|| Volume 3 || Issue 4 || April 2018 || ISSN (Online) 2456-0774
INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH
AND ENGINEERING TRENDS

Peer to Peer On-Demand Valet Service System using Mobile Application

Mr. Bhushan Naik[1], Miss. Shivangi Bhat[2], Miss. Sneha Awhale[3], Mr. Mahesh Shelke[4], Prof. Aruna Verma[5]
UG Students Department of Computer Engineering, Dhole Patil College of Engineering, Wagholi, Near Eon IT park., Pune-412207, Maharashtra, India[1 2 3 4]
Assistant Professor, Department of Computer Engineering, Dhole Patil College of Engineering, Wagholi, Near Eon IT park., Pune-412207, Maharashtra,India[5]

Abstract— **Valet parking system consists of thoughtfully crafted yet easily mastered software applications which are easy to use for technologically less oriented people to use ride on demand service. Our findings have made us understand the need of on demand valet service system in the metropolitan cities of India due to the increasing population and the subsequent increase in the vehicle traffic. This paper introduces a novel algorithm that provides a valet parking system and develops a architecture based on the Firebase Cloud Messaging (FCM) technology. This paper proposed a system that helps users find parking solutions at the least cost based on new performance metrics to calculate the user parking cost by considering the distance and time. This cost will be used to offer a solution of finding an available valet service provide upon a request by the user and a solution of suggesting a new valet service provider if the current valet service provider does not accept the request or nobody is available to serve at the moment. The simulation results show that the algorithm helps in providing on demand valet service to the users and minimizes the user's hassle is finding parking in the real time. We also successfully implemented the proposed system in the real world.**

Keywords: *Mobile services, on-demand Services, performance metrics, FCM.*

I INTRODUCTION

A. Motivation

Influx of millions of vehicles in the present scenario has resulted in the over brimming of roads. Value of the real estate is shooting up. Unnecessary wastage of fuel is observed due to traffic congestion. As a consequence vehicular exhaust is contributing to the atmospheric pollution. Valet Parking system that has the ability to absorb the traffic, convenient for the vehicle owners, approved by the builders, compact, time saving is today a requirement.

B. Solution

This project will be leading the charge on a new industry of on demand services that address aubiquitous urban challenge: finding a parking spot.It is a mobile based application that helps to find secure parking spots in your city and provide valets on-demand who will assist in parking and attending to your vehicle. With the help of this project we will strive to make parking a vehicle both convenient and affordable. By removing cars from city centres, making better use of underutilized parking spots, and relieving the pain of parking for drivers, we can truly improve our customer's daily routines and relieve civic pain with an affordable, convenient, and fast service. Within our app, tell us where to meet you our Valet will be there when you arrive. When you're ready to get your car back, schedule the return, and your car will meet you at the time and place of your choosing. In this paper, we explore valet parking's role in On demand mobile services, particularly in comparison with ride sourcing like uber, ola, etc, through a systematic on demand service model which will provide the best possible experience to the user.

Some interesting features about Parking:

- Typically, car owners spend an average of 10-15 min looking for parking spot.
- While parking is free or cheap in most places (at least in India), there is a cost in terms of lost time and uncertainty of finding a suitable parking space.

As in most informal fragmented markets, 'Jugaad' (workaround) solutions exist at the local level – e.g many offices lease space from empty or unused properties OR buildings with offices and commercial spaces use the available parking at complementary timing etc.

II RELATED WORK

There are expanding number of outing arranging administrations that propose open transport courses to clients with timetables and other data. A considerable lot of them are coordinated with continuous information about travel times, which can be consolidated with data stages in light of group sourced data [1], including movement blockage, prepare flag disappointments, and so forth. As of late, auto sharing of cabs or private vehicles has turned out to be progressively well known. Portable administrations are created to allot a reasonable taxi or shared auto to each excursion in separation. Such versatile administrations can be considered as an augmentation of

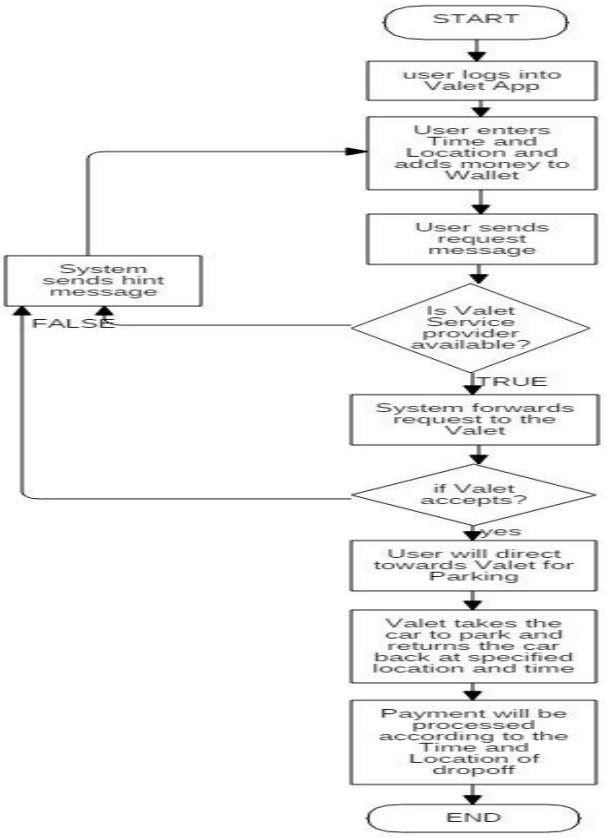
existing taxi calling administrations. They use methods for blurring a reasonable ride among an arrangement of applicant vehicles in view of specific criteria. Various worldwide arranging systems have been explored, for example, heuristic systems for the heap adjusting of accessible taxicabs to serve anticipated future request with bring down inertness.

Uber [4] is a versatile application like the taxi calling framework which empowers clients to ask for drivers keeping in mind the end goal to be transported. Customarily, one who needs to call a taxi normally needs to influence demand to a call to focus. With the Uber portable application, everything winds up noticeably less demanding and quicker. One can ask for an auto on the web, then monitor the auto's GPS area on a guide until the point that the auto comes to pick one up. The application gives a toll estimation for the ride, with the goal that the client knows the cost ahead of time. At last, the client can pay effortlessly through the application so the requirement for money is disposed of. Uber likewise gives clients the chance to give input after a ride. Kutsuplus [5] is another transport benefit transport framework conveyed at Helsinki in Finland. The idea has been produced by Ajelo and is very like Uber however as opposed to getting a taxicab on-request you get a transport on-request. A us demands a transport on a versatile application picking the beginning stage, the goal, the takeoff time and the quantity of seats you need to book. Kutsuplus will at that point search for other individuals requesting a transport to a similar goal and will propose conceivable courses. Once the excursion is confirmed, the client can pay straightforwardly through the versatile application with a virtual wallet. The cost is not fixed as in standard transport travel. For each excursion recommendation after a request, the toll is shown. The toll can be part and ends up noticeably less expensive in the event that one books a transport for a gathering. Ajelo was a start-up organization that built up the calculation and was procured in November 2014 by Split [8] to send a common ride benefit in the United States.

III PROPOSED SYSTEM

1) SYSTEM OPERATIONS

When a user wants to book a valet to find parking at the desired location and time, he must login to our system. After successful login, a request message is sent to search for the available valet. Then, the system will send back a response message containing the information, including the valet name, contact details and the directions to reach him. The choice of the valet is based on the function F(α,β), which is calculated based on the current location of the user and the location of the valet. The system will forward the user to a valet with a minimum F(α,β) value if there are more than one valet present at the location. When the user arrives at the location, he must identify the valet using his given details and drop the car to him and specify the location of drop off and time of the same. After the time is up the valet will go to drop the vehicle and notify on the app that the trip is completed and the money will be deducted from the wallet of the user. If there is no valet available to serve at that moment then, the system will send a suggestion message that includes information on a new valet details, including the address and new directions, with a minimum cost. The new valet will be selected based on the neighbour table of the current car park (the first node in the neighbour table), as shown in figure



Each node has a neighbour table to maintain information on the current status of the network and a queue with preened length. The neighbour table for each node contains information on the neighbouring nodes directly linked to it. On the other hand, the queue is used to control the number of vehicles forwarded to the node, which aims to prevent overloading in the number of vehicles beyond the capacity of the node. In our proposed system, each node will broadcast a message to its neighbouring nodes after a new node joins or leaves it. This message includes information on its total free resources. The neighbouring node that receives this message will update its neighbour tables. We have assumed that, in our network, where _ is a coefficient that depends on the length of the path between two nodes and _ is a coefficient that depends on the number of free slots in the destination node. F(α, β) is inversely proportional to the distance between two nodes and

directly proportional to the total free slots in the destination node. Depending on which parameter we consider to be the more important of the two parameters, i.e., the distance or the free slots, we can adjust _ and _ to achieve better network performance. _ and _ are parameters derived from the experiment, and their value is [0, 1]. If _ D 0, we only consider the number of free spaces to calculate the cost to the user. If _ D 0, we only consider the distance between two nodes to calculate the cost to the user. In Eq. (1), we calculate the cost function based on the distance between two nodes and the percentage of free parking spaces at each node. We use the upper bound of the distance between two nodes and the upper bound of the capacity for parking in each car park. In Eq. (1),*dij*is the distance between nodes Pi and Pj, *Dup* is the upper bound of the distance and is a global parameter, *tj*is the number of spaces that are occupied at node Pj, and *Tup* is the upper bound of the capacity of the overall parking network and is a global parameter. We assume a network with seven nodes as in Fig. 5 and calculate the value of function F with _ D 0:2, _ D 0:8, D D2 km, T D 200 spaces: F12 D 0:36; F13 D 0:72; F21 D 0:44; F23 D 0:76; F27 D 0:34; F31 D 0:48; F32 D 0:44; F34 D 0:27; F37 D 0:42; F43 D 0:71; F45 D 0:36; F54 D 0:24; F56 D 0:44; F65 D 0:32; F67 D 0:36; F72 D 0:34; F73 D 0:74; F76 D 0:48. The neighbour table of each node with the F(α,β) function is shown in Fig. 6. Fig. 6 shows that the new neighbour table for each node follows Eq. (1). We will use this routing table in choosing the next node where to forward the user when a car park is full.

P1 Neighbour Table

Neighbour	F(α,β)	Distance (km)	Percentage of free spaces	Capacity (space)
P2	0.36	1.2	0.5	120
P3	0.72	1.6	0.7	200

P2 Neighbour Table

Neighbour	F(α,β)	Distance (km)	Percentage of free spaces	Capacity (space)
P7	0.34	1.8	0.6	120
P1	0.44	1.2	0.8	100
P3	0.76	2	0.7	200

P3 Neighbour Table

Neighbour	F(α,β)	Distance (km)	Percentage of free spaces	Capacity (space)
P4	0.27	1.5	0.3	100
P7	0.42	1.8	0.6	100
P2	0.44	2	0.5	120
P1	0.48	1.6	0.8	100

P4 Neighbour Table

Neighbour	F(α,β)	Distance (km)	Percentage of free spaces	Capacity (space)
P5	0.36	1.2	0.5	120
P3	0.71	1.5	0.7	200

P5 Neighbour Table

Neighbour	F(α,β)	Distance (km)	Percentage of free spaces	Capacity (space)
P4	0.24	1.2	0.3	100
P6	0.44	0.8	0.75	120

P6 Neighbour Table

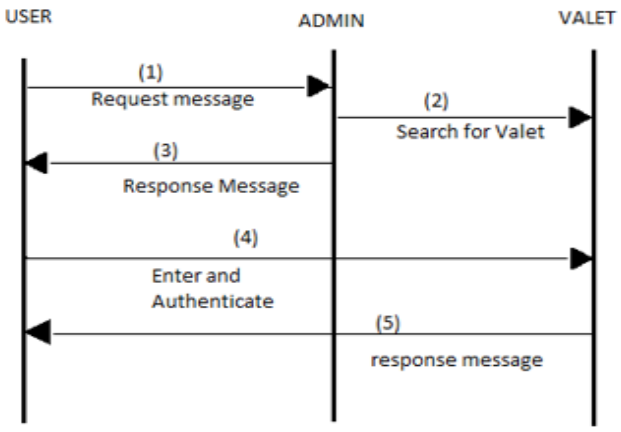
Neighbour	F(α,β)	Distance (km)	Percentage of free spaces	Capacity (space)
P5	0.32	0.8	0.5	120
P7	0.36	1.2	0.75	100

P7 Neighbour Table

Neighbour	F(α,β)	Distance (km)	Percentage of free spaces	Capacity (space)
P2	0.34	1	0.6	120
P6	0.48	1.2	0.75	120
P3	0.74	2	0.7	200

Reservation Process: Starting from (1) to (3) shown in Fig. 8, if the user is looking for a free parking space,he will send a request message to the system (1), which is done using a smartphone. When the system receives this request, it will find valet P1 with the least cost [minimum value of F(α,β)] and forward this message to the user. In this case, the least cost is the minimum value of function F(α,β). The value of F(α,β) is calculated as the distance (between the user and valet) and the number of valets at the location. If this valet is available, it will send a response message to the user (3). The response message includes the address of valet P1 and its directions. Because we use the percentage of total free valet in suggesting a new valet, a high probability of success exists in finding a available valet.

Entering Process: Starting from (4) going to (5), if a user arrives at the valet location P1; he must identify the valet using the given details (4).then the user will drop the car to the valet, and the count will increase by one. The system will send a response message to the user to notify successful parking (5). If the valet is currently unavailable, it will send a response message suggesting an alternative valet, including relevant information on new car park P2, with the least cost.



Valet booking process

IV SYSTEM ARCHITECTURE

Our project is an on-demand valet parking service that aims to optimise the process of parking cars. Through the app, a user can request a valet to come to a designated spot (within their service location), take their vehicle, and park it in an enclosed space till the patron needs it again. In the following figure we can understand the flow of events, such as customer using its application for booking the valet, setting the time and location based on which they will be charged. They may also opt for value added services at this time. After this they will be sent to the payment page from where the payment will be processed through the admin and sent to the valet service provider. Meanwhile the valet service provider will receive the request with time and location and act accordingly to assist the user to park their vehicle. Once the vehicle is handed over to the valet they will park the car is the nearest safe parking spot available and once the user demands its car back or as requested it will be dropped off to the user's desired location. Thus making this service fast convenient and affordable

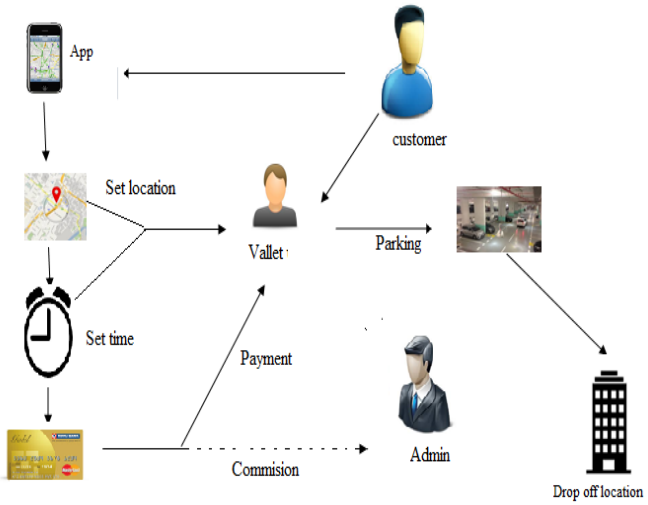


Figure 1 System Architecture

V TECHNOLOGIES USED

1. AES ALGORITHM

Step 1 : First round key [0]is added to the plain text then sub sequent rounds are performed which use same algorithm.

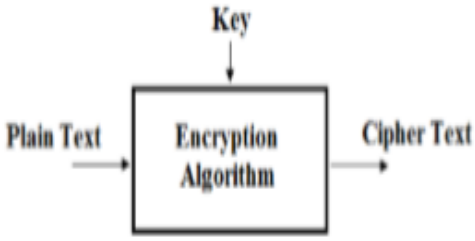
Step 2 :A simple substitution of each byte using a look up table is done. Each byte of is replaced by byte indexed by row (left 4-bits) & column (right 4-bits).

Step 3: Shift row layer : A circular byte shift in each row.

Step 4: Mix column layer :The Mix Column step is a linear transformation which mixes each column of the state matrix. Each byte is replaced by a value dependent on all 4 bytes in the column and is performed by the following multiplication.

Step 5: Add round key layer XOR current state with 128-bits of the round key.

Step 6: Key transformation/schedule :The 128 bit key is divided into four 32 bit words. These words are then further processed to produce key in each round



AES login parameters

2.KNN Algorithm

A case is classified by a majority vote of its neighbours, with the case being assigned to the class most common Amongst its K nearest neighbours measured by a distance function. If K = 1, then the case is simply assigned other class of its nearest neighbour.

Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

It should also be noted that all three distance measures are only valid for continuous variables. In the instance of Categorical variables the Hamming distance must be used. It also brings up the issue of standardization of the numerical variables between 0 and 1 when there is a mixture of numerical and categorical variables in the dataset.

Choosing the optimal value for K is best done by first inspecting the data. In general, a large K value is more Precise as it reduces the overall noise but there is no guarantee. Cross-validation is another way to retrospectively determine a good K value by using an independent dataset to validate the K value. Historically, the optimal K for most datasets has been between 3-10. That produces much better results than 1NN.

|| Volume 3 || Issue 4 || April 2018 || ISSN (Online) 2456-0774
INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH
AND ENGINEERING TRENDS

***Example*:**

Consider the following data concerning credit default. Age and Loan are two numerical variables (predictors) and Default is the target.

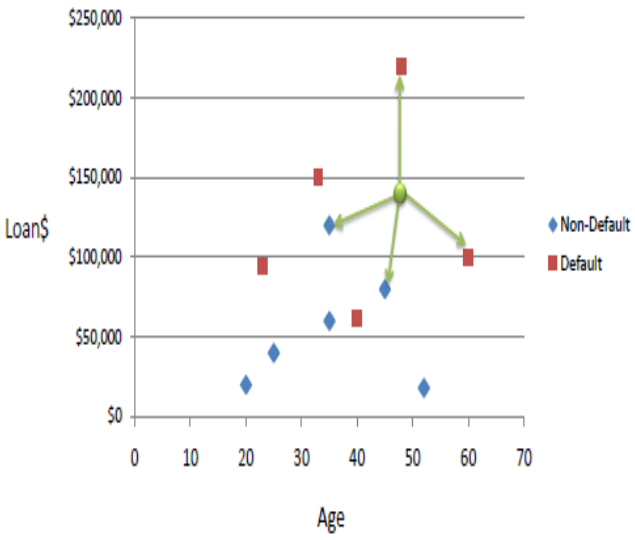
Hamming Distance

$$D_H = \sum_{i=1}^{k} |x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

X	Y	Distance
Male	Male	0
Male	Female	1



We can now use the training set to classify an unknown case (Age=48 and Loan=$142,000) using Euclidean distance. If K=1 then the nearest neighbour is the last case in the training set with Default=Y.

D = Sqrt[(48-33)^2 + (142000-150000)^2] = 8000.01 >> Default=Y

With K=3, there are two Default=Y and one Default=N out of three closest neighbours. The prediction for the Unknown case is again Default=Y.

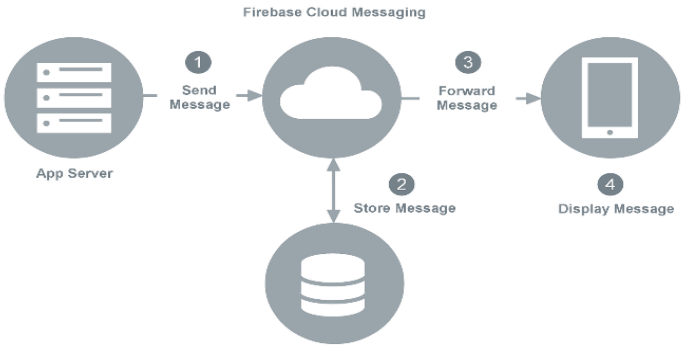
Age	Loan	Default	Distance	
25	$40,000	N	102000	
35	$60,000	N	82000	
45	$80,000	N	62000	
20	$20,000	N	122000	
35	$120,000	N	22000	2
52	$18,000	N	124000	
23	$95,000	Y	47000	
40	$62,000	Y	80000	
60	$100,000	Y	42000	3
48	$220,000	Y	78000	
33	$150,000	Y	8000	1
48	$142,000	?		

Euclidean Distance

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

2) Firebase Cloud Messaging

Firebase Cloud Messaging (commonly referred to as **FCM**), formerly known as Google Cloud Messaging (GCM), is a cross-platform solution for messages and notifications for Android, iOS, and web applications,which currently canbe used at no cost. The service is provided by Firebase, a company owned by Google. On 21 October 2014, Firebase announced it had been acquired by Google for an undisclosed amount.[2] The official Google Cloud Messaging website points to Firebase Cloud Messaging (FCM) as the new version of GCM.



VI IMPLEMENTATION

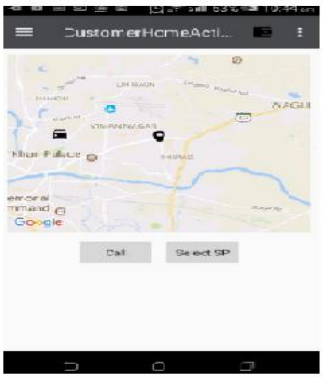
A. SOFTWARE SYSTEM

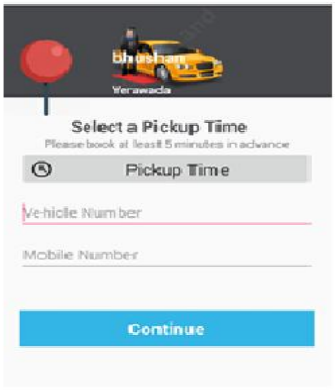
We designed a software client that runs on a smartphone based on the Android platform [8], which was built from the ground up to enable developers to create compelling mobile applications that take full advantage of all that a handset can offer. In this phase, we use the Android SDK Tools, which is a set of development tools used to develop applications for Android platform that can be used to write Android programs in the Android studio. Our cloud-based server is implemented on Firebase Cloud Messaging (FCM), and we use MySQL 5.5 as our database. MySQL is a SQLweb database, a distributed, scalable, and large data store. Fig. 1 shows the login interface of the system. Fig. 2 is a map describing the distribution of all available valets at a given location. In this map, there is one valet. Dynamic red cursor is the current location of the user. Fig. indicates the result returned by the system when users search for a valet following the shortest distance.

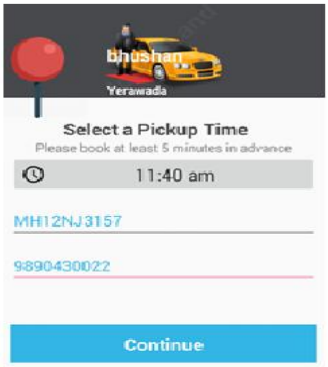
The result returned is P5 and the distance from the user to the valet is specified in the white area. Fig. indicates the result returned by the system when users search for a valet following our algorithms, here called the best case. The result returned is valet P2, the distance from the user to the valet is specified.

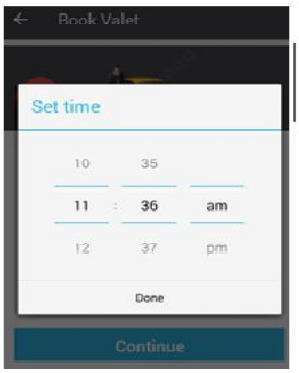
VII SCREENSHOTS OF THE APPLICATION

Available Valet, User Location and Booking process

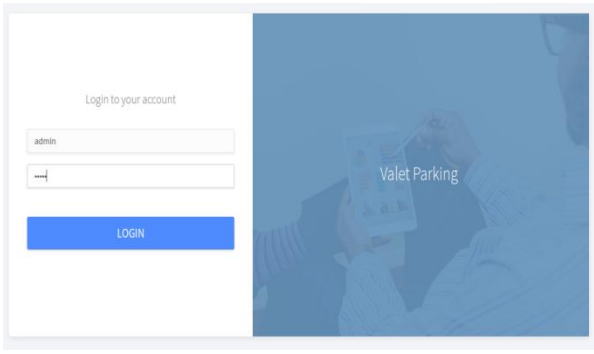


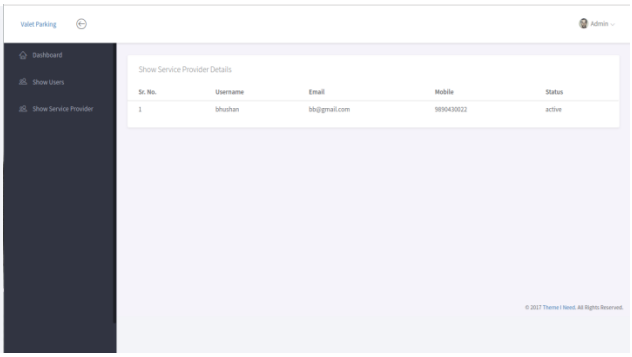


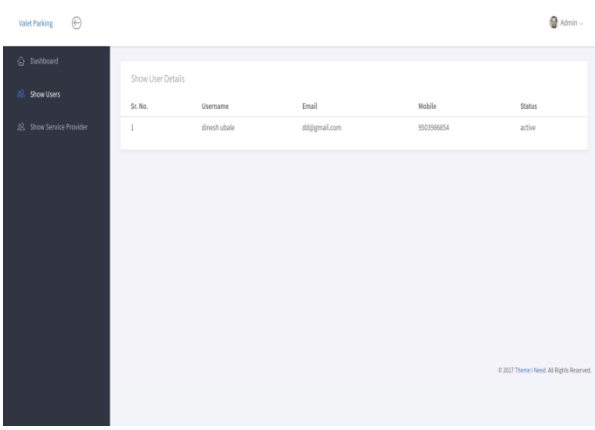


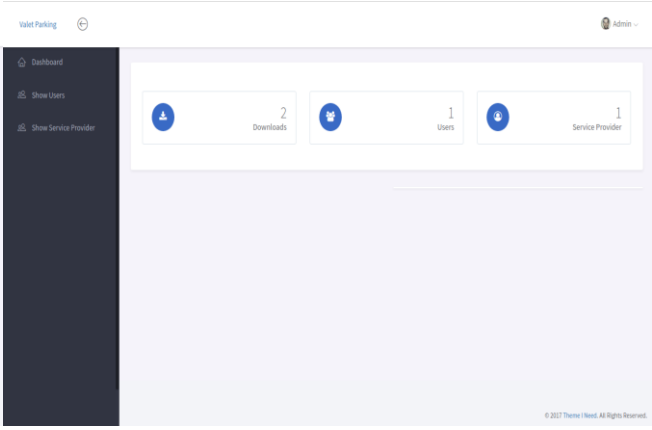


Admin Login & dashboard









VIII CONCLUSION

This study has proposed a On Demand Valet parking system that provides an easy to use android application and minimizes the costs of finding the parking. Our proposed architecture and system has been successfully simulated and implemented in a real situation. The results show that our algorithm significantly reduces the average waiting time of users for parking. The simulation of our system achieved the optimal solution when most of the users successfully found a valet. In our future study, we will consider the security aspects of our system as well as implement our proposed system in large scales in the real world

REFERENCES

1. Y. Geng and C. G. Cassandras, ``A new `smart parking' system based onoptimal resource allocation and reservations," in *Proc. 14th Int. IEEE Conf.Intell. Transp. Syst. (ITSC)*, Oct. 2011, pp. 979_984.
2. Y. Geng and C. G. Cassandras, ``New `smart parking' system based onresource allocation and reservations," *IEEE Trans. Intell. Transp. Syst.*,
3. vol. 14, no. 3, pp. 1129_1139, Sep. 2013.
4. X. Zhao, K. Zhao, and F. Hai, ``An algorithm of parking planningfor smart parking system," in *Proc. 11th World Congr. Intell. ControlAutom. (WCICA)*, 2014, pp. 4965_4969.
5. L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi, and R. Vergallo``Integration of RFID and WSN technologies in a smart parking system,"
6. in *Proc. 22nd Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM)*,2014, pp. 104_110.
7. C. W. Hsu, M. H. Shih, H. Y. Huang, Y. C. Shiue, and S. C. Huang,``Veri_cation of smart guiding system to search for parking space viaDSRC communication," in *Proc. 12th Int. Conf. ITS Telecommun. (ITST)*,2012, pp. 77_81.
8. R. E. Barone, T. Giuffrè, S. M. Siniscalchi, M. A. Morgano, and G. Tesoriere,``Architecture for parking management in smart cities," *IETIntell. Transp. Syst.*, vol. 8, no. 5, pp. 445_452, 2014.S.Ma,Y. Zheng, and O. Wolfson. T-share: A large-scale dynamic taxi ridesharing service. In IEEE le DE, pages 410-421, 2013.
9. UBER Get there: Your day belongs to you. Feb 2016. http://www.uber.com .
10. Zhan, S. Hasan, S. V. Ukkusuri, and C. Kamga, "Urban link travel time estimation using large-scale taxi data with partial information," Transportation Research Part C: Emerging Technologies, vol. 33, pp. 37–49, 2013.
11. L. Chen, A. Mislove, and C. Wilson, "Peeking beneath the hood of Uber," in Proceedings of the 2015 ACM Conference on Internet Measurement Conference , ser. IMC '15. New York, NY, SA: ACM, 2015, pp. 495–508.
12. L. Rayle, S. Shaheen, N. Chan, D. Dai, and R. Cervero, "App-based, on-demand ride services: Comparing taxi and ridesourcing trips and user characteristics in San Francisco," Tech. Rep., 2014.