Smart Indoor Parking System Based on Dijkstra's Algorithm

K.I. Mohammad Ata*, A. Che Soh*, A. J. Ishak, H. Jaafar, N. A. Khairuddin

Department of Electrical & Electronic Engineering,
Faculty of Engineering,
University Putra Malaysia
43400, Serdang Selangor
*corresponding authors: g.piwanttobe@yahoo.com, azuracs@upm.edu.my

Abstract – *Smart Indoor Parking System is a parking system that assigns the car to the nearest parking to the entrance by using Dijkstra's Algorithm and assigns according to the size of the car. There are many types of parking system have been proposed such as smart parking system by using Wireless Sensor Network (WSN) but all these methods have their own advantage and limitation. Besides that, there are also several problems with the current parking system such as lack of parking management system efficiency. Therefore, this Smart Indoor Parking System is proposed to increase the efficiency of current parking management system. The aim of this Smart Indoor Parking System is to provide the customer with the nearest parking to the entrance. The parking is assigned according to the size of the car to utilize the parking space. Then, the parking place is displayed on the monitor besides the boom gate before allowing the driver to enter the parking lot. The parking number will help the driver to be able to find a parking lot in a short period of time without the need to search the parking by themselves. The priority is given to the driver who presses the button first. The input signal from the IR sensor located at the entrance and the parking lot is processing to assign the nearest parking lot based on Dijkstra's Algorithm. In this project, the simulation is carried out by using Proteus. This simulation includes all the circuits for determining the direction of the nearest parking lot. Programming in the Arduino is implemented to calculate the nearest parking lot based on Dijkstra's Algorithm. In conclusion, the Smart Indoor Parking System can be used in the real-time application.*

Keywords: *Dijkstra's algorithm, a smart parking system, Arduino, midsize class car, compact class car*

Article History
Received 11 July 2018
Received in revised form 23 August 2018
Accepted 24 August 2018

I. Introduction

The conventional and current parking system in Malaysia is developed in a standard size of parking lots have created the wastage in using limited parking spaces. Because of that, the parking lots will be easily full and most of the driver that enter the parking will be parked car easily especially at favorite spot closed to the mall entrance without thinking the traffic congestion and others comfortability. Other than that, the conventional parking system requires the driver to queue at the boom gate while waiting for the ticket which will be available when there is an empty parking space. The driver also needs to search the parking lot by themselves. This will bring difficulties in searching the available parking lot leads to time and fuel wasting especially while in idling or driving around parking lots that lead to more carbon dioxide (CO2) emission being produced. In addition, potential accidents will be occurred caused by an abundance of moving vehicles in brake and go while searching for an empty parking lot at the disorganized parking system. These entire factors will cause high frustration and stress level of the driver. Thus, every inch of parking lot area needs to be utilized optimally base on the car size and the traffic flow must be controlled systematically to prevent congestion in tight spaces.

To cater with the limitation of parking lot spaces, "Smart Indoor Parking System" is proposed to make sure these conditions can be fulfilled to be applied at indoor parking zone which able to scan, count, display, assign the nearest lot and display the location on the monitor at the boom gate. Upon arrival at the entrance gate of parking space, the car will be located between two bumpers where the infrared (IR) sensor located to scan the size of the car. Then, the driver will press the button to get the specified direction of a nearest vacant lot to the main entrance of the mall based on the size of the car displayed on the monitor at the boom gate to park their vehicles. However, once the parking lots are full, the system will not allow any driver entered the parking lot by preventing the boom gate opened until there is an available parking lot. Thus, the system able to save a lot of time in searching for empty lots and allowing the parking lot traffic to run smoothly. Before the boom gate open, the driver needs to reconfirm the location in form of parking lot number as a kinesthetic mode to enhance the driver alertness about the parking lot location due to the repeal of ticket system for encouraging the green environmentally friendly and queue a long time at the entrance gate.

The assigned vacant lot is locked for few minutes to allow the driver to park and will be available for next driver when the time allocated has ended or no car is parked as the IR sensors installed on top of every parking lot will always detect the occupancy. The signal from the sensor will be sent to the controller for distance calculation and display the number of occupancies on the LCD display. Besides, on top of every parking lot, there are two LEDs as an indicator for the availability of parking lot. The LED indicator will be turned red when the parking lot is occupied and will be turned green when the parking lot is empty. Modified Dijkstra's algorithm is applied to calculate the nearest distance base on first come first serve basis between each of the lot and the entrance of the mall. All parking lots that are arranged based on the distance priority sequence calculated by the algorithm.

To keep updating the parking status, the main controller will acquire data input from each of the sensors for triggering the system rescanned at a certain time so that the LCD counter display is always synchronized in real time for the parking lot. In addition, this system integrated with an online monitoring system displays the availability of parking lots based on the IR sensor. Those IR sensors have connected each other via wireless. This feature has helped the parking system operator to control and observe the status for each parking lot through LCD counter display. This indoor parking guidance system will bring a lot of benefits to the driver and the parking lots management.

The paper is organized as follows. In the next section, the literature review for the smart parking system is described. Section 3 briefly discusses the details of the methodology for the development of smart indoor parking systems such as system design, the layout of parking, Dijkstra's algorithm and software programming. Section 4 discusses the simulation results of the system. The conclusion of this paper is summarized in the last section.

II. Literature Review

Considering the growing requirement of a smart parking system, different parking systems have been presented in the past. Several researchers have designed and enhanced smart parking systems to make them efficient and reliable so that people can easily locate an ideal parking lot according to their requirements. Various approaches and techniques have been deployed in the past years with different outcomes.

Propositioned an intelligent parking system that lets users reserve parking for cars and directs them to the free parking for cars simultaneously. The given solution not only assists in a significant reduction in traffic arising due to parking space search but also improves the occupancy by easily managing the space for parking, an expensive resource in large cities [1]. This solution substantially reduces the adverse impact on the environment by decreasing the carbon footprint.

Alkharabsheh [2] has introduced an intelligent cooperative parking system based on multi-agents. The given solution chiefly consisted of a multi-agent approach combined with hardware elements like IR sensor nodes to detect the parking slot status and accordingly transmit the information to Arduino. A device called gateway can be used to gather the status from the sensors wirelessly. The system of agents will be used to assess the information collected, and then send the drivers the relevant results in real time through the Internet. This solution can be used to solve many issues like the cost of petrol, car accidents, and traffic management. The outcomes demonstrate the practicality of the proposed solution, which also achieved excellent results.

Mendiratta *et al*. [3] put forward a mechanism for the detection of car parking space with the help of an ultrasonic sensor, combined with the concept of the Internet of Things. It was achieved by transmitting the parking slot status to the Internet. This technique sends the data from the ultrasonic sensor via Wi-Fi in which the IoT platform is also involved. The problem in that system was that ultra-sonic sensor was getting rust on its surface. That led the system to misbehave. Caballero-Gil *et al*. [4] also used IoT for smart parking guidance system. Their solution includes a central system to estimate available indoor parking areas, and an inexpensive smartphone application to get data of predicted and actual parking occupancy. This scheme utilizes data from each of the two sources directionally in order that the central forecast system is provided with data acquired from the distributed system which is based on smartphones, and the other way around. The mobile application employs various wireless technologies to supply the estimation system with real-time parking data and obtain useful suggestions from the system regarding the parking space.

Their outcomes confirm the proposed system's strengths, i.e. the decrease in the energy and time requirements to find the available parking area.

Bo *et al.*[5] recommended an algorithm for smart parking regulation by considering three representative decision parameters (i.e. walk duration, parking charges, and the number of empty parking spaces) where the estimated number of empty parking spaces is considered an important parameter to reflect the degree of difficulty in finding existing parking spaces, as well as a queuing theory-based technique was suggested to calculate approximately the expected number of vacant parking spaces with diverse capacities, rates of arrival, and rates of service. Another theory known as MADM (multiple attributes decision making) was also proposed to obtain a suitable space for parking. Experimental outcomes show that the recommended MADM-based technique is efficient in getting suitable spaces for parking to fulfill the users' preferences.

Based on the literature review, there are many smart parking systems have been developed to give good facilities to the user. However, a smart indoor parking system, especially in a shopping complex, is still not implemented based on the guidance system for the user to find the parking lot closer to the entrance. The layout of the parking still based on the standard bay. So, the proposed smart indoor parking system will give a good guide to the user to finding the closest parking lot to nearest entrance and gives more space for parking management with new parking bay will introduce according to two different sizes of vehicles.

III. Methodology

The Smart Indoor Parking System is designed to arrange the parking system efficiently by utilizing all the provided parking and guide the driver to find vacant parking without wasting their time and car's fuel consumption. Besides that, this parking system has differentiated the parking space according to the size of the car. To develop this parking system, software programming and circuit design implementation need to be done. The simulation circuit is designed by using Proteus software with Arduino Mega 2560. Arduino Mega 2560 is the major component used to undergo this simulation.

The proposed coverage of study and implementation is illustrated as in Fig. 1 and 2 respectively. This project overview is the step in designing the Smart Indoor Parking System and makes the project successful.

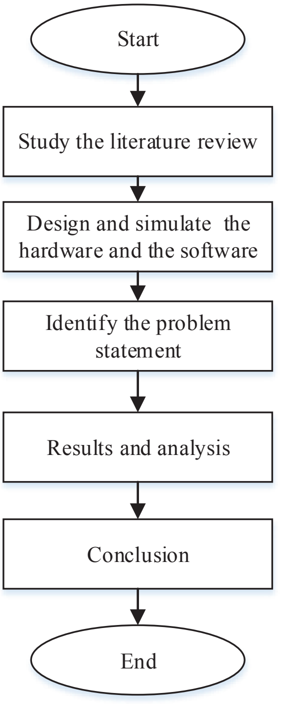


Fig 1. The proposed coverage of study

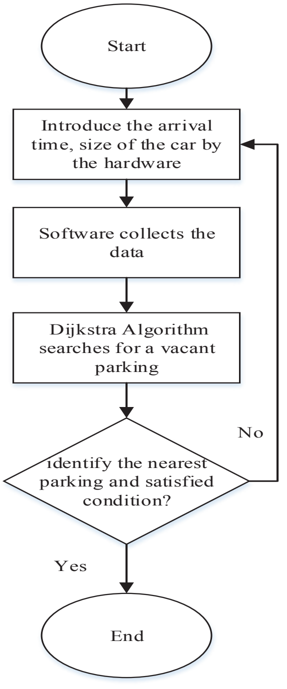


Fig 2. Methodology flowchart

A. System Design

The Smart Indoor Parking System can be illustrated as in Fig. 3. The car is assigned according to the vacant parking that nearest to the entrance door. The system operation started when the car is placed between two bumpers in front of the boom gate. Then, the Infrared (IR) sensor scans and specifies the size of the car. The size of the car and the shortest distance from the entrance to the parking lot will determine the placement of the parking.

Based on this system, 32 cars are provided for midsize class car and above while the other 32 parking is for the compact class car and below. To determine the size of the car, 3 LEDs are placed at both entrance gate to indicate the size of the car. Those sensors are placed based on a threshold for both sizes. Two values are determined and assigned as a threshold to identify the size of the car. When 3 LEDs are light on, it shows the car is midsize class and the car will be placed on the parking specified for the midsize class car. If less than 3 LEDs are light on, it shows that the car is compact class and the car will be placed on the parking specified for the compact class car. The size and dimension of the parking lot are illustrated in Fig. 4. This parking lot size is based on the worldwide parking bay standard. However, in this parking system design, the parking lot size is resized to merge with the project scale.

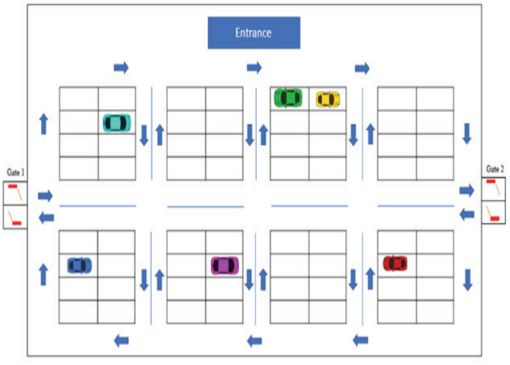
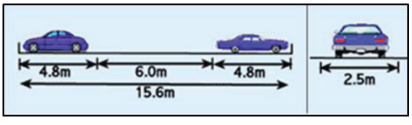
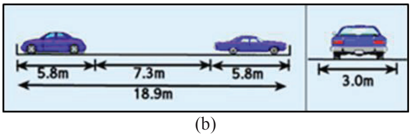


Fig 3. Floor plan of the parking system



(a)



(b)

Fig 4. Size and dimension (a) standard bay and (b) large bay [6]

This parking system consists of four main circuits which are a counter circuit, a gate circuit, a scanning circuit and LCD display circuit. IR sensor is used as both main circuit input, but for the entrance gate, it used a push button as the input. Then, the Analogue Digital Converter (ADC) is used to convert the analog signal from IR sensor to digital signal. This digital signal is the input to the minor circuit in this system which is Light-Emitting Diode (LED), scanning circuit, and counter circuit. All these circuits are controlled by Arduino Mega 2560. It controls the circuit based on the programming input that inserts in the Arduino. The programming is executed in the Arduino. For display circuit, the output is display on the Liquid Crystal Display (LCD) while the gate circuit output is the movement of the servo motor that controlled by sending them to pulse of variable width.

The push buttons at both entrance gate are the other input for the gate circuit. When the push button is press, the counter circuit in the system will calculate whether there is available vacant parking lot in the system and the shortest distance of the parking lot from the boom gate. If the vacant parking lot available, the LCD will display the information of parking lot available and the parking placement to the driver and the entrance gate will open to allow the car to enter the parking. The timer is counted based on distance calculation for parking lot reservation to give the driver time to find the reserve's parking. After the car is parked, the IR sensor on the parking will scan the present of the car and sends the occupied signal to the counter circuit. The counter will then be recalculated to update the availability status. If there is no car parked on the reserved parking, the IR sensor on the parking will scan the absent of the car and reset the system. The parking is now vacant and available to another car.

B. Dijkstra's Algorithm

Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks. It was conceived by computer scientist Edsger W. Dijkstra in 1956 and published three years later [7]. The algorithm exists in many variants; Dijkstra's original variant found the shortest path between two nodes, but a more common variant fixes a single node as the source node and finds shortest paths from the source to all other nodes in the graph, producing a shortest-path tree. The Dijkstra algorithm has been implemented in the parking system. The way that has been used to determine the distance via a pathway. The algorithm used the minimum value between two points.

The algorithm finds the shortest path between that node and every other [8]. It can also be used for finding the shortest paths from a single node to a single destination node by stopping the algorithm once the shortest path to the destination node has been determined.

For example, if the nodes of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities.

The algorithm can be explained and understood properly using an example. The example briefly explains each step that is taken and how distance is measured [9]-[10]. Consider the examples are shown in Fig. 5.

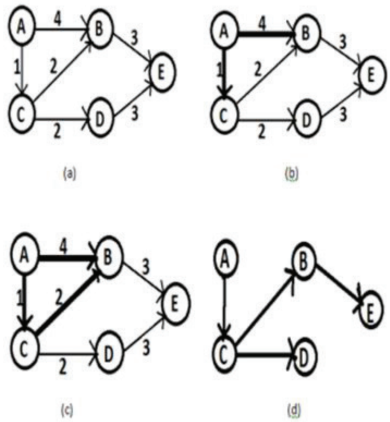


Fig.5. (a) Weighted-Directed Graph, (b) Shortest Path to vertices B, C from A, (c) Shortest Path from B, D using C as intermediate vertex and (d) The path obtained using Dijkstra's Algorithm [9]-[10]

The above-weighted graph has 5 vertices marked from A-E. Edge cost is the term used for the value between two vertices [9]-[10]. For example, the edge cost between A and B is 4. To find out the shortest path from the source A to the remaining vertices, Dijkstra's algorithm is applied to the above-shown graph. The example is solved as follows:

Step 0: sDist[A]=0; the value to the source itself

sDist[B]= inf, sDist[C]= inf, sDist[D]= inf, sDist[E]= inf ; as the nodes have not processed till now.

Step 1: Adj[A]={B,C}; computing the value of the adjacent vertices of the graph.

sDist[B]=4;sDist[C]=1;

Step 2: Computation from vertex C Adj[C]={B,D};

sDist[B]>sDist[C]+EdgeCost[C,B] 4>1+2(True)

Therefore,sDist[B]=3;sDist[D]=2;

Step 2 is clearly explained in Fig. 5(b).

Step 3: Computation from vertex B

Adj[B]={E};

sDist[E]=sDist[B]+EdgeCost[B,E]

=3+3=6;

Adj[D]={E};

sDist[E]=sDist[D]+EdgeCost[D,E]

=3+3=6

This is same as the initial value that was computed so sDist[E] value is not changed. See Fig. 5(c).

Step 4: Adj[E]=0; shows that there are no outgoing edges from E And no more vertices branching from it, algorithm terminated. Hence the path which follows the algorithm is shown in Fig. 5(d).

C. Software Programming

The programming for the parking system is simulated in Arduino. The programming is then downloaded in the Arduino Mega 2560 to run the circuit according to the instruction in the program. The Smart Indoor Parking System used Dijkstra's Algorithm to calculate the distance between the gate to the parking and from the parking to the entrance. The algorithm will determine based on both conditions and therefore it becomes the highest priority in the parking places. Some modification is applied in the Dijkstra's Algorithm which is the calculation of the distance is based on distance from the parking lot to the entrance and driving distance from the boom gate to the parking lot. Gate 1 and Gate 2 are not specified by any size of the car. Hence, the parking lot is assigned the nearest entrance door.

As shown in Fig. 6, the parking lot is divided into Zone A and Zone B to make the driver easier to find the parking lot. Each zone has specified a place for midsize class car and compact size class car. The parking lot with label A1 until A16 and B1 until B16 are specified for the midsize class car and parking lot with label AA1 until AA16 and BB1 until BB16 is specified for compact size class car. This figure also shows small circles indicating nodes which used to calculate the walking distance between the parking lot and entrance. There are three types of nodes which are initial nodes, several nodes, and destination nodes. The initial nodes, in the blue circle, are nodes at the parking lot while several nodes (yellow circle) are nodes that people walk to reach the entrance. Lastly, destination nodes (green circle) are nodes at the entrance. To find the shortest distance from the parking lot to the entrance, the distance from each of the initial nodes to the entrance are calculated. The system will continue looping to get the shortest distance from the vacant parking lot. This step is important to ensure the highest priority is given to the car that enters to the parking system.

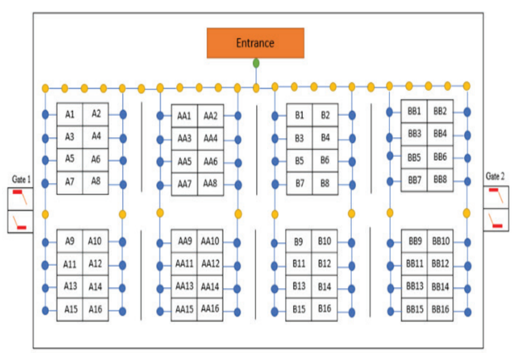


Fig 6. Distance calculation according to nodes

IV. Results and Discussions

8 parking lots are chosen as vacant parking in the system. Another parking lots are set as a dummy. These 8 parking lots are calculated based on their distance from the parking lot to the entrance. Parking lots A4, A7, B2, and B5 are for the midsize class car while parking lots AA6, AA7, AA14, and BB6 are for compact size class car. These vacant parking lot placements are shown in Fig. 7. The number in the yellow box is the vacant parking and the other parking are assumed occupied or dummy.

Fig 7. Vacant parking lot

The system then will calculate the distance of this vacant parking to the entrance and the shortest distance will the priority compared to the other parking. The distance and the priorities are shown in Table 1 for the midsize class car and Table 2 for the compact class size car. It shows that if the driver of midsize class car pushes the button, the system will assign to the parking lot B5 because the parking lot is nearer to the entrance compare to the other vacant parking and B5 is the priority in the system. However, if the driver of the compact class size car pushes the button, the system will assign the parking AA6 because of it is near to the entrance and it is a priority among the vacant parking.

TABLE I
DISTANCE AND PRIORITY FOR MIDSIZE CLASS CAR

Parking Lot	Distance from parking lot to the entrance	Priority
A4	11m	3
A7	17m	4
B2	8m	2
B5	6m	1

TABLE II
DISTANCE AND PRIORITY FOR COMPACT CLASS CAR

Parking Lot	Distance from parking lot to the entrance	Priority
AA6	6m	1
AA7	11m	2
AA14	12m	3
BB6	16m	4

In circuit design, each of the 8 parking is differentiated as shown in Fig. 8.

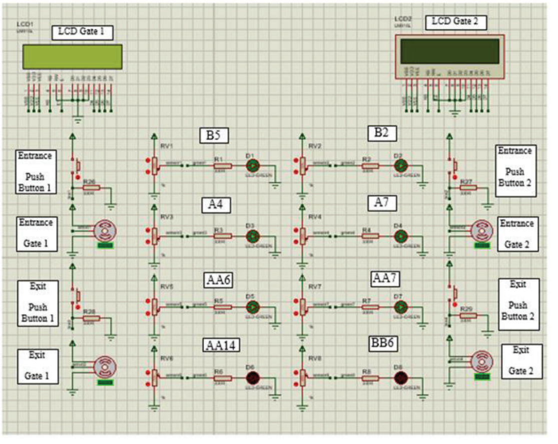
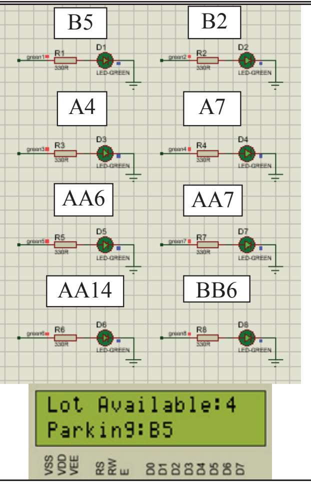
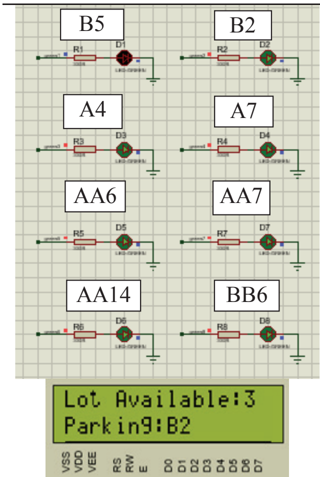


Fig 8. Circuit design

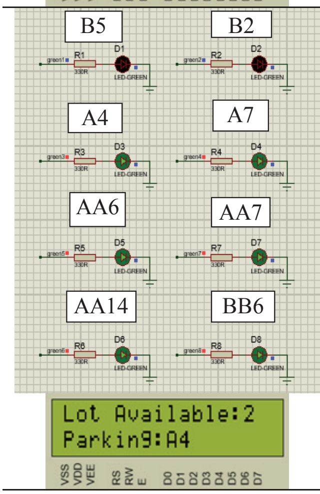
Next, as shown in Table 3, the LCD display has printed up the assigned nearest available parking lot based on Dijkstra's Algorithm and after determining the size of the car. The nearest parking lot also based on the IR signal to determine the occupancy status. The LCD display can be either for Gate 1 or Gate 2 according to first come first serve.

TABLE III
THE STATUS OF LCD FROM PARKING LOT SIGNAL

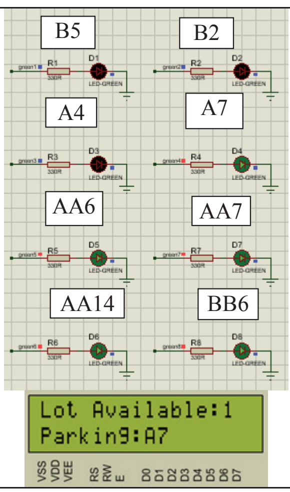
LED and LCD Display in Parking Lot	Description
	Parking Lot B5 is chosen for the midsize class car because it is the nearest parking to the entrance and it is a priority according to the Dijkstra Algorithm.



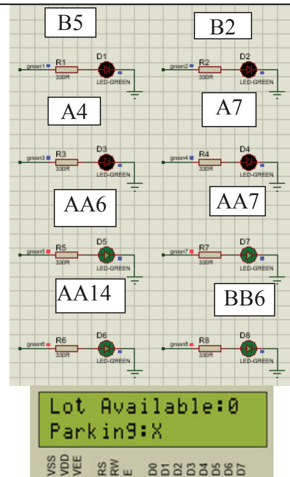
Parking Lot B2 is chosen for the midsize class car after B5 is occupied because it is the nearest parking to the entrance and it is the second priority according to the Dijkstra Algorithm.



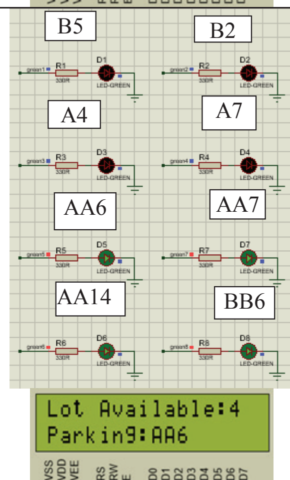
Parking Lot A4 is chosen for the midsize class car after B5 and B2 are occupied because it is the nearest parking to the entrance and it is the third priority according to the Dijkstra Algorithm.



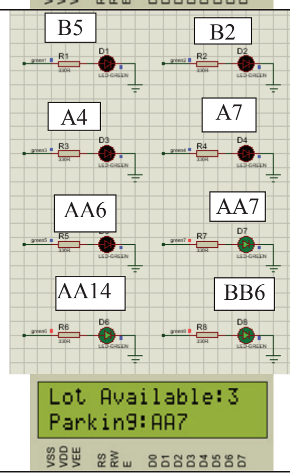
Parking Lot A7 is chosen for the midsize class car after B5, B2 and A7 are occupied because it is the nearest parking to the entrance and it is the fourth priority according to the Dijkstra Algorithm.



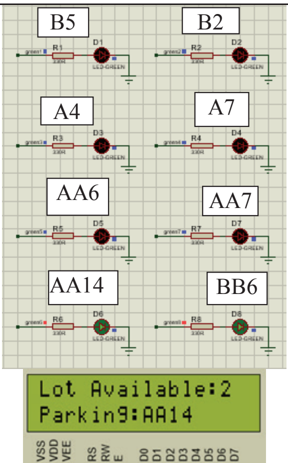
No parking lot is displayed because there is no parking available for the midsize class car.



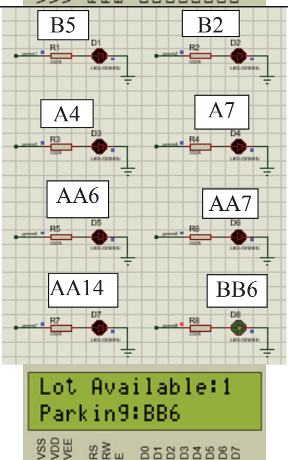
Parking Lot AA6 is chosen for compact size class car because it is the nearest parking to the entrance and it is the first priority according to the Dijkstra Algorithm.



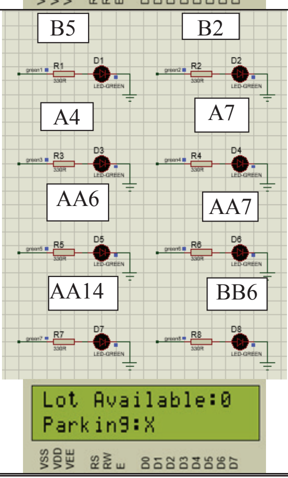
Parking Lot AA7 is chosen for compact size class car after AA6 is occupied because it is the nearest parking to the entrance and it is the second priority according to the Dijkstra Algorithm.



Parking Lot AA14 is chosen for compact size class car after AA6 and AA7 are occupied because it is the nearest parking to the entrance and it is the third priority according to Dijkstra Algorithm.



Parking Lot BB6 is chosen for compact size class car after AA6, AA7 and AA14 are occupied because it is the nearest parking to the entrance and it is the fourth priority according to the Dijkstra Algorithm.



No parking lot is displayed because there is no parking available for compact size class car.

V. Conclusions

The smart indoor parking system has been successfully designed and developed accordingly. Based on this system, it is proved that the system has provided convenience parking system to the user. since the priority is to find and provide the nearest parking lot to the entrance for the user. The driver doesn't need to waste their time searching the vacant parking and he also can obtain the nearest parking to the entrance with the help of the system. It is also provided parking according to the size of the car, that made all parking can be utilized efficiently. Besides that, the smart indoor parking system has provided the nearest parking lot to entrance according to the Dijkstra Algorithm. Dijkstra Algorithm is the calculation method for determining the shortest distance. By applying Dijkstra Algorithm, all the parking lot in the parking system can be utilized efficiently.

Acknowledgments

The authors wish to acknowledge the Department of Electrical and Electronic Engineering, University Putra Malaysia for supporting and funding this research.

References

[1] R. Sehdehi, T. Henderson, M. Nadeem, A novel inductively powered intelligent parking solution, *International Conference on Industrial Electronics for Sustainable Energy Systems (IESES), pp. 391-396, Waikato, New Zealand, 31 Jan-2 Feb 2018*

[2] A.R.A. Alkharabsheh, An intelligent cooperative multi-agents-based parking system: design and implementation. *Journal of Theoretical & Applied Information Technology, Vol. 96*(No.10) *: 2804-2815, May 2018*.

[3] S. Mendiratta, D. Dey, and D. R. Sona, Automatic car parking system with visual indicator along with IoT. *2017 International conference on Microelectronic Devices, Circuits, and Systems (ICMDCS), pp. 1-3, TamilNadu, India, August 2017*.

[4] C. Caballero-Gil, P. Caballero-Gil, and J. Molina-Gil, Cellular Automata-Based Application for Driver Assistance in Indoor Parking Areas. *Sensors, Vol. 16*(11)*, 1921: 1-17, November 2016*

[5] L. Bo, P. Yijian, W. Hao and H. Dijiang, MADM-based smart parking guidance algorithm. *PLoS ONE 12*(12): *1-30, December 2017*

[6] R. John, and Ltd. Sons, Car Parking Bays, [Online]. http://www.multi-storey-car parks.com/car_parking_bays.html [Accessed 23 Novermber 2016].

[7] E. W. Dijkstra, A note on two problems in connexion with graphs. *Numerische Mathematik*, pp. 269-271, 1959.

[8] K. Mehlhorn, and P. Sanders, Algorithms and Data Structures: The Basic Toolbox (Springer, 2008.)

[9] G. Abishek, Path Finding: A or Dijkstra's. *International Journal of Management,IT and Engineering (IJMIE), Vol. 4*(No. 7), 2014.

[10] J. Charika, and K. Jitendra, Processing delay consideration in dijkstra's algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3*(No. 8): *407-411, 2013*.