ONLINE SHOPPING BASED HOME DELIVERY SYSTEM USING A*

MYAT SAN NWE

M.C.Sc. (THESIS)

JULY, 2016

ONLINE SHOPPING BASED HOME DELIVERY SYSTEM USING A*

BY

MYAT SAN NWE

B.C.Sc. (Honours)

Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Computer Science
(M.C.Sc.)

of the

University of Computer Studies, Mandalay

JULY 2016

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to **Dr. Moe Pwint**, Rector of the University of Computer Studies, Mandalay, for giving me general guidance and encouragement for the completion of this thesis.

I would like to express my deeply gratitude and my sincere thanks to **Dr. Myat Myat Min**, Dean of the Class, Associate Professor, Faculty of Computer Science at the University of Computer Studies, Mandalay, for giving me the valuable guidance and suggestions during the period of developing this thesis.

I wholeheartedly express my sincere gratitude to my thesis supervisor, **Dr. Zin Mar Kyu**, Lecturer of Faculty of Computer Science at the University of Computer Studies, Mandalay, for her encouragement, patient supervision, invaluable detailed guidance, giving expert advice, motivation and support throughout the period of study.

I would like to express my special thanks to **Daw Naw Wint Wint**, Lecturer of Department of Linguistics at the University of Computer Studies, Mandalay, for editing my thesis from the language point of view.

I would like to express my special thanks to **all my teachers** who gave me their time and guidance, and all my friends who helped in the task of developing this thesis. Finally, I would like especially to thank **my parents** for their continuous support and encouragement throughout my whole life.

ABSTRACT

Since the Internet has become popular, the consumer oriented e-commerce market has grown. Home delivery plays an important role in online shopping. Online shopping may stimulate a shift from the need for personal travel to demand for the distribution of goods. The system provides a product delivery system for online shopping market. In this system, the new branches can be opened in Mandalay division and goods are delivered to customers who live in Mandalay division. This system use Dijkstra's algorithm to find distances between cities of Mandalay division. When customer order the goods, the system find the nearest branch of the customer's address and search the shortest paths from nearest branch to the customer address by using A* search algorithm. This system arranges the home delivery paths for each branch. This system is implemented by using Java Programming Language.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

and therefore there is no possibility for the customers to collect the goods themselves: instead they have to deliver to the customer.

1.2 Application Areas of Path Finding Algorithms

Path finding algorithms have many usage areas. These algorithms are useful in the field of robotics, since they can be used to guide a robot around difficult terrain without human intervention. This would be useful if the robot were on another planet like Mars, where some terrain must be avoided, but due to the extreme distances involved, controlling it completely via remote control would be impossible (too much delay in the radio transmission). It could also be useful if the robot were to operate underwater, where radio waves could not get to it. Path finding algorithms could also be used in almost any case where a vehicle needs to go somewhere, while avoiding obstacles, without human intervention. Another use is in computer games where something needs to be moved from one place to another, avoiding any walls or other obstacles in the way. These algorithms could also be used to find the shortest path to drive between two points on a map, the best way to route an e-mail through a computer network, or the shortest path to run telephone wires through existing circuits.

1.3 Heuristic Search Techniques

Many if the problems are too complex to be solvable by direct techniques. They have to be solved only by suitable heuristic search techniques. Though the heuristic techniques can be described independently, they are domain specific. They are called "Weak Methods", since they are vulnerable to combinatorial explosion. Even so,

they provide the frame work into which domain specific knowledge can be placed.

Every search process can be viewed as a traversal of a directed graph, in which the nodes represent problem states and the arcs represent relationships between states. The search process must find a path through this graph, starting at an initial state and ending in one or more final states.

A heuristic procedure, or heuristic, is defined as having the following properties.

- It will usually find good, although not necessary optimum solutions.
- It is faster and easier to implement than any known exact algorithm (one which guarantees an optimum solution).

In general, heuristic search improve the quality of the path that are exported. Using good heuristics we can hope to get good solutions to hard problems such as the traveling salesman problem in less than exponential time. There are some good general purpose heuristics that are useful in a wide variety of problems. It is also possible to construct special purpose heuristics to solve particular problems.

1.4 Objectives of the Thesis

The objectives of the system are as following:

- To develop a business to consumer (B2C) home delivery system
- To study about A* search algorithm and Dijkstra's algorithm
- To solve the difficulty for shopping at physical locations

1.5 Organization of the Thesis

This thesis consists of five chapters including this chapter. Chapter 2 explores relevant previous works and the background theory of the system. Chapter 3 explains the system design. Chapter 4 describes the implementation of the system. In chapter 5 as the conclusion, this chapter will mention the significance of this work, and present directions of further research.

CHATPER 2

BACKGROUND THEORY

2.1 Path Finding Algorithms

There are many algorithms that are commonly known and used, ranging from simplex to complex, in order to be able to solve the path finding problem. The simplest approach is to walk directly towards the goal until it met with any kind of obstacles. When it met with any object, direction will be changed and it can be passed by tracing around the obstacle. This type of algorithm is an example of the "visually impaired search" since it does not know or have any information about the path. Some other examples of the visually impaired Search are Breadth-First Search, Dijkstra's Algorithm and Depth-First Search. There also exist some algorithms that plan the whole path before moving anywhere. Best-first algorithm expands nodes based on a heuristic estimate of the cost to the goal. Nodes, which are estimated to give the best cost, are expanded first. The most commonly used algorithm is A* algorithm, which is a combination of the Dijkstra algorithm and the best-first algorithm. The different algorithms work in different ways. The breadth-first search begins at the start node, and then examines all nodes one step away, then all nodes two steps away, then three steps, and so on, until the goal node is found. This algorithm is guaranteed to find a shortest path as long as all nodes have a uniform cost. The bi-directional breadth-first search is where two breadth-first searches are started simultaneously, one at the start and one at the goal, and they keep searching until there is a node that both searches have examined. The final path is the combination of the path from the start to the intersection node, and the path from the goal to the intersection node.

2.2 Dijkstra's Algorithm

Dijkstra's algorithm looks at the unprocessed neighbors of the node closest to the start, and sets or updates their distances (in terms of cost, not number of nodes) from the start. The Dijkstra algorithm expands the node that is farthest from the start node, so it ends up "stumbling" into the goal node. Just like the breadth-first search; it is guaranteed to find the shortest path.

The depth-first search extends nodes (it extends a node's descendants before its siblings) until it either reaches the goal or a certain cut-off point, it then goes onto the next possible path. The best-first search algorithm is a heuristic search algorithm, meaning that it can take knowledge about the map into account. It is similar to Dijkstra's algorithm, but it goes to the node closest to the goal, rather than the node furthest from the start.

```
1  function Dijkstra(Graph, source):
2    dist[source] ← 0                          // Initialization
3    for each vertex v in Graph:
4      if v ≠ source
5        dist[v] ← infinity                    // Unknown distance from source to v
6        prev[v] ← undefined                   // Predecessor of v
7      end if
8      Q.add_with_priority(v, dist[v])
9    end for
10
11   while Q is not empty:                     // The main loop
12     u ← Q.extract_min()                     // Remove and return best vertex
13     for each neighbor v of u:
14       alt = dist[u] + length(u, v)
15       if alt<dist[v]
16         dist[v] ← alt
17         prev[v] ← u
18         Q.decrease_priority(v, alt)
19       end if
```

```
20      end for
21    end while
21    return prev[]
```

Figure 2.1 Pseudo Code for Dijkstra Algorithm

2.3 A* Algorithm

The algorithm was first described in 1968 by Peter Hart, Nils Nilsson, and Bertram Raphael [13]. It is a global space-search algorithm that can be used to find solutions too many problems, path finding being just one of them. It has been used in many real-time strategy games and is probably the most popular path finding algorithm. The A* algorithm works much like the Dijkstra and best-first algorithms only it gives values to the nodes in a different way. Each node's value is the sum of the actual cost to that node from the start and the heuristic estimate of the remaining cost from the node to the goal. In this way it combines the tracking of previous length from Dijkstra's algorithm with the heuristic estimate of the remaining path from the best first search. In fact, the Dijkstra search is an A* search, where the heuristic is always 0. This algorithm also makes the most efficient use of the heuristic function, meaning that no other algorithm using the same heuristic will expand fewer nodes and find an optimal path. A* algorithm uses a starting point and a destination point to produce the desired path, if it exists as in Figure 2.2. In figures, the cell marked with "O" is our starting point/node and the cell marked with "X" is the destination. White squares are walkable nodes and the black ones are walls, shelves or any other obstacles.

Figure 2.2 Starting and Destination Points in A* Search Algorithm

A* algorithm starts to execute by looking at the starting node first and then expanding to the surrounding nodes as describe in Figure 2.3.

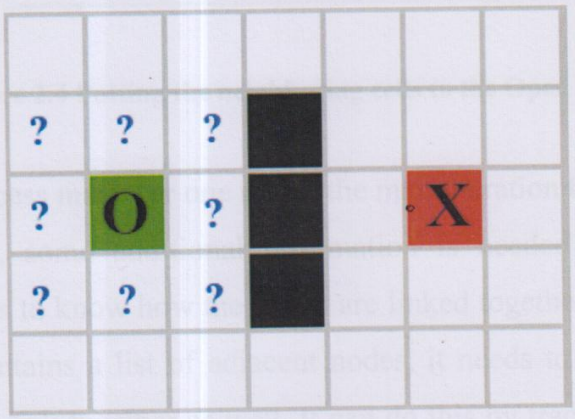


Figure 2.3 Starting the A* Search Algorithm

This operation continues until the destination node is found. In order to know the nodes, which will be used in search, A* algorithm needs a way to keep track of the nodes. So the nodes to be examined are held in a list, called Open List. At the beginning we place the starting nodes to the Open List, and after examining all of its surrounding nodes we will move it from the Open List and place in another list called the

Closed List as in Figure 2.4. Closed List holds the nodes that are visited and there is no need to re-visit its members. When building the Open List, algorithm checks if the node is walkable. If the node is not walkable, it is not added to the Open List.

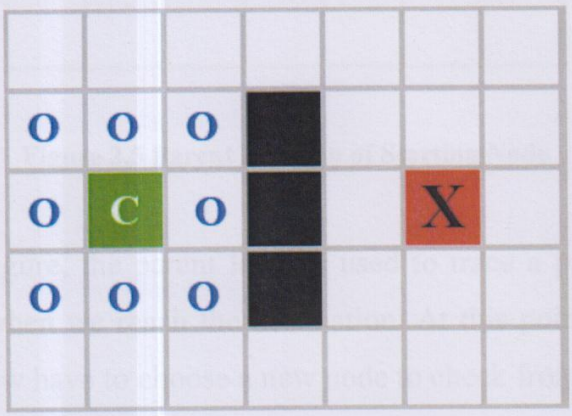


Figure 2.4 Putting the neighboring cells to the Open List

This process made for one cell is the main iteration through one A* loop; however, some additional information is needed to track. The algorithm needs to know how the nodes are linked together. Although the Open List maintains a list of adjacent nodes, it needs to know how the adjacent nodes link together as well. It can do this by tracking the parent node of each node in the Open List. A node's parent is the single node that we step from to get to its current location. On the first iteration through the loop, each node will point to the starting node as its parent as describe in Figure 2.5.

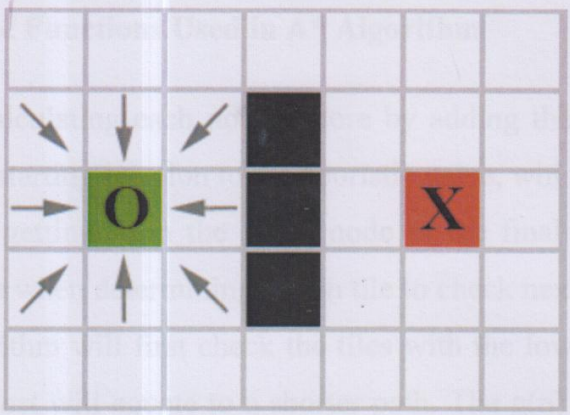


Figure 2.5 Parent Relation of Starting Node

In this figure, the parent links is used to trace a path back to the starting node when we reach the destination. At this point we start over process. We now have to choose a new node to check from the Open List. At the first iteration we had only a single node in the Open List. We now have eight nodes in the Open List, and the node which will first be inspected is determined by assigning a score to each node. This score, f(n), is the combination of two scores:

$$f(n) = g(n) + h(n)$$

where g(n) is the cost of the path from the starting node to any node n, h(n) is the heuristic estimated cost from any node n to the goal. We select the node with the lowest score. We use a Priority Queue to build the Open List, and the nodes that will be added to Open List is sorted by this score. So when we pop an element from this Queue, we always get the node with the lowest score. There are some well-known heuristics used in scoring.

2.3.1 Heuristic Functions Used in A* Algorithm

When calculating each node's score by adding the cost of getting there from the starting location to the heuristic value, which is an estimate of the cost of getting from the given node to the final destination. By using this score when determining which tile to check next from the Open List. The algorithm will first check the tiles with the lowest cost. In this case, a lower cost will equate to a shorter path. The g(n) value shown in each open node is the cost of getting there from the starting node. In this case, each value is 1 because each node is just one step from the starting node. The h(n) value is the heuristic. The heuristic is an estimate of the number of steps from the given node to the destination node. The algorithm does not take obstacles into consideration when determining the heuristic. So, it has not examined the nodes between the current node and the destination node, so it is not really know yet if they contain any obstacles. At this point, it simply wants to determine the cost, assuming that there are no obstacles. The final value is f(n), which is the sum of g(n)and h(n). This is the cost of the node. It represents the known cost of getting there from the starting point and an estimate of the remaining cost to get to the destination.

Heuristics used in A* algorithm have some variations, but for grid based maps there are three heuristic functions that work well.

(i) Manhattan Distance:

$$h(n)=D*(|n.x-goal.x| +|n.y-goal.y|)$$

where D represents the cost of moving one node to one of its neighbors Manhattan Distance as in Figure 2.6 allows us to move in four directions (North, South, East and West).

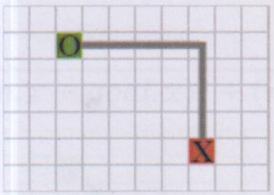


Figure 2.6 Manhattan Distance Calculation

The Manhattan heuristic is computed by adding the differences in the x and y components together. The advantage of using this heuristic is that, it is computationally inexpensive, and it can run faster than the others. The major disadvantage of the Manhattan heuristic is that it tends to overestimate the actual minimum cost to the goal (unless 4-adjacency is used) which means that the road being found may not be an optimal solution. If we are not interested in an optimal solution, but just a good one, then using an overestimating heuristic can speed up the road finding.

(ii) Diagonal Distance:

$$h(n)=D*max(|n.x-goal.x|,|n.y-goal.y|)$$

Diagonal Distance allows us to move in 8 directions as in Figure 2.7.

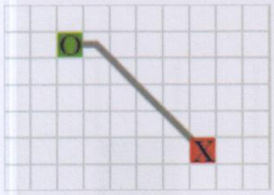


Figure 2.7 Diagonal Distance Calculation

Diagonal method balances the h(n) and g(n) in calculation of the total cost, but it is a bit slower than Manhattan method.

(iii) Euclidean Distance:

$$h(n) = D * \sqrt{((n.x - goal.x)^2 + (n.y - goal.y)^2)}$$

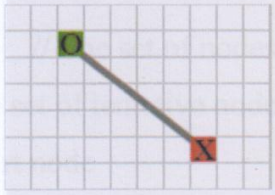


Figure 2.8 Euclidean distance calculation

The Euclidean heuristic as in Figure 2.8 is admissible, but usually underestimates the actual cost by a significant amount. This means that we may visit too many nodes unnecessarily which, as a result, increases the time it takes to find the road. The Euclidean distance heuristic is also computationally more expensive to apply compared to the Manhattan heuristic, as it additionally involves two multiplication operations and calculating the square roots. As it search the nodes toward the goal we add them to Open List and transfer to Closed List when our job with them completed. These operations continue until we reach to the goal. When we reach to the destination, we do a back-track by using the parent links for all of the nodes and construct the path as the following Figure 2.9.

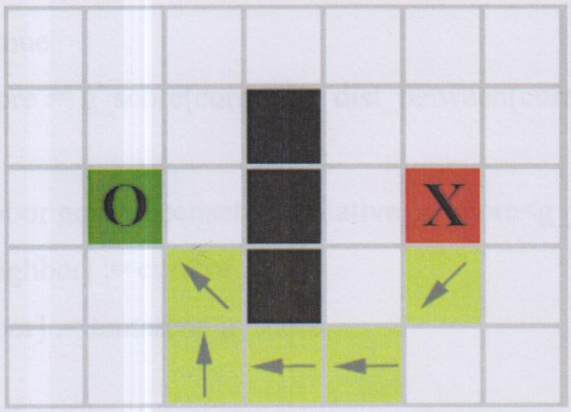


Figure 2.9 Constructing Final Path in A* Search

2.3.2 Pseudo Code for A* algorithm

```
function A*(start,goal)
closedset := the empty set    // The set of nodes already evaluated.
openset := {start}    // The set of tentative nodes to be evaluated,
initially containing the start node
came_from := the empty map    // The map of navigated nodes.

g_score[start] := 0    // Cost from start along best known path.
  // Estimated total cost from start to goal through y.
f_score[start] := g_score[start] + heuristic_cost_estimate(start, goal)

  while openset is not empty
     current := the node in openset having the lowest f_score[] value
     if current = goal
        return reconstruct_path(came_from, goal)

     remove current from openset
     add current to closedset
     for each neighbor in neighbor_nodes(current)
        if neighbor in closedset
           continue
tentative_g_score := g_score[current] + dist_between(current,neighbor)

        if neighbor not in openset or tentative_g_score<g_score[neighbor]
came_from[neighbor] := current
g_score[neighbor] := tentative_g_score

f_score[neighbor] := g_score[neighbor] +
```

```
heuristic_cost_estimate(neighbor, goal)
        if neighbor not in openset
            add neighbor to openset

    return failure

function reconstruct_path(came_from, current_node)
    if current_node in came_from
        p := reconstruct_path(came_from, came_from[current_node])
        return (p + current_node)
    else
        return current_node
```

Figure 2.10 Pseudo Code for A* algorithm

In road networks it is getting more important to find the way to the destination point. If a person is new to a place, much time can be wasted until finding the destination. There exist some products developed to overcome this difficulty, providing a map of the region. After entering the starting point and the final location, it is possible to get the shortest path. Experimental studies are performed to select the best algorithm for using in path finding process. It can be compared the shortest path algorithms by doing experimental analysis on a big database [14]. They evaluated the Dijkstra, A*and modified A* algorithms with respect to the computation time on real networks, solution quality, ease of implementation and the extensibility of the algorithm. They found out that A* algorithm has better time efficiency. They have showed that A* Search is also much

more efficient in terms of the number of nodes visited. Dijkstra's Algorithm searches much more nodes than A* does.

A brief overview of the optimal shortest path algorithms, which is often considered as the starting point for the development of many heuristic shortest path algorithms [16]. It examines four heuristic search strategies: (i) limit the area searched, (ii) decompose the search problem, (iii) limit the links searched, and (iv) some combination of above. The review summarizes the distinguishing idea of each search strategy and its algorithmic variations.

The strategic planning process in public and rail transport is usually divided into consecutive steps of network design, line planning, and timetabling. Operations research methods can support the planning decisions in each of these steps, The line planning problem (LPP) in public transport to design line routes and their frequencies in a given street or track network such that a given transportation volume, given by a so-called origin-destination matrix (OD-matrix). The frequency of a line is supposed to indicate a basic timetable period and controls the lines' transportation capacity. There are two competing objectives: on the one hand to minimize the operating costs of lines and on the other hand to minimize user discomfort. User discomfort is usually measured by the total passenger traveling time or the number of transfers during the ride, or both.

The recent literature on the LPP mainly deals with railway networks. One common assumption is the so-called system split, which fixes the traveling paths of the passengers before the lines are known. A second common assumption is that an optimal line plan can be chosen from a (small) pre-computed set of lines. Third, maximization of direct travelers, i.e., travelers without transfers, is sometimes considered as the objective. In such an approach, transfer waiting times do not play a role.

A new multi-commodity flow model for the LPP is described in [4]. The model minimizes a combination of total passenger traveling time and operating costs. It generates lines dynamically, handles frequencies implicitly by means of continuous frequency variables, and allows passengers to change their routes according to the computed line system; in particular, we do not assume a system split. These properties aim at line planning scenarios in public transport, where we see less justification for a system split and fewer restrictions in line design than one seem to have in railway line planning.

The first approaches to the line planning problem had the idea to assemble lines from shorter pieces in an iterative (and often interactive) process. An early example is the so-called skeleton method described in [23], that chooses the endpoints of a route and several intermediate nodes which are then joined by shortest paths with respect to length or traveling time.

In a similar way, it constructed lines by adjoining small pieces of streets/tracks in order to maximize the number of direct travelers [24]. Another branch of the literature considers two-step approaches that pre compute some set of lines in a first phase and choose a line plan from this set in a second phase [19].

Path finding using a two-level hierarchy is described in [20]. The author provides only a high-level presentation of the approach. The problem map is abstracted into clusters such as rooms in a building or square blocks on a field. An abstract action crosses a room from the middle of an entrance to another. This method has similarities to our work. First, both approaches partition the problem map into clusters such as square blocks. Second, abstract actions are block crossings (as opposed to going from one block center to another block center).Third, both techniques abstract a block entrance into one transition point (in fact, we

allow either one or two points). This leads to fast computation but gives up the solution optimality. There are also significant differences between the two approaches. It extend our hierarchy to several abstraction levels and do this abstraction in a domain independent way. We also pre-compute and cache optimal distances for block crossing, reducing the costs of the on-line computation.

Another important hierarchical approach for path finding in commercial games uses *points of visibility* [21]. This method exploits the domain local topology to define an abstract graph that covers the map efficiently. The graph nodes represent the corners of convex obstacles. For each node, edges are added to all the nodes that can be seen from the current node (i.e., the can be connected with a straight line).This method provides solutions of good quality. It is particularly useful when the number of obstacles is relatively small and they have a convex polygonal shape (i.e., building interiors). The efficiency of the method decreases when many obstacles are present and/or their shape is not a convex polygon. Consider the case of a map containing a forest, which is a dense collection of small size obstacles. Modeling such a topology with points of visibility would result in a large graph (in terms of number of nodes and edges) with short edges. Therefore, the key idea of traveling long distances in a single step wouldn't be efficiently exploited. When the problem map contains concave or curved shapes, the method either has poor performance or needs sophisticated engineering to build the graph efficiently. In fact, the need for algorithmic or designer assistance to create the graph is one of the disadvantages of the method. In contrast, our approach works for many kinds of maps and does not require complex domain analysis to perform the abstraction.

2.4 Related Works

The 21st century has witnessed dramatic sophistications of internet business models and its related services such as e-shopping. E-shopping identified as the process of buying goods and services directly over the internet which considered as a form of e-commerce transactions. Since the internet is the only medium for e-shopping, it became one of the most attractive facilities for internet users where consumers place an online orders and the retailer is responsible for fulfilling their orders. It identified the process of order fulfilling as the process of planning, organizing and dispatching consumers' orders and prepare them to be delivered to the consumer's doorstep or any other delivery location [25]. Home delivery service is one of the most important services which play a crucial role in the success of e-shopping. Physical distribution of items bought online must be operated by the seller's delivery fleet or by third-party logistics (3PL) Company [5]. Therefore, retailers/3PL must meet their consumers' expectations by having a reliable and efficient delivery system that fills their needs in order to gain their trust and satisfaction [9]. Home delivery service also called "last mile" which play an important role in the preservation of the environment by generating less CO2 in comparison with conventional shopping and reducing energy consumption by reducing consumers trips into shops for shopping or collecting their items. In addition, it reduces the impact of traffic by delivering consumers' orders using one vehicle within specific time windows [6]. However, the development of the delivery service needs a developed addressing system to enable the consumers to provide their shipping address details to the delivery couriers to be able find this address. This system is called "postcode or ZIP code system" which consists of series of letters and digits to give each house a unique

functions in the toolbox. Three related membership functions are the Z, S, and Pi curves, all named because of their shape. The function zmf is the asymmetrical polynomial curve open to the left; smf is the mirror-image function that opens to the right, and pimf is zero on both extremes with a rise in the middle.

2.3 Neuro-Fuzzy Design

The Artificial Neural Networks (ANNS) have been very successful in recognizing nonlinear patterns from noisy, high frequency data and have been very useful forecasting tools. However, there has been criticism of their inability to transparently explain how a decision (forecast) is reached. Also, unlike fuzzy logic it is impossible to incorporate *a priori* information about the problem into the ANN system. Fuzzy logic can quantify vague information and produce transparent decision-making logic. The drawbacks of fuzzy logic are the lack of a learning capability and the necessity for an expert knowledge about the system.

Neural fuzzy inference systems introduce a parallel architecture and learning capability to a fuzzy inference system. Each fuzzy rule is created using the ANN and it is a data driven process. Fuzzy neural networks embed fuzzy logic into the ANN by fuzzifying the learning algorithms. This system uses a neuro-fuzzy combination where the ANN is used for identification (forecasting) and FLC extracts the decision from the ANN's output.

Finding a trading rule with FLC should not be confused with training of an ANN. An ANN uses a learning algorithm to map input variables (e.g. lagged interest rate, lagged order flow) into output variables (e.g. exchange rate change). In other words, an ANN is a

2.4 Related Works

The 21st century has witnessed dramatic sophistications of internet business models and its related services such as e-shopping. E-shopping identified as the process of buying goods and services directly over the internet which considered as a form of e-commerce transactions. Since the internet is the only medium for e-shopping, it became one of the most attractive facilities for internet users where consumers place an online orders and the retailer is responsible for fulfilling their orders. It identified the process of order fulfilling as the process of planning, organizing and dispatching consumers' orders and prepare them to be delivered to the consumer's doorstep or any other delivery location [25]. Home delivery service is one of the most important services which play a crucial role in the success of e-shopping. Physical distribution of items bought online must be operated by the seller's delivery fleet or by third-party logistics (3PL) Company [5]. Therefore, retailers/3PL must meet their consumers' expectations by having a reliable and efficient delivery system that fills their needs in order to gain their trust and satisfaction [9]. Home delivery service also called "last mile" which play an important role in the preservation of the environment by generating less CO2 in comparison with conventional shopping and reducing energy consumption by reducing consumers trips into shops for shopping or collecting their items. In addition, it reduces the impact of traffic by delivering consumers' orders using one vehicle within specific time windows [6]. However, the development of the delivery service needs a developed addressing system to enable the consumers to provide their shipping address details to the delivery couriers to be able find this address. This system is called "postcode or ZIP code system" which consists of series of letters and digits to give each house a unique

identifier for the purpose of mail sorting [10]. Thus, retailers and 3PL will be able to plan their deliveries routes to the customers' house location which lead to reduce GHG emissions and energy consumption. This study aims to design a system that will improve the delivery service in Jordan and the other developing countries. The system is based on using positioning technologies such as web GIS and GPS as an application for desktop and Smartphone platforms. In addition, the system will help in saving energy and reduce greenhouse gas (GHG) emission in these countries.

E-shopping is an Internet application that has spread rapidly in the developed countries, but whose progress has been markedly slower in developing countries, due to some barriers. These barriers lie in infrastructural barriers (e.g. information technology hardware, Internet access, and internet bandwidth), cultural and social barriers (e.g. high uncertainty avoidance), and lack of required technologies to apply such transaction models (e.g. postcode system, delivery system, and financial system) [18]. E-shopping is defined as buying a basket of commodities and its related services (e.g. delivery service) over the Internet [7]. According to Shergill and Chen (2005), customers can be attracted to shopping websites depending on its commodity value, quality of service (QoS) and customer service, convenience, experience of using e-shopping websites, payment security and privacy, and finally home delivery service availability [12]. These factors affect customers' behaviour when shopping online. However, Jordan is one of the developing countries which faces problems affect retailers and consumers decision to adopt e-shopping phenomenon due to its less popularity among them. In the last few years, Jordan has witnessed improvements and developments in the information and communication technology (ICT) sector which offer a developed ICT infrastructure to reach the required level of ICT readiness

to start initiating e-businesses [2]. ICT readiness has been identified by the ministry of information and communication technologies (MICTs) in Jordan as the developments of ICT infrastructure which attract the community to benefit from these developments [18]. On the one hand, Internet service providers (ISPs) number in Jordan has been increased which added more improvements on the internet services offered by those providers (e.g. WiMAX and Wi-Fi) with lower subscription prices. On the other hand, internet penetration has been increased by the users since smart phones being connected to the internet and used in daily life activities [1]. These developments led to design the proposed system in order to improve e-shopping and home delivery service. Then, reduce GHG emissions by reducing traffic and energy consumption.

Jordan post company (JPC) provides a variety of services to the citizens, such as postal services (e.g. private post boxes, EMS and parcel mail service), financial services (e.g. bills collection, money order, etc.), E-services (e.g. bills payment, P.O. Box rent or renew, etc.), SMS services and finally ancillary services [8]. In addition, JPC provides the postal and financial services on behalf of institutions, departments and companies in the public and private sectors by collection of invoices as well as telegram and phone booth services [15]. However, JPC do not provide home delivery service for customers due to the lack of postcode system which leads to the lack of efficient and reliable delivery service system in the kingdom. Therefore, JPC is still providing poor delivery service quality especially in rural and remote areas, and the express delivery service is supplied by foreign couriers such as TNT, ARAMEX, DHL and UPS. Even those couriers do not have a delivery system for delivering customers' orders where house location detected by phone and depending on the driver's experience in the delivery area which increases energy consumption and thus lead to increase in GHG emissions.

E-shopping home delivery service in Jordan was chosen as a case study because of the potential impact of e-shopping for reducing GHG emissions. The current imperfection of e-shopping home delivery service in Jordan has affected customers' and retailers' decision to adopt such services. In addition, this imperfection increased traffic and energy consumption which led to increase GHG emissions. However, the need for a successful home delivery model will improve e-shopping and its services, and influence consumers and retailers positively toward adopting e-shopping. Thus, the positive influence of consumers' behaviour could influence the whole supply chain and reduce GHG emissions [22]. Also, the use of positioning technologies play a significant role in reducing energy consumption and traffic which leads to lower GHG emissions [3].

The main purpose was to solve the problem of lack of postcode system in Jordan in order to improve the delivery service of online purchased goods which lead to improvements in e-shopping its self [11]. Thus, customers' and retailers' attitudes toward e-shopping will be affected positively toward adopting such businesses as a channel for selling and buying goods online instead of traditional shopping. However, Adoption of e-shopping reduces traffic and energy consumption and then lead to lower CO2 emissions which lead to greener environment. The preliminary findings of the distributed survey indicate that most of the results are significant, which means that the proposed system solves the problem of home delivery service for online purchased goods and affect customers' and retailers' attitudes toward such businesses. In addition, the system could help in reducing GHG emissions caused by customers' travel for conventional shopping.

This study aims to design a system to be used as an alternative for postcode system to enable e-shopping and improve the delivery service in

Jordan. The system is based on using state-of-art positioning technologies such as GIS and GPS. These technologies have a significant role in reducing greenhouse gas (GHG) emissions by reducing traffic and energy consumption. The designed system has been tested by distributing a questionnaire among 37 participants (retailer employees, delivery logistics employees, and university students & academic staff) whereas the collected data were analyzed using SPSS. The findings indicate that there are statistically significant role.

CHATPER 3
SYSTEM DESIGN

3.1 System Flow

The main purpose of the system is to provide a product delivery system for online shopping market that can apply in Mandalay Division. Two sides are considered in this system: Administrator site and Customer site. Before starting two sides, the system execute preprocessing step that is shown in Figure 3.1. In this figure, the system convert Mandalay Division map into "Map" file. And then calculate estimate distance between cities by using Dijkstra's Algorithm and stores these distances into "Estimated distance" file.

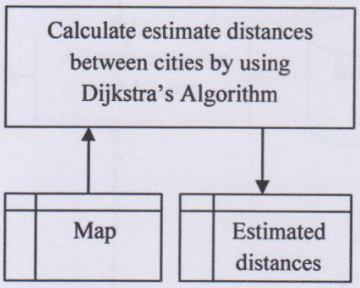


Figure 3.1 Preprocessing Step of the System

At the Administrator site, the system arranges for the administrator to add new branch and to view delivery paths for each branch. If the administrator chooses adding new branch, the system retrieves all cities of Mandalay Division and allows the administrator to choose a city that is he wants to set as a new branch and stores this city into "Branch"

database. Otherwise, the system provides viewing paths for each branch. In this process the system loads current branches of the system from the "Branch" database and allows the administrator to choose a branch. After choosing the branch, the system retrieves all order paths from the database and groups the same order paths because order group is applicable for delivery system.

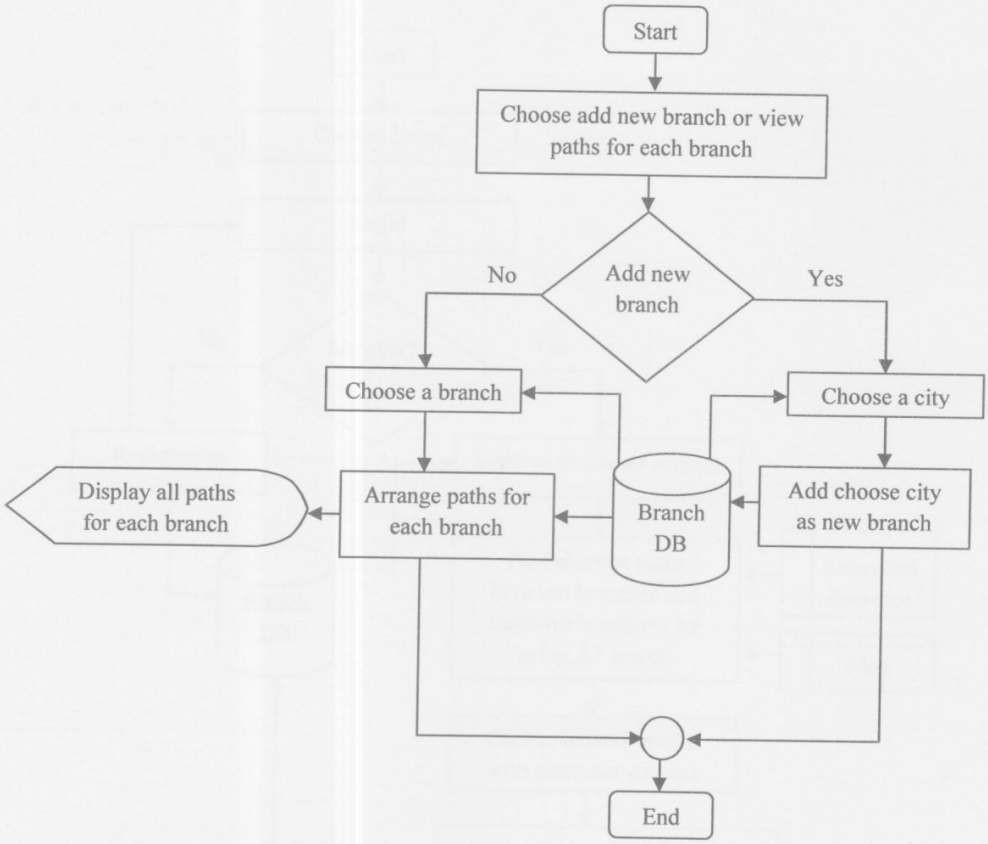


Figure 3.2 System Flow Diagram for Administrator

At the Customer site, the system shows items information and allows the customer to choose items. After choosing item, the system requests the customer to provide login and checks this customer is

member or not. If the customer is not a member, the system does not accept customer order. So the system prepares for member registration. If the customer is member, the system retrieves customer's address from the database and finds shortest paths between branches and customer's address by using A* search. And then the system finds the lowest distance from the customer's address among shortest paths. After defining the nearest branch, the system stores order items, branch name and shortest path into the database.

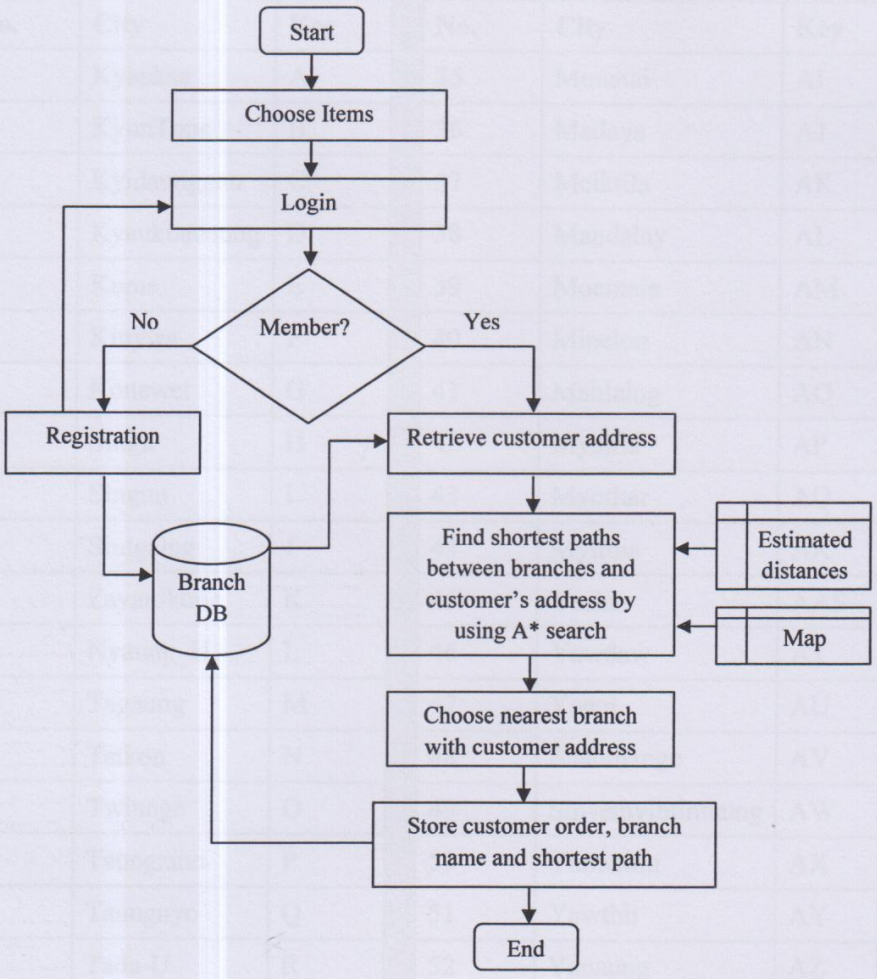


Figure 3.3 System Flow Diagram for Customer

3.2 Finding Estimate Distances using Dijkstra's Algorithm

Finding shortest path between two nodes using A* search requires estimate distance. In this system, Dijkstra's algorithm is used to find estimate distances between cities of Mandalay Division. Table 3.1 shows the list of cities and the reference keys used in the system. For example, city name "Mandalay" is used "AL" in system calculation.

Table 3.1 Cities of Mandalay Division

No.	City	Key	No.	City	Key
1	Kyaukse	A	35	Monetai	AI
2	KyunTone	B	36	Madaya	AJ
3	Kyidaunggan	C	37	Meiktila	AK
4	Kyaukbadaung	D	38	Mandalay	AL
5	Kume	E	39	Moemate	AM
6	Kinywa	F	40	Minelon	AN
7	Konewet	G	41	Mahlaing	AO
8	Singu	H	42	Myohla	AP
9	Singuu	I	43	Myothar	AQ
10	Sintgaing	J	44	Myittha	AR
11	Zayardkone	K	45	Yesin	AAS
12	Nyaung_U	L	46	Yawdaw	AT
13	Tagaung	M	47	Yaeni	AU
14	Tatkon	N	48	Shanmange	AV
15	Twinnge	O	49	Shwemyintintaung	AW
16	Taungzinn	P	50	Yamethin	AX
17	Taungnyo	Q	51	Yawthit	AY
18	Tada-U	R	52	Yanaung	AZ
19	Taungtha	S	53	Hlawgyi	BA
20	Htonebo	T	54	Latnyoehtoe	BB
21	Natogyi	U	55	Letpanhla	BC

22	Nanpantit	V	56	Latpanchipaw	BD
23	Nabuaing	W	57	Lewe	BE
24	Bagan	X	58	Wundwin	BF
25	Poppa	Y	59	Wabyudaung	BG
26	Pyawbwe	Z	60	Welaung	BH
27	Pyinmana	AA	61	Wetwun	BI
28	Palate	AB	62	Thazi	BJ
29	Patheingyi	AC	63	Thabeikkyin	BK
30	Pyinoolwin	AD	64	Thapyaywa	BL
31	Phayarngarhsu	AE	65	Taphanchaung	BM
32	Mogoke	AF	66	Thawatthi	BN
33	Myingyan	AG	67	Hanmyintmoe	BO
34	Myitnge	AH	68	Innwa	BP

In this section, the system presents how to find estimate distances between cities with example map. Figure 3.4 shows sub map of the Mandalay Division map and the distance between cities are shown with kilometer. To find estimate distance by using Dijkstra's algorithm, the system converts the tree diagram of the input map into adjacency matrix and stores these distances into "Map" file that can be seen in Table 3.2.

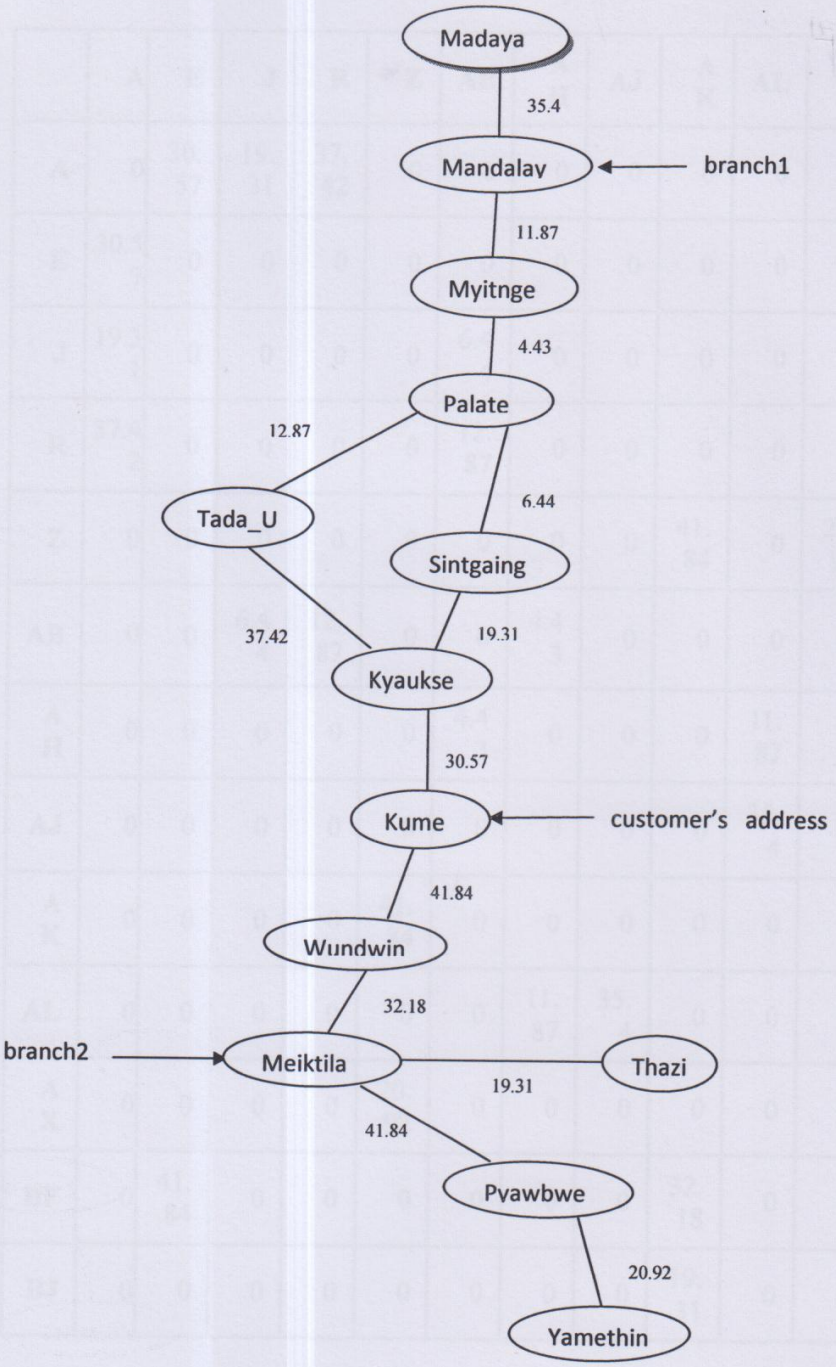


Figure 3.4 Sub Map of the Mandalay Division Map

Table 3.2Map File

	A	E	J	R	Z	AB	AH	AJ	AK	AL	AX	BF	BJ
A	0	30.57	19.31	37.42	0	0	0	0	0	0	0	0	0
E	30.57	0	0	0	0	0	0	0	0	0	0	41.84	0
J	19.31	0	0	0	0	6.44	0	0	0	0	0	0	0
R	37.42	0	0	0	0	12.87	0	0	0	0	0	0	0
Z	0	0	0	0	0	0	0	0	41.84	0	20.92	0	0
AB	0	0	6.44	12.87	0	0	4.43	0	0	0	0	0	0
AH	0	0	0	0	0	4.43	0	0	0	11.87	0	0	0
AJ	0	0	0	0	0	0	0	0	0	35.4	0	0	0
AK	0	0	0	0	41.84	0	0	0	0	0	0	32.18	19.31
AL	0	0	0	0	0	0	11.87	35.4	0	0	0	0	0
AX	0	0	0	0	20.92	0	0	0	0	0	0	0	0
BF	0	41.84	0	0	0	0	0	0	32.18	0	0	0	0
BJ	0	0	0	0	0	0	0	0	19.31	0	0	0	0

After transformation into adjacency matrix ("Map" file), Figure 3.5 shows step by step process of finding minimum distances that can reach from city 'A' to other cities by using Dijkstra's algorithm and estimated distances for all cities ("Estimated distances " file) can be seen in Table 3.3.

Initial Priority Queue:

v	**A**	E	J	R	Z	AB	AH	AJ	AK	AL	AX	BF	BJ
d[v]	**0**	α	α	α	α	α	α	α	α	α	α	α	α

v	A	E	J	R	Z	AB	AH	AJ	AK	AL	AX	BF	BJ
d[v]	0	30.57	19.31	37.42	α	α	α	α	α	α	α	α	α
pred[v]	nil	A	A	A	nil	nil	nil	nil	nil	nil	nil	nil	nil

Priority Queue:

v	E	J	R	Z	AB	AH	AJ	AK	AL	AX	BF	BJ
d[v]	30.57	19.31	37.42	α	α	α	α	α	α	α	α	α

v	A	E	J	R	Z	AB	AH	AJ	AK	AL	AX	BF	BJ
d[v]	0	30.57	19.31	37.42	α	25.75	α	α	α	α	α	α	α
pred [v]	nil	A	A	A	nil	J	nil	nil	nil	nil	nil	nil	nil

Priority Queue:

v	E	R	Z	AB	AH	AJ	AK	AL	AX	BF	BJ
d[v]	30.57	37.42	α	25.75	α	α	α	α	α	α	α

v	A	E	J	R	Z	AB	AH	AJ	AK	AL	AX	BF	BJ
d[v]	0	30.57	19.31	37.42	α	25.75	30.18	α	α	α	α	α	α
pred [v]	nil	A	A	A	nil	J	AB	nil	nil	nil	nil	nil	nil

Priority Queue:

v	E	R	Z	AH	AJ	AK	AL	AX	BF	BJ
d[v]	30.57	37.42	α	30.18	α	α	α	α	α	α

v	A	E	J	R	Z	AB	AH	AJ	AK	AL	AX	BF	BJ
d[v]	0	30.57	19.31	37.42	α	25.75	30.18	α	α	42.05	α	α	α
pred [v]	nil	A	A	A	nil	J	AB	nil	nil	AH	nil	nil	nil

Priority Queue:

v	E	R	Z	AJ	AK	AL	AX	BF	BJ
d[v]	30.57	37.42	α	α	α	42.05	α	α	α

v	A	E	J	R	Z	AB	AH	AJ	AK	AL	AX	BF	BJ
d[v]	0	30.57	19.31	37.42	α	25.75	30.18	α	α	42.05	α	72.41	α
pred [v]	nil	A	A	A	nil	J	AB	nil	nil	AH	nil	E	nil

Priority Queue:

v	R	Z	AJ	AK	AL	AX	BF	BJ
d[v]	37.42	α	α	α	42.05	α	72.41	α

v	A	E	J	R	Z	AB	AH	AJ	AK	AL	AX	BF	BJ
d[v]	0	30.57	19.31	37.42	α	25.75	30.18	α	α	42.05	α	72.41	α
pred [v]	nil	A	A	A	nil	J	AB	nil	nil	AH	nil	E	nil

Priority Queue:

v	Z	AJ	AK	AL	AX	BF	BJ
d[v]	α	α	α	42.05	α	72.41	α

v	A	E	J	R	Z	AB	AH	AJ	AK	AL	AX	BF	BJ
d[v]	0	30.57	19.31	37.42	α	25.75	30.18	77.45	α	42.05	α	72.41	α
pred[v]	nil	A	A	A	nil	J	AB	AL	nil	AH	nil	E	nil

Priority Queue:

v	Z	AJ	AK	AX	BF	BJ
d[v]	α	77.45	α	α	72.41	α

v	A	E	J	R	Z	AB	AH	AJ	AK	AL	AX	BF	BJ
d[v]	0	30.57	19.31	37.42	α	25.75	30.18	77.45	104.59	42.05	α	72.41	α
pred[v]	nil	A	A	A	nil	J	AB	AL	BF	AH	nil	E	nil

Priority Queue:

v	Z	AK	AX	BJ
d[v]	α	104.59	α	α

v	A	E	J	R	Z	AB	AH	AJ	AK	AL	AX	BF	BJ
d[v]	0	30.57	19.31	37.42	146.43	25.75	30.18	77.45	104.59	42.05	α	72.41	123.9
pred[v]	nil	A	A	A	AK	J	AB	AL	BF	AH	nil	E	AK

Priority Queue:

v	Z	AX	BJ
d[v]	146.43	α	123.9

v	A	E	J	R	Z	AB	AH	AJ	AK	AL	AX	BF	BJ
d[v]	0	30.57	19.31	37.42	146.43	25.75	30.18	77.45	104.59	42.05	α	72.41	123.9
pred[v]	nil	A	A	A	AK	J	AB	AL	BF	AH	nil	E	AK

Priority Queue:

v	Z	AX
d[v]	146.43	α

v	A	E	J	R	Z	AB	AH	AJ	AK	AL	AX	BF	BJ
d[v]	0	30.57	19.31	37.42	146.43	25.75	30.18	77.45	104.59	42.05	167.35	72.41	123.9
pred[v]	nil	A	A	A	AK	J	AB	AL	BF	AH	Z	E	AK

Priority Queue:

v	AX
d[v]	167.35

v	A	E	J	R	Z	AB	AH	AJ	AK	AL	AX	BF	BJ
d[v]	0	30.57	19.31	37.42	146.43	25.75	30.18	77.45	104.59	42.05	167.35	72.41	123.9
pred[v]	nil	A	A	A	AK	J	AB	AL	BF	AH	Z	E	AK

Priority Queue: $Q = \Phi$

Figure 3.5 Step by Step Calculation of Dijkstra's algorithm

Table 3.3Estimated Distance File

	A	E	J	R	Z	AB	AH	AJ	AK	AL	AX	BF	BJ
A	0.00	30.57	19.31	37.42	146.43	25.75	30.18	77.45	104.59	42.05	167.35	72.41	123.90
E	30.57	0.00	49.88	67.99	115.86	56.32	60.75	108.02	74.02	72.62	136.78	41.84	93.33

J	19.31	49.88	0.00	19.31	165.74	6.44	10.87	58.14	123.90	22.74	186.66	91.72	143.21
R	37.42	67.99	19.31	0.00	183.85	12.87	17.30	64.57	142.01	29.17	204.77	109.83	161.32
Z	146.43	115.86	165.74	183.85	0.00	172.18	176.61	223.88	41.84	188.48	20.92	74.02	61.15
AB	25.75	56.32	6.44	12.87	172.18	0.00	4.43	51.70	130.34	16.30	193.10	98.16	149.65
AH	30.18	60.75	10.87	17.30	176.61	4.43	0.00	47.27	134.77	11.87	197.53	102.59	154.08
AJ	77.45	108.02	58.14	64.57	223.88	51.70	47.27	0.00	182.04	35.40	244.80	149.86	201.35
AK	104.59	74.02	123.90	142.01	41.84	130.34	134.77	182.04	0.00	146.64	62.76	32.18	19.31
AL	42.05	72.62	22.74	29.17	188.48	16.30	11.87	35.40	146.64	0.00	209.40	114.46	165.95
AX	167.35	136.78	186.66	204.77	20.92	193.10	197.53	244.80	62.76	209.40	0.00	94.94	82.07
BF	72.41	41.84	91.72	109.83	74.02	98.16	102.59	149.86	32.18	114.46	94.94	0.00	51.49
BJ	123.90	93.33	143.21	161.32	61.15	149.65	154.08	201.35	19.31	165.95	82.07	51.49	0.00

3.3 Finding Shortest Path using A* Search

In this example, the system assumes node AL(Mandalay) and node AK(Meiktila) as the system branches and node E(Kume) as customer's address. The system assumes AL(Mandalay) as nearest branch because

AL(Mandalay) is the minimum estimated distance among two branches. So, the system calculates shortest path from node AL(Mandalay) to node E(Kume) by using A* search. If the system finds shortest path, the system uses Table 3.2 and Table 3.4. Table 3.2 shows a map file for actual distance from a node to another node and Table 3.4 shows estimated distances from other nodes to E of sample graph. In figure3.3, the system's branch node AL(Mandalay) have two paths. The two paths are from node AL(Mandalay) to AJ(Madaya) and from node AL(Mandalay) to AH(Myitnge). So, the system finds shortest path for two paths. Firstly, the system finds shortest path from AL(Mandalay) to AJ(Madaya). Thus, the system must sum the cost of actual distance from Table 3.2 from AL(Mandalay) to AJ(Madaya) is 35.4 and the cost of estimated distance from Table 3.4 from AJ(Madaya) to E(Kume) is108.02. So, the cost of distance from AL(Mandalay) to AJ(Madaya) is 143.42. And then, the system finds shortest path from AL(Mandalay) to AH(Myitnge). Thus, the system must sum the cost of actual distance from Table 3.2 from AL(Mandalay) to AH(Myitnge) is 11.87 and the cost of estimated distance from Table 3.4 from AH(Myitnge) to E(Kume) is 60.75. So, the cost of distance from AL(Mandalay) to AH(Myitnge) is 72.62. In this two path, the cost AL(Mandalay) to AH(Myitnge) is less than the cost of AL(Mandalay) to AJ(Madaya). So, the system choose path from AL(Mandalay) to AH(Myitnge). And then, the system must go path from AL (Mandalay) to AH(Myitnge) to AB(Palate). The system must sum the cost of actual distance from Table 3.2 from AL(Mandalay) to AH(Myitnge) is 11.87, from AH(Myitnge) to AB(Palate) is 4.43 and the cost of estimated distance from Table 3.4 from AB(Palate) to E(Kume) is 56.32. So, the cost of distance AL(Mandalay) to AH(Myitnge) to AB(Palate) is 72.62. In this figure node AB(Palate) have two paths. This two paths is from node AL(Mandalay) to AH(Myitnge) to AB(Palate) to

J(Sintgaing) and AL(Mandalay) to AH(Myitnge) to AB(Palate) to R(Tada_U). Firstly, the system find shortest path AL(Mandalay) to AH(Myitnge) to AB(Palate) to J(Sintgaing). Thus, the system must sum the cost of actual distance from Table 3.2, from AL(Mandalay) to AH(Myitnge) is 11.87, from AH(Myitnge) to AB(Palate) is 4.43, from AB(Palate) to J(Sintgaing) is 6.44 and the cost of estimated distance from Table 3.4 from J(Sintgaing) to E(Kume) is 49.88. So, the cost of distance AL(Mandalay) to AH(Myitnge) to AB(Palate) to J(Sintgaing) is 72.62. And then, the system find shortest path from AL(Mandalay) to AH(Myitnge) to AB(Palate) to R(Tada_U). Thus, the system must sum the cost of actual distance from Table 3.2 from AL(Mandalay) to AH(Myitnge) is 11.87, from AH(Myitnge) to AB(Palate) is 4.43, from AB(Palate) to R(Tada_U) is 12.87 and the cost of estimated distance from Table 3.4 R(Tada_U) to E(Kume) is 67.99. So, the cost of distance AL(Mandalay) to AH(Myitnge) to AB(Palate) to R(Tada_U) is 97.16. In this two paths, the cost AL(Mandalay) to AH(Myitnge) to AB(Palate) to J(Sintgaing) is less than AL(Mandalay) to AH(Myitnge) to AB(Palate) to R(Tada_U). So, the system choose path from AL(Mandalay) to AH(Myitnge) to AB(Palate) to J(Sintgaing). And then, the system must go path from AL(Mandalay) to AH(Myitnge) to AB(Palate) to J(Sintgaing) to A(Kyaukse). Thus, the system must sum the cost of actual distance from Table 3.2 from AL(Mandalay) to AH(Myitnge) is 11.87, from AH(Myitnge) to AB(Palate) is 4.43, from AB(Palate) to J(Sintgaing) is 6.44, from J(Sintgaing) to A(Kyaukse) is 19.31 and the cost of estimated distance from Table 3.4 A(Kyaukse) to E(Kume) is 30.57. So, the cost of distance AL(Mandalay) to AH(Myitnge) to AB(Palate) to J(Sintgaing) to A(Kyaukse) is 72.62. And then, the system must go path from AL(Mandalay) to AH(Myitnge) to AB(Palate) to J(Sintgaing) to A(Kyaukse) to E(Kume). Thus, the system must sum the

cost of actual distance from Table 3.2 from AL(Mandalay) to AH (Myitnge) is 11.87, from AH(Myitnge) to AB(Palate) is 4.43, from AB(Palate) to J(Sintgaing) is 6.44, from J(Sintgaing) to A(Kyaukse) is 19.31, from A(Kyaukse) to E(Kume) is 30.57 and the cost of estimated distance from Table 3.4 E(Kume) to E(Kume) is 0. So, the cost of distance AL(Mandalay) to AH(Myitnge) to AB(Palate) to J(Sintgaing) to A(Kyaukse) to E(Kume) is 72.62. Finally, the system reach to customer' address E(Kume). So, the system must go to reach E(Kume) from AL(Mandalay) to AH(Myitnge) to AB(Palate) to J(Sintgaing) to A(Kyaukse) to E(Kume). Thus, the shortest path from AL(Mandalay) to E(Kume) is from AL(Mandalay) to AH(Myitnge) to AB(Palate) to J(Sintgaing) to A(Kyaukse) to E(Kume).

Table 3.4 Estimated Distances from E

State	h
A	30.57
E	0.00
J	49.88
R	67.99
Z	115.86
AB	56.32
AH	60.75
AJ	108.02
AK	74.02
AL	72.62
AX	136.78
BF	41.84
BJ	93.33

{[AL->AJ, 35.4+108.02], [AL->AH, 11.87+60.75]}

{[AL->AJ, 143.42], [AL->AH, 72.62]}

{[AL->AH->AB, (11.87+4.43)+56.32]}

{[AL->AH->AB, 16.30 +56.32]}

{[AL->AH->AB, 72.62]}

{[AL->AH->AB->J, (11.87+4.43+6.44) +49.88], [AL->AH->AB->R (11.87+4.43+12.87) +67.99]}

{[AL->AH->AB->J, 72.62], [AL->AH->AB->R, 97.16]}

{[AL->AH->AB->J ->A, (11.87+4.43+6.44+19.31) +30.57]}

{[AL->AH->AB->J ->A, 72.62]}

{[AL->AH->AB->J ->A->E, (11.87+4.43+6.44+19.31+30.57) +0]}

{[AL->AH->AB->J ->A->E, 72.62]}

So, The Shortest path from AL(Mandalay) to E(Kume) is AL(Mandalay) to AH(Myitnge) to AB(Palate) to J(Sintgaing) to A(Kyaukse) to E(Kume) = 72.62 km

3.4 Arrange Paths

Arranging paths is also the import part of the delivery system. For each branch, this system will arrange all paths by finding the same ways. The following is the example of arrange paths. In this example, the system has three orders for AL(Mandalay) branch. In these orders, the path of the order1 contains in the path of order3. So the system assumes these two orders are the same direction and defines as Group1. Order3 has different direction from order1 and order2. So the system defines this path as Group2.

order1→ AL, AH, AB, J

order2→ AL, AH, AB, R

order3→ AL, AH, AB, J, A, E

Results:

Group1

order1→ AL, AH, AB, J

order3→ AL, AH, AB, J, A, E

Group2

order2→ AL, AH, AB, R

CHAPTER 4

IMPLEMENTATION OF THE SYSTEM

On-line shopping is often seen as an application that can potentially play an important role in economic activity. Moreover, item delivery also important in the online shopping system. Home delivery service is an essential service for online purchased goods. The need for reliable delivery system becomes necessary. So, this system is implemented online shopping based home delivery system by using J2EE, MySQL 5.0 and Microsoft Excel.

4.1 Implementation

This home delivery system is implemented based on online shopping system. In this implementation, two main sides are considered (customer side and administrator side). So, interface design for the customer is the same as online shopping design. Basic operations of online shopping (adding new items and checking remain items) are not considered in the system because the main purpose of this system to implement a home delivery system. So, the system is mainly considered to find nearest branch with customer address and home delivery path and showed customer home page with a little item information.

In customer home page, the system designs add to card of online shopping. So, the customer can choose their require items and can add the number of quantities for selected item into shopping basket. If the customer wants to change the number of quantities in the shopping basket, the system allows updating the item quantities. The system also allows removing selected items from the shopping basket. The system shows the number of quantities in the shopping cart at the top of the page.

The system arranges "Finish Shopping" link to press after finish shopping. The system only accepts orders from the member of the system. So, the system shows login page after pressing this link. In this login page, the system requests email and password from the customer to check this customer is already member of the system or not. If the customer is not a member, the system allows the customer to register the system by pressing "Registration" link.

In the registration page, the customer must fill all blanks. The error message will show if the customer miss at least one blank. If the customer is already member, the system retrieves the customer's address from the database and calculates the shortest paths from all branches by using A* search algorithm. And then, choose the branch with shortest distance among the distances. After finding the nearest branch from the customer address, the system shows shopping cart content and records these order information to the customer.

In the administrator home page, the system arranges for the administrator to add new branch and to view delivery list. If the administrator wants to add new branch, the system will shows all cities of Mandalay Division excepting existing branches names. The administrator must choose a city that he wants to set a new branch. The system shows successful message after inserting selected city as a new branch.

The system allows the administrator to view delivery information. Therefore, the administrator must choose a branch name and order date to view delivery information. The system collects all order information for selected branch from the database. And then, classified orders if the orders address are the same ways.

4.2 Sequence Diagram

This section discusses the process of the system with two sequence diagram. The first diagram shows the process of the system for the customer and the second diagram for administrator. The diagrams can be seen in Figure 4.1 and 4.2 respectively.

In Figure 4.1, the system sends item information to "AddToCart" class when the customer adding/ removing/ updating item. . After accepting item information, the system do login process. To do login process the system requests email and password from the customer and calls "CheckMember" function that will check customer email and password in the database. If the customer is not a member, the system does registration process and stores customer information in the database. And then, the system do login process again. If the customer is the member, the system finds the nearest branch with the customer address in the "FindBranch" class and sends branch name to the customer.

In the sequence diagram of administrator Figure 4.2, the administrator can do two processes. If the administrator wants to add new branch in the system, the system sends a new branch name to the "NewBranch" class. In this class, the system adds a new branch name in the database and send successful message to administrator. If the administrator wants to view delivery information, the system requests branch name and date from the administrator and send these information into the "DeliveryInformation" class. In this class, the system retrieves delivery information from the database according to branch name and date.

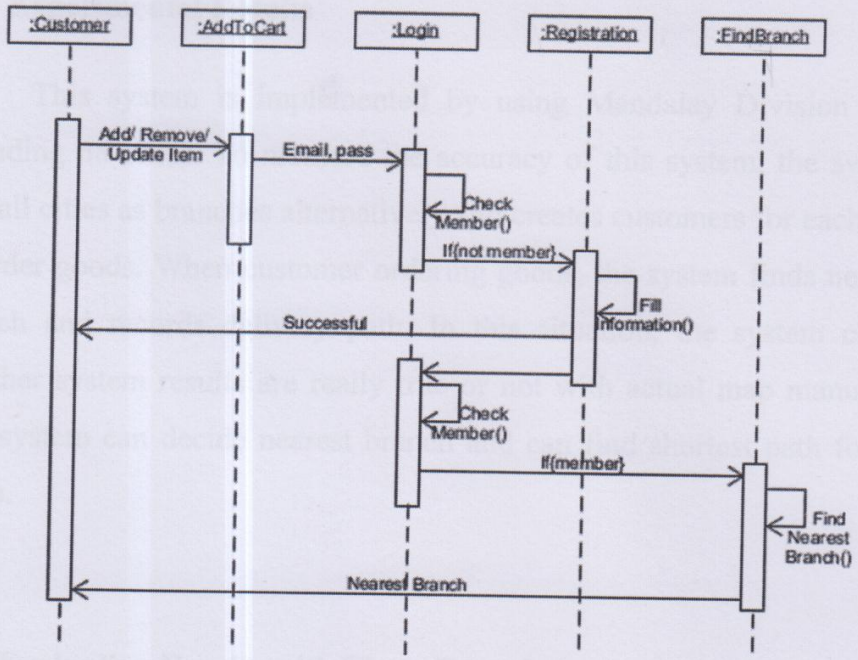


Figure 4.1 Sequence Diagram for Customer

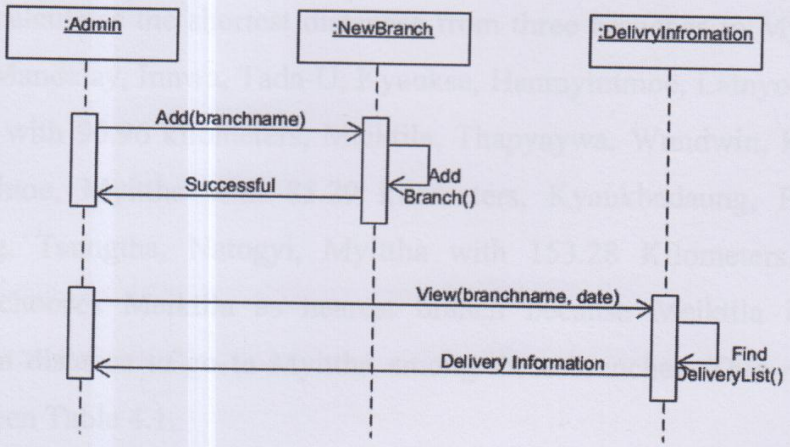


Figure 4.2 Sequence Diagram for Administrator

4.3 Experimental Results

This system is implemented by using Mandalay Division map including 68 cities. To measure the accuracy of this system, the system sets all cities as branches alternatively and creates customers for each city to order goods. When customer ordering goods, the system finds nearest branch and records delivery path. In this situation, the system check whether system results are really true or not with actual map manually. The system can decide nearest branch and can find shortest path for all cities.

4.4 Evaluation Results with Three Branches

In this section, the system is tested with three branches such as Mandalay, Meiktila, Kyaukbadaung and ordered the goods from five cities. For example, a customer ordered the goods form Myittha city. The system calculates the shortest distances from three branches to Myittha. This is Mandalay, Innwa, Tada-U, Kyaukse, Hanmyintmoe, Latnyoehtoe, Myittha with 96.96 kilometers, Meiktila, Thapyaywa, Wundwin, Kume, Latnyoehtoe, Myittha with 85.29 kilometers, Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi, Myittha with 153.28 Kilometers. The system chooses Meiktila as nearest branch because Meiktila is the minimum distance to go to Myittha among three branches. This results can be seen Table 4.1.

Table 4.1 Sample Testing with Three Branches

Branches:Mandalay,Meiktila,Kyaukbadaung			
Customer's City	Path	Distance(km)	Nearest Branch
Myittha	Mandalay, Innwa, Tada-U, Kyaukse, Hanmyintmoe, Latnyoehtoe, Myittha	96.96	Meiktila
	Meiktila, Thapyaywa, Wundwin, Kume, Latnyoehtoe, Myittha	85.29	
	Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi, Myittha	153.28	
Latnyoehtoe	Mandalay, Innwa, Tada-U, Kyaukse, Hanmyintmoe, Latnyoehtoe	88.91	Meiktila
	Meiktila, Thapyaywa, Wundwin, Kume, Latnyoehtoe	77.24	
	Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi, Myittha, Latnyoehtoe	161.33	
Natogyi	Mandalay, Innwa, Tada-U, Kyaukse, Hanmyintmoe, Latnyoehtoe, Myittha, Natogyi	146.85	Meiktila
	Meiktila, Mahlaing, Taungtha, Natogyi	96.96	
	Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi	103.39	
Kume	Mandalay, Innwa, Tada-U, Kyaukse, Hanmyintmoe, Latnyoehtoe, Kume	92.13	Meiktila
	Meiktila, Thapyaywa, Wundwin, Kume	74.02	
	Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi, Myittha, Latnyoehtoe, Kume	164.55	
Hanmyintmoe	Mandalay, Myitnge, Palate, Sintgaing, Kyaukse, Hanmyintmoe	56.53	Mandalay
	Meiktila, Thapyaywa, Wundwin, Kume, Latnyoehtoe, Hanmyintmoe	90.11	
	Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi, Myittha, Latnyoehtoe, Hanmyintmoe	174.22	

4.5 Evaluation Results with Four Branches

In this section, the system is tested with four branches such as Mandalay, Meiktila, Kyaukbadaung, Myingyan and ordered the goods from five cities. For example, a customer ordered the goods from Myittha city. The system calculates the shortest distances from four branches to Myittha. This is Mandalay, Innwa, Tada-U, Kyaukse, Hanmyintmoe, Latnyoehtoe, Myittha with 96.96 kilometers, Meiktila, Thapyaywa, Wundwin, Kume, Latnyoehtoe, Myittha with 85.29 kilometers, Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi, Myittha with

153.28 Kilometers, Myingyan, Natogyi, Myittha with 82.08 kilometers. The system chooses Myingyan as nearest branch because Myingyan is the minimum distance to go to Myittha among four branches. This results can be seen Table 4.2.

Table 4.2 Sample Testing with Four Branches

Branches:Mandalay,Meiktila,Kyaukbadaung,Myingyan			
Customer's City	Path	Distance(km)	Nearest Branch
Myittha	Mandalay, Innwa, Tada-U, Kyaukse, Hanmyintmoe, Latnyoehtoe, Myittha	96.96	Myingyan
	Meiktila, Thapyaywa, Wundwin, Kume, Latnyoehtoe, Myittha	85.29	
	Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi, Myittha	153.28	
	Myingyan, Natogyi, Myittha	82.08	
Latnyoehtoe	Mandalay, Innwa, Tada-U, Kyaukse, Hanmyintmoe, Latnyoehtoe	88.91	Meiktila
	Meiktila, Thapyaywa, Wundwin, Kume, Latnyoehtoe	77.24	
	Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi, Myittha, Latnyoehtoe	161.33	
	Myingyan, Natogyi, Myittha, Latnyoehtoe	90.13	
Natogyi	Mandalay, Innwa, Tada-U, Kyaukse, Hanmyintmoe, Latnyoehtoe, Myittha, Natogyi	146.85	Myingyan
	Meiktila, Mahlaing, Taungtha, Natogyi	96.96	
	Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi	103.39	
	Myingyan, Natogyi	32.19	
Kume	Mandalay, Innwa, Tada-U, Kyaukse, Hanmyintmoe, Latnyoehtoe, Kume	92.13	Meiktila
	Meiktila, Thapyaywa, Wundwin, Kume	74.02	
	Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi, Myittha, Latnyoehtoe, Kume	164.55	
	Myingyan, Natogyi, Myittha, Latnyoehtoe, Kume	93.95	
Hanmyintmoe	Mandalay, Myitnge, Palate, Sintgaing, Kyaukse, Hanmyintmoe	56.53	Mandalay
	Meiktila, Thapyaywa, Wundwin, Kume, Latnyoehtoe, Hanmyintmoe	90.11	
	Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi, Myittha, Latnyoehtoe, Hanmyintmoe	174.22	
	Myingyan, Natogyi, Myittha, Latnyoehtoe, Hanmyintmoe	102.99	

4.6 Evaluation Results with Five Branches

In this section, the system is tested with five branches such as Mandalay, Meiktila, Kyaukbadaung, Myingyan, Kyaukse and ordered the goods from five cities. For example, a customer ordered the goods form Myittha city. The system calculates the shortest distances from five branches to Myittha. This is Mandalay, Innwa, Tada-U, Kyaukse, Hanmyintmoe, Latnyoehtoe, Myittha with 96.96 kilometers, Meiktila, Thapyaywa, Wundwin, Kume, Latnyoehtoe, Myiththa with 85.29 kilometers, Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi, Myittha with 153.28 Kilometers, Myingyan, Natogyi, Myittha with 82.08 kilometers, Kyaukse, Hanmyintmoe, Latnyohtoe, Myittha with 35.39 kilometers. The system chooses Kyaukse as nearest branch because Kyaukse is the minimum distance to go to Myittha among five branches. This results can be seen Table 4.3.

Table 4.3 Sample Testing with Five Branches

Branches:Mandalay,Meiktila,Kyaukbadaung,Myingyan,Kyaukse			
Customer's City	Path	Distance(km)	Nearest Branch
Myittha	Mandalay, Innwa, Tada-U, Kyaukse, Hanmyintmoe, Latnyoehtoe, Myittha	96.96	Kyaukse
	Meiktila, Thapyaywa, Wundwin, Kume, Latnyoehtoe, Myittha	85.29	
	Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi, Myittha	153.28	
	Myingyan, Natogyi, Myittha	82.08	
	Kyaukse, Hanmyintmoe, Latnyoehtoe, Myittha	35.39	
Latnyoehtoe	Mandalay, Innwa, Tada-U, Kyaukse, Hanmyintmoe, Latnyoehtoe	88.91	Kyaukse
	Meiktila, Thapyaywa, Wundwin, Kume, Latnyoehtoe	77.24	
	Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi, Myittha, Latnyoehtoe	161.33	
	Myingyan, Natogyi, Myittha, Latnyoehtoe	90.13	
	Kyaukse, Hanmyintmoe, Latnyoehtoe	27.35	
Natogyi	Mandalay, Innwa, Tada-U, Kyaukse, Hanmyintmoe, Latnyoehtoe, Myittha, Natogyi	146.85	Myingyan
	Meiktila, Mahlaing, Taungtha, Natogyi	96.96	

	Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi	103.39	
	Myingyan, Natogyi	32.19	
	Kyaukse, Hanmyintmoe, Latnyoehtoe, Myittha, Natogyi	85.29	
Kume	Mandalay, Innwa, Tada-U, Kyaukse, Hanmyintmoe, Latnyoehtoe, Kume	92.13	Kyaukse
	Meiktila, Thapyaywa, Wundwin, Kume	74.02	
	Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi, Myittha, Latnyoehtoe, Kume	164.55	
	Myingyan, Natogyi, Myittha, Latnyoehtoe, Kume	93.95	
	Kyaukse, Hanmyintmoe, Latnyoehtoe, Kume	30.57	
Hanmyintmoe	Mandalay, Myitnge, Palate, Sintgaing, Kyaukse, Hanmyintmoe	56.53	Kyaukse
	Meiktila, Thapyaywa, Wundwin, Kume, Latnyoehtoe, Hanmyintmoe	90.11	
	Kyaukbadaung, Poppa, Welaung, Taungtha, Natogyi, Myittha, Latnyoehtoe, Hanmyintmoe	174.22	
	Myingyan, Natogyi, Myittha, Latnyoehtoe, Hanmyintmoe	102.99	
	Kyaukse, Hanmyintmoe	14.48	

CHAPTER 5
CONCLUSION

5.1 Conclusion

Online shopping is an Internet application that has spread rapidly in the developed countries. Online shopping identified as the process of buying goods and services directly over the internet which considered as a form of e-commerce transactions. Since the internet is the only medium for online shopping, it became one of the most attractive facilities for internet users where consumers place an online orders and the retailer is responsible for fulfilling their orders. Home delivery service is an essential service for online purchased goods. Therefore, reliable delivery system is becoming necessary. The proposed delivery system is necessary for both online shopping users to identify their delivery addresses and find the nearest branch with the location of the delivery address and plan delivery routes. Therefore, this system identifies the process of order fulfilling as the process of planning, organizing and dispatching consumers' orders and prepares them to be delivered to the consumer's doorstep or any other delivery location in Mandalay Division.

5.2 Future Works

The system is only implemented for Mandalay Division. Therefore, the future works of the system is to extend this home delivery system that contains all States and Division of Myanmar.

REFERENCES

[1] Al Bakri, A. (2013) "An Overview of Information and Communication Technology (ICT) in Jordan: Review the Literature of Usage, Benefits and Barriers", vol. 1, no. 2, pp.9-15.

[2] Almeida, G., Avila, A., & Boncanoska, V. (2006) "Promoting e-commerce in developing countries, Internet Governance and Policy-Discussion papers", no. 2006.

[3] Baumgartner, M., Léonardi, J. and Krusch, O. (2008) "Improving computerized routing and scheduling and vehicle telematics: A qualitative survey", Transportation Research Part D: Transport and Environment, vol. 13, no. 6, pp. 377-382.

[4] Borndörfer, R., Grötschel, M., Pfetsch, M.E., "A path-based model for line planning in public transport", Technical Report 05-18, Zuse Institute, Berlin (2005).

[5] Browne, M., Allen, J., Anderson, S. and Jackson, M. (2001) "Overview of Home Deliveries in the UK (A study for DTI)", Freight Transport Association.
UniversityofWestminster.Disponívelem:
http://wmin.ac.uk/transport/Acessoem, vol. 17, pp. 1-6.

[6] Cairns, S. (2005) "Delivering supermarket shopping: more or less traffic?", Transport Reviews, vol. 25, no. 1, pp. 51-84.

[7] Chaffey, D. (2009) E-Business and E-Commerce Management, Fourth edn,.

[8] Chaudhry, A. (2006) "Jordan retail food sector", GAIN Report, USDA Foreign Agriculture Service, .

[9] De Koster, R. (2003) "Distribution strategies for online retailers", Engineering Management, IEEE Transactions on, vol. 50, no. 4, pp. 448-457.

[10] Dzuba, G., Filatov, A. and Volgunin, A. (1997) "Handwritten zip code recognition", Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference onIEEE, , pp. 766.

[11] Effect of Online Purchased Goods Delivery Service on Environment

[12] Ghazali, E., Mutum, D. and Mahbob, N.A. (2006) "Exploratory study of buying fish online: are Malaysians ready to adopt online grocery shopping?", International Journal of

[13] Hart P. E., Nilsson N. J., Raphael B., A Formal Basis for the Heuristic Determination of Minimum Cost Paths, IEEE Transactions on Systems Science and Cybernetics SSC4 (2): pp. 100–107, 1968.

[14] Jacob R., Marathe M. and Nagel K., "A Computational Study of Routing Algorithms for Realistic Transportation Networks", ACM Journal of Experimental Algorithms Vol 4 No 6 (1999).

[15] Jordan Post (2013) Jordan Post Company (JPC). Available at:http://www.jordanpost.com.jo/index.php?option= com_content &view=article&id=50&Itemid=55&lang=en (Accessed: 08/13 2013).

[16] L. Fua, , , D. Sunb, , L.R. Rilett, "Heuristic shortest path algorithms for transportation applications: State of the art", Volume 33, Issue 11, November 2006, Pages 3324–3343

[17] Ministry of Information and Communication Technologies in Jordan (2011) The E-Readiness Assessment of the Hashemite Kingdom of Jordan. Available at: www.moict.gov.jo (Accessed: 11/25 2013).

[18] Nuseir, M.T., Arora, N., Al-Masri, M.M.A. and Gharaibeh, M. (2010) "Evidence of Online Shopping: A Consumer Perspective", International Review of Business Research Papers, vol. 6, no. 5, pp. 90-106.

[19] Pape, U., Reinecke, Y.-S. , and Reinecke,E. , "Line network planning", in Daduna et al. [11], pp. 1-7.

[20] Rabin, S., "A* Aesthetic Optimizations", In Mark Deloura, editor, Game Programming Gems,pages 264 271. Charles River Media, 2000.

[21] Rabin, S., "A* Speed Optimizations", In Mark Deloura, editor, Game Programming Gems, pages 272 287. Charles River Media, 2000.

[22] Siikavirta, H., Punakivi, M., Kärkkäinen, M. and Linnanen, L. (2002) "Effects of E-Commerce on Greenhouse Gas Emissions: A Case Study of Grocery Home Delivery in Finland", Journal of Industrial Ecology, vol. 6, no. 2, pp. 83-97.

[23] Silman, L. A., Barzily, Z., and Passy, U., "Planning the route system for urban buses", Comput. Oper.Res. 1 (1974), pp. 201-211.

[24] Sonntag, H. , Ein heuristisches Verfahren zum Entwurf nachfrageorientierter Linienfuhrung imoffentlichen Personennahverkehr, Z. Oper. Res. 23 (1979), pp. B15-B31.

[25] Xia, X., Huang, Y. and Zhu, H. (2010) "Consumer logistics tradeoffs in EGS environment", Logistics Systems and Intelligent Management, 2010 International Conference onIEEE, vol. 3, pp. 1549.

APPENDIXES

Shopping Cart Quantity: 2 Finish Shopping:

Delivery System of Online Shopping

	Description	Price				
	Samsung RH29H9000SRAA	$2695.10		Add	Update	Remove
	Samsung RF18HFENBWW	$1149.00		Add	Update	Remove

Figure A1. Customer Home Page

User Login Here!

Email:

Password:

Login

New Member? Registration

Figure A2. Login Page

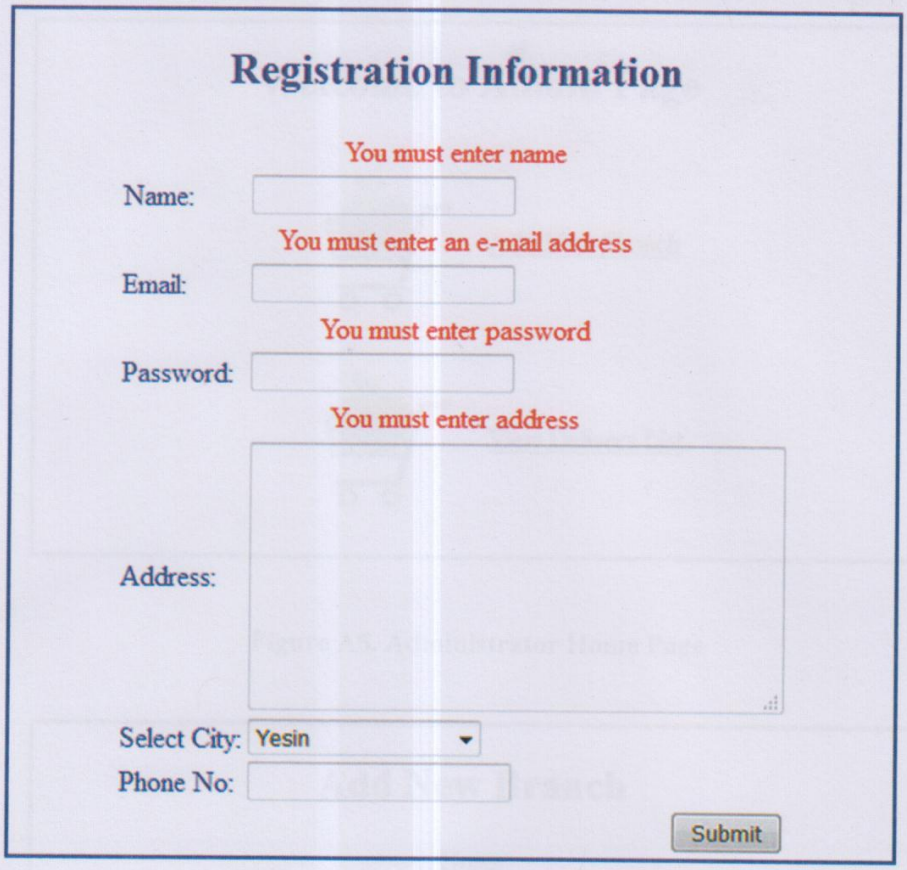


Figure A3. Registration Page

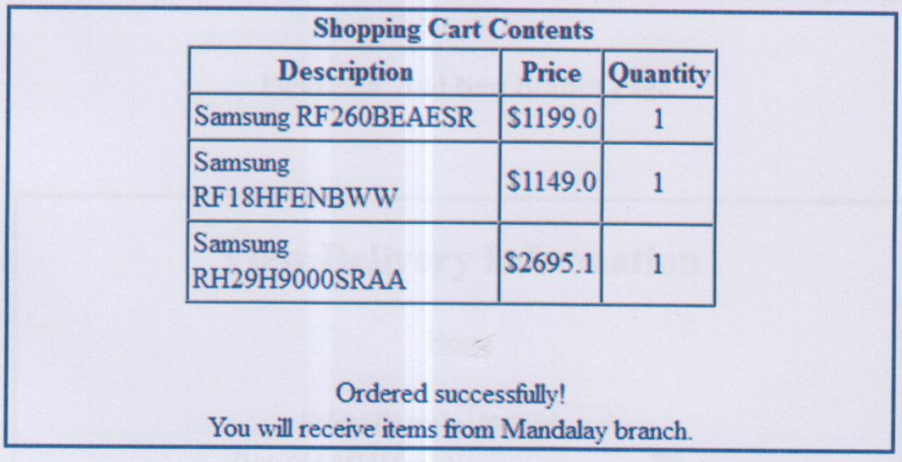


Figure A4. Order Successful Page

Figure A5. Administrator Home Page

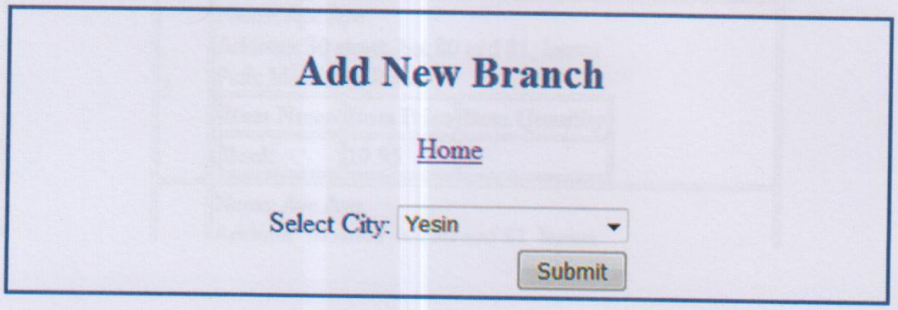
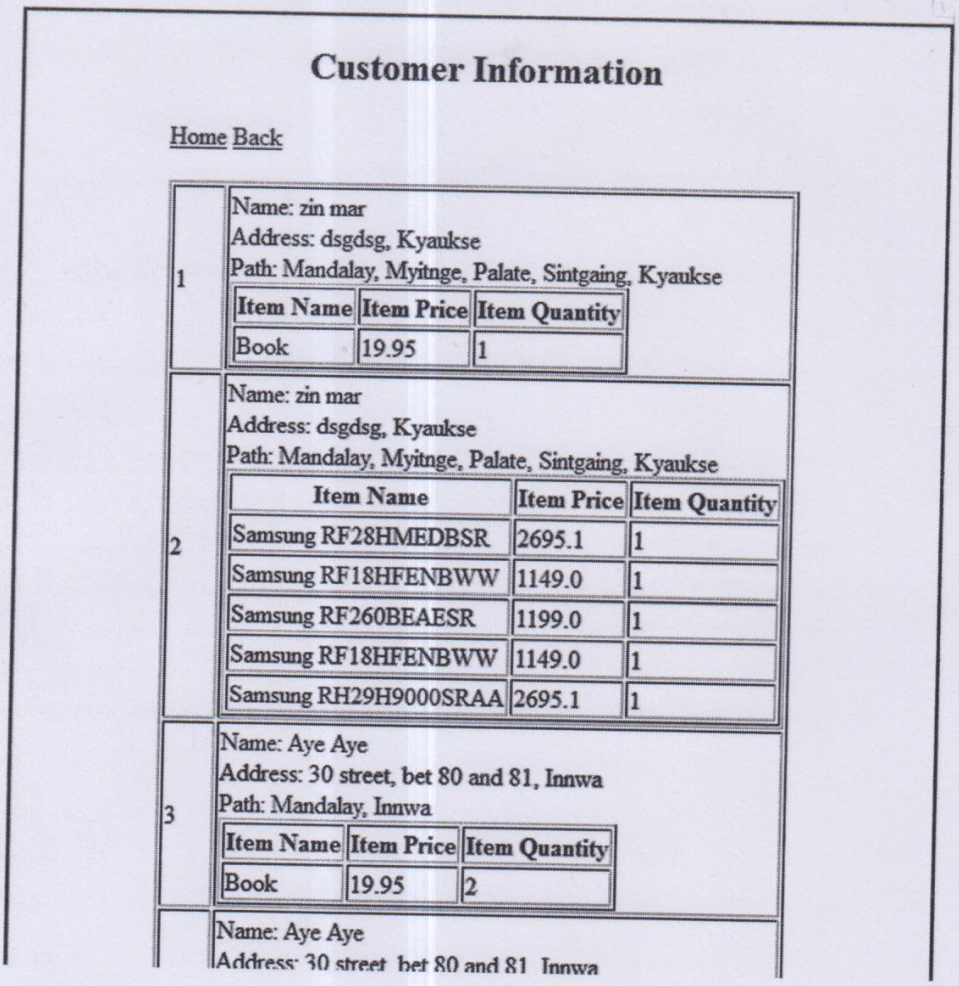


Figure A6. Add New Branch Page

Figure A7. View Delivery Information Page



Customer Information

Home Back

1	Name: zin mar Address: dsgdsg, Kyaukse Path: Mandalay, Myitnge, Palate, Sintgaing, Kyaukse

Item Name	Item Price	Item Quantity
Book	19.95	1

2	Name: zin mar Address: dsgdsg, Kyaukse Path: Mandalay, Myitnge, Palate, Sintgaing, Kyaukse

Item Name	Item Price	Item Quantity
Samsung RF28HMEDBSR	2695.1	1
Samsung RF18HFENBWW	1149.0	1
Samsung RF260BEAESR	1199.0	1
Samsung RF18HFENBWW	1149.0	1
Samsung RH29H9000SRAA	2695.1	1

3	Name: Aye Aye Address: 30 street, bet 80 and 81, Innwa Path: Mandalay, Innwa

Item Name	Item Price	Item Quantity
Book	19.95	2

Name: Aye Aye
Address: 30 street, bet 80 and 81, Innwa

Figure A8. Customer Information Page

Customer Information

Home Back

1	Name: zin mar Address: dsgdsg, Kyaukse Path: Mandalay, Myitnge, Palate, Sintgaing, Kyaukse

Item Name	Item Price	Item Quantity
Book	19.95	1

2	Name: zin mar Address: dsgdsg, Kyaukse Path: Mandalay, Myitnge, Palate, Sintgaing, Kyaukse

Item Name	Item Price	Item Quantity
Samsung RF28HMEDBSR	2695.1	1
Samsung RF18HFENBWW	1149.0	1
Samsung RF260BEAESR	1199.0	1
Samsung RF18HFENBWW	1149.0	1
Samsung RH29H9000SRAA	2695.1	1

3	Name: Aye Aye Address: 30 street, bet 80 and 81, Innwa Path: Mandalay, Innwa

Item Name	Item Price	Item Quantity
Book	19.95	2

Name: Aye Aye
Address: 30 street, bet 80 and 81, Innwa

Figure A9. View Common Paths Page

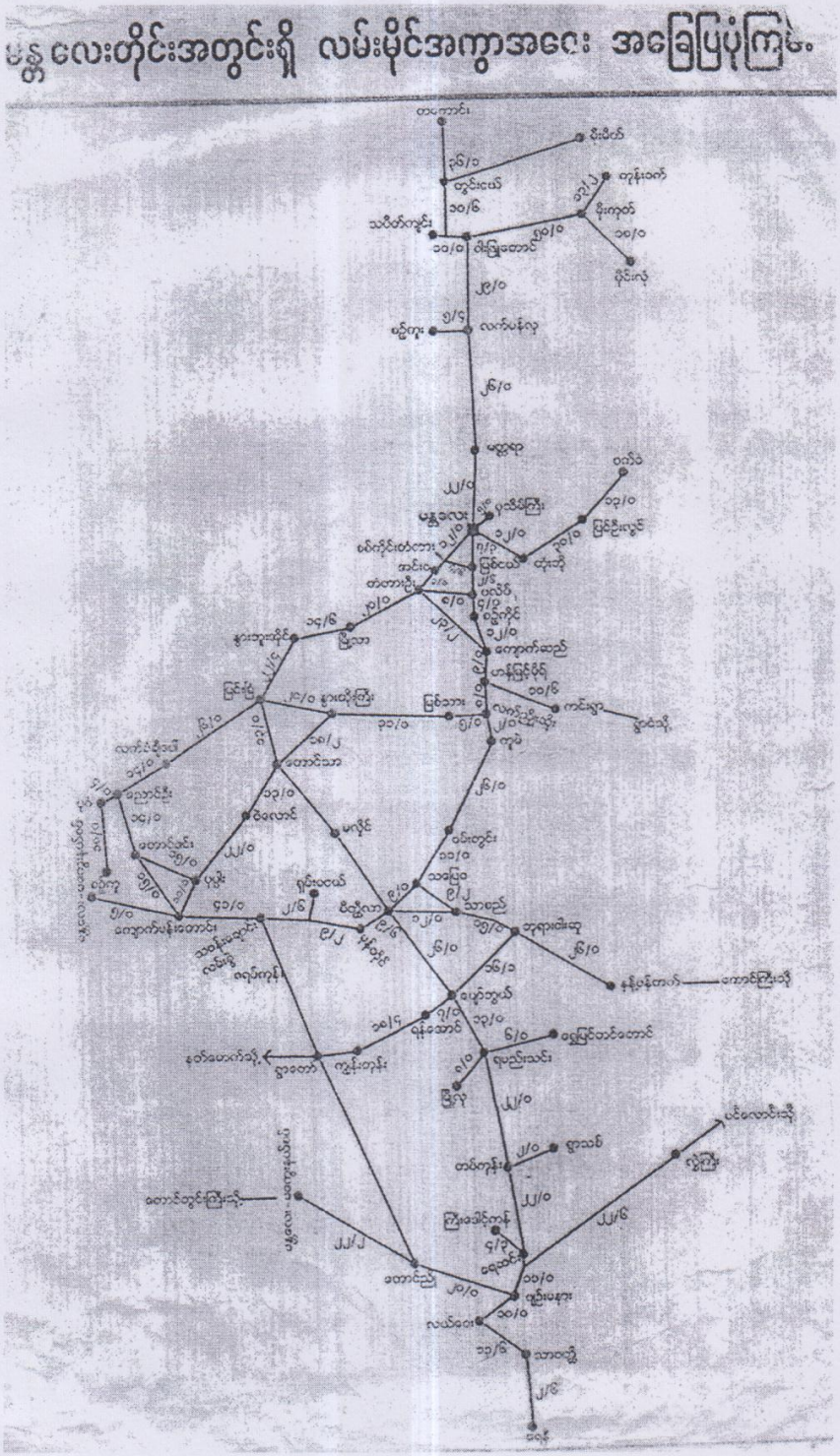


Figure A10.Mandalay Division Map