE I
REPRESENTATION OF ADJACENCY MATRIX

Columns Rows	*P1*	*P2*	*P3*	*P4*	*P5*	*P6*
P1	*0*	*8.06*	*4*	*8*	*2*	*7.62*
P2	*8.06*	*0*	*8.54*	*1*	*8.06*	*2.24*
P3	*4*	*8.54*	*0*	*8.94*	*2*	*7.07*
P4	*8*	*1*	*8.94*	*0*	*8.25*	*3.16*
P5	*2*	*8.06*	*2*	*8.25*	*0*	*7.07*
P6	*7.62*	*2.24*	*7.07*	*3.16*	*7.07*	*0*

From Table I the nodes (p1, p1), (p1, p2), (p1, p3), … (p6, p6) are the distance between the corresponding row and column element which is found using the algorithm for finding the distance between the two items shown in the previous sub-section.

2) Adjacency Matrix Reduction

The various parameters used in this algorithm are as follows:

$P_1, P_2, P_{3,\ldots}P_n$ be n items

T = {1,2,… k … n} where k represents k^{th} position in the item.

R is a set to store ordered item sequence.

Q is a set to store T-R.

The algorithm depicted by [9] is shown below.

Step 1: Select an item x such that swa_init[x] is minimum in the array swa_init[ ]. Add x to set R.

Step 2: Select an item y such that $D[x][y_{min}]$ is minimum in the array.

Step 3: Add y_{min} to R; Q←(T-R); x←y .

Step 4: From the array $D[x_{min}, j]$ when j varies as in Q, find y_{min} so that $D[x_{min}][y_{min}]$ is the smallest value. Go to step 3.

Step 5: In the step 4, if $D[x_{min}][j]$ has more than one smallest value, then such number of reordered sequence will be generated for every x_{min}. These sequences are called as sub-optimal sequences.

Finally the set R will have reordered item sequence with minimum distance.

Now, on application of this algorithm to the matrix formed in the previous sub-section, we obtain:

Unordered Sequence: p1→p2→p3→p4→p5→p6

Distance: 40.87

Reordered Sequence: p2→p1→p6→p3→p4→p5

Distance: 12.4 units

Thus, the total distance for the reordered sequence is 12.4, which gives a 69.7% reduction in the total distance when compared to the unordered sequence. The difference in travel distance between two models is because of the design which forces the conventional model to have a constant in addition to mark the distance to return to the initial position. Thus, this approach will reduce the time taken to retrieve the items from the various racks.

V. Results and Discussion

In this paper a conceptual design model of an automated customer-friendly supermarket has been proposed. In this model a 10-sided, space benefited, regular polygon shaped gravity shelves have been designed for goods storage and effective customer-specific algorithms have been built-in for quick automatic delivery of the randomly listed goods [1]. Although the effective volume of the proposed polygon-shaped rack is relative lesser than the conventional rack occupied in the give space, the remaining unused space can be effectively utilized for storage place for the highest consumable items for quick automatic refilling.

In this model the algorithm is developed with two main objectives, viz., delivery time and priority. For meeting these objectives the randomly listed items are reorganized according to the critical-path of the robotic arm specific to the identified shop and its layout and the items are categorized according to the demand, shape, size, similarity and nature of the product for an efficient pick-up, packing and delivery process. An experiment was conducted implementing our algorithm to find the shortest path in picking up items for various item sets comparing the conventional design and our proposed design. The results saw a 31.5 % overall efficiency in our proposed design when compared to the conventional one.

TABLE II
Distance Comparison of Various Item Sets

Item Set	Frequentl y Bought (%)	Moderately Bought (%)	Rarely Bought (%)	Convention al Design (units)	Proposed Design (units)
1	100	0	0	31.72	26.72
2	0	100	0	34.41	25.41
3	0	0	100	26	15
4	50	30	20	41.08	27.67
5	30	20	50	42.38	26.55
6	20	50	30	12.41	7.41

Table II shows the difference in the distance taken for the shortest path in item retrieval for the conventional versus the proposed design.

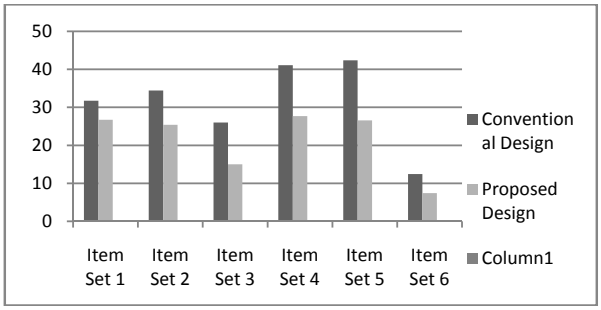

Fig. 4 Graph Comparing the Shortest Distance between the items for conventional design and proposed design

The graph (see Fig. 4) is further used to restate our proposition that the circular rack design is superior to the conventional rack design in terms of time and power consumption.

VI. Concluding Remarks

The concept of automated supermarket is implemented and it was observed that our proposed model out-performs the conventional model in the aspect of time consumption. We propose the conceptual design of an automated supermarket by providing effective customer-specific algorithm coupled with a novel rack design and arrangement, layout and packing method for meeting the needs of the customers. Admittedly the RFID technology will reduce inventory and the problems caused by manual intervention. Therefore in stores, through mounting electronic RFID devices on the store shelves, using the wireless network to be rapid and accurate positioning of products, in order to realize the fast interaction between the customers and the commodity. The proposed algorithm is developed with two main objectives, viz., delivery time and priority. For meeting these objectives the randomly listed items are reorganized according to the critical-path of the automated arm specific to the identified shop and its layout and the items are categorized according to the demand, shape, size, similarity and nature of the product for an efficient pick-up, packing and delivery process. We concluded that our proposed model reduces business operating costs and the cost of the customers and thereby both sides realize a win-win situation. The constraints such as item recognition, robotic arm design and conveyer belt arrangement are beyond the scope of this paper and will be addresses in the future connected paper.

Acknowledgment

The authors would like to thank Shankar Vanavarayar, Joint Correspondent of Kumaraguru College of Technology, Coimbatore – 641 049, Tamil Nadu, India for his extensive support of this research work.

References

[1] Sathya Narayanan. V., Sidharth. P., Sanal Kumar. V. R, "Robot Controlled Automated Supermarket Model," Indian Patent (pending), October, 2014 (5070/CHE/2014).

[2] Goodman. J, "Who Does the Grocery Shopping, and When Do They Do It?" survey by The Time Use Institute,2008.
[3] Environics Research Group "Checking out the Canadian Grocery Shopping Experience" survey by MasterIndex^{TM} May 14-19, 2008.
[4] Agrawal R. and Srikant R. "Mining Sequential Patterns". In proceedings of the 1995 *International Conference Data Engineering*,1995, pp-3-14
[5] Christian Bunse H. H., "Choosing The "Best" Sorting Algorithm For Optimal Energy Consumption." *ICSOFT-International Joint Conference on Software Technologies* INSTICC Press, 2009. pp. 199-206.
[6] Floyd, R. W., "Algorithm 97, Shortest Path," Comm. ACM, Vol. 5, No. 6, June 1962, p. 345.
[7] Dijkstra, E. W., "A Note on Two Problems in Connexion with Graphs," NumerischeMathematik, Vol. 1, 1959, pp. 269-271.
[8] Hitchner, L. E., A Comparative Investigation of the Computational Efficiency of Shortest Path Algorithms, Operations Research Center, University of California, Berkeley, Report ORC 68-17, July 1968.
[9] C.A.R. Hoare, "Algorithm 64: Quicksort". *Communications of the ACM*,Vol. 4 Issue 7, 1961.pp 321
[10] K.Paramasivam D., "Reordering Algorithm for Minimizing Test Power in VLSI Circuits" *Engineering Letters, 14:1*,February 2007.
[11] Susan Van Zelst, KarelvanDonselaar, TomvanWoensel,Rob Broekmeulen, JanFransoo, "Logistics drivers for shelf stacking in grocery retail stores: Potential for efficiency improvement" *International Journal of Production Economics,* 2009 pp 620-632
[12] Ming-Hsien Yang, "An efficient algorithm to allocate shelf space" *International Journal of Operational Research*, 2001 pp 107-118.
[13] Chih-Ming Hsu, Kai-Ying Chen, Mu-Chen Chen, "Batching orders in warehouses by minimizing travel distance with genetic algorithms" *Computers in Industry*, 2005 pp 169-178
[14] Jeffrey, S.L., Bradlow, E.T., & Fader, P. An exploratory look at in-store supermarket shopping paths. *International Journal of Research in Marketing, 22(4)*, 2005, pp 395-414.
[15] Farley, J., & Ring, L. W. A stochastic model of supermarket traffic flow. *Operational Research, 14,*1966, pp555-567.
[16] Batty, M. Agent-based pedestrian modeling. In P. Longley & M. Batty (Eds.), *Advanced spatial analysis.*2003. ESRI Press.
[17] Ibrahim Cil. Consumption universes based supermarket layout through association rule mining and multidimensional scaling. *Expert Systems with Applications.*2012. pp 8611-8625.
[18] Punjaisri, K., & Wilson, A. The role of internal branding in the delivery of employee brand promise. Journal of Brand Management, 2007, 15(1), 57–70.