

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Special Issue 4, April 2018

## Online Taxi Fleet

Deepak T<sup>1</sup>, Adline Freeda R<sup>2</sup>

B.Tech, Department of Information Technology, KCG College of Technology, Chennai, India<sup>1</sup>

Assistant Professor, Department of Information Technology, KCG College of Technology, Chennai, India<sup>2</sup>

**ABSTRACT:** The main aim of doing this project is to provide the efficient webapp for the users to take rides easier through taxi. This project is intended to help the drivers and users to make easy rides to travel from one place to another. Live updates are given to track their ride easier. To replace the phone call method web app are introduced to emerge more among people. This system will save time and it will be efficient. This project intends to introduce more user friendly in the various activities such as record updating, maintenance, and searching. The objective and scope of my project Online Cab Hiring System is to record the details various activities of use. A registration portal to hold traveller's details, monitor their transaction and used same to offer better and improve services to them. It increases the efficiency of the management at offering quality services to the customers.

**KEYWORDS:** Taxi Booking, Web Application, Fare Biding.

### I. INTRODUCTION

A web-based application is any application that uses a website as the interface or front-end. Users can easily access the application from any computer connected to the Internet using a standard browser. This contrasts with traditional desktop applications, which are installed on a local computer. With web-based applications, users access the system via a uniform environment—the web browser. While the user interaction with the application needs to be thoroughly tested on different web browsers, the application itself needs only be developed for a single operating system. There is no need to develop and test it on all possible operating system versions and configurations.

Business systems created by web applications can be securely accessed any time from any PC, tablet, laptop or smart phone that is connected to the internet. Storing all the data 'in the cloud' and accessing it via a web application means that this paper will have a peace of mind that it can very quickly be 'business as usual' if any of the computer equipment is damaged or stolen. Web applications store information on remote servers, so by knowing the URL (web address), user name and password, to log-in securely to any computer or mobile device connected to the internet and the business can be up and running again in no time. A web application can be customised to the own branding and have different user permission access levels. By only having features and functions which are relevant to the business, that reduces the amount of training time needed and can add additional functionality as in business grows. By using a web application this can avoid the hassle of installing software on every computer. Once a new version or upgrade is installed on the host server, all users can access it straight away. Customers who like to book a taxi for their ride, enters the website for booking. The first page of the booking collects all the required details of the passengers in the form. The information collected are pickup and drop off post-codes, number of passengers and luggage's, date and time of the journey. This information is collected to check various details for the journey. It is also used for getting journey quote for their desired rides without actually booking a ride. After the initial filling of the passenger details for the journey, the passengers are re-directed to the availability checking page. It is here in this page the passenger gets quotation for the specified journey by various cab or taxi providers. Here the passenger can browse through the list of providers to see the best fare for that journey. Also additional details like the cab type, the provider details are listed along with the book now button to book the cab. Once the passenger goes through all the details and can press the book now button near each provider. Now the passenger will be re-directed to another page where the customer needs to fill up the details in the form. This form contains all the details which are given by the passenger in initial submission of quote plus additional details like full address of the passenger, mobile and email address for communication, payment method.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Special Issue 4, April 2018

Here is the proposed SSSP algorithm inherits important design aspects of three existing algorithms. In a sequential implementation, Dijkstra's algorithm is efficient in terms of processing speed since it runs in time linear in the number of edges. However, the algorithm requires many phases (i.e., iterations) and it is less amenable to parallelization. In contrast, Bellman-Ford involves fewer phases and each phase is highly parallelizable. However, the algorithm may process each edge multiple times and is likely to incur high processing time. In this HTML5 and JavaScript sample application, PubNub provides the realtime communication capability that let the web application receive incoming events as the position of a remote object (latitude and longitude) changes. The core PubNub functionality that we use is Real time Messaging. This application uses a PubNub channel to send and receive position changed events. In this example, the same application both sends and receives the position events, so we say it is both the publisher and subscriber. In this application, the publisher (sender) system(s) may be different from the subscriber (receiving) system(s). This project uses the JavaScript Google Maps API and the HTML5 Map Widget for this tutorial. The HTML5 map widget displays a map of the vicinity configuration. The map size, map center latitude and longitude, zoom level, map style and other options may be configured to the initial preferences and updated on the fly. The Google Maps JavaScript API provides an initialization callback option that calls a function specified by user when the map is loading. In addition, it use features like Map Markers and Polylines which allows to place friendly map makers and flight paths on the map. All the points are user-specified and may be updated in real time as updates arrive.

## II. RELATED WORK

Taxi-booking services have recently gained attention to address congestion and sustainability. However, current booking services have a low success rate, due to uncooperative behaviours of passengers and drivers. In this paper, it propose a centralized Carrying Reservation System (CRS) to exchange real-time information for drivers and passengers and support a taxi dispatch mechanism to balance real-time supply and demand. Evolutionary game theory is applied to analyze behaviour and optimize the utility for taxi drivers and passengers. Global Position System (GPS) trajectory data from the Transport Commission of Shenzhen Municipality is used to evaluate the performance of proposed system. Results show that this model could reduce locating time as much as 46%, which will in turn lower passengers' waiting times. With the game theory model, drivers profit could be increased by over 18%. Compared with an off-line mining and online recommendation method (OMOM), This method improves the gain of drivers during non-rush hours. Having cooperated with taxi driver and passenger, this paper identifies several factors that lead to the low success rate of current booking services. Drivers are reluctant to bear the cost of picking up assigned passengers, mainly due to the risk of passengers breaking the appointment. Passengers are more willing to phone for taxi service.

1) during rush hours, and

2) when their location is remote and few taxis are accessible.

However, in both cases, taxi drivers are reluctant to pick up the passenger. Taxi drivers do not need to worry about booking passengers during rush hours and do not have enough motivation to pick up passengers from a remote place. In both cases, passengers are also liable to take a vacant taxi that is passing by. To account for this, this paper proposes that passengers' utility functions can best be represented by the expected waiting time.

Drivers prefer passengers that

1) that request a long-distance ride, or

2) whose destination area is one of the driver's preferred areas.

Both cases will decrease the time required to locate passengers, which could increase the revenue of drivers in turn. Thus, it formulate drivers' utility based on the current destination and the next trip after arriving at the destination.

It is important to note that drivers know where and when to locate passengers based on their experience. It defines  $E$  as the assertion of a quantitative measure of knowledge, which reflect preference for an area. Higher  $E$  means that driver will locate another passengers in the destination with a high probability. From the study, it appears that the most optimal situation for a driver is that a long-distance ride and destination is located within one of his/her favorite areas. After reaching the destination, drivers have to locate another passenger for the next driving, so a favorite area will decrease the expected time to locate passengers. Consequently, the utility function mainly contains three aspects: the time cost to locate passengers, the current trip and the next trip after reaching the destination. SSSP algorithm inherits important design aspects of three existing algorithms. In a sequential implementation, Dijkstra's algorithm is efficient in terms of processing speed since it runs in time linear in the number of edges. However, the algorithm requires many

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Special Issue 4, April 2018

phases (i.e., iterations) and it is less amenable to parallelization. In contrast, Bellman-Ford involves fewer phases and each phase is highly parallelizable. However, the algorithm may process each edge multiple times and is likely to incur high processing time. Meyer and Sanders proposed the  $\Delta$ -stepping algorithm, a trade-off between the two extremes of Dijkstra's and Bellman-Ford. The algorithm involves a tunable parameter  $\Delta$ . Setting  $\Delta = \infty$  (a sufficiently large number) yields the Bellman-Ford algorithm. On the other hand, Dijkstra's algorithm is equivalent to setting  $\Delta = 1$ . By varying  $\Delta$  in the range  $[1, \infty]$ , it gets a spectrum of algorithms with varying degrees of processing time and parallelism. All the three algorithms maintain a tentative distance  $d(v)$ , for each vertex  $v \in V$ . At any stage of the algorithm, the value  $d(v)$  is guaranteed to be an upper bound on the actual shortest distance  $d^*(v)$ . The tentative distance of the root vertex is initialized to 0 and for all the other vertices to  $\infty$ . As the algorithm proceeds, the tentative distance  $d(v)$  is monotonically decreased. The algorithms guarantee that, at the end of the process,  $d(v)$  matches the actual shortest distance  $d^*(v)$ . The tentative distances are modified by an operation called relaxation of an edge. When it reduces the tentative distance of a vertex  $u$ , which can possibly reduce the tentative distance of its neighbors as well. Given an edge  $e = (u, v)$ , the operation Relax( $u, v$ ) is defined as follows:

$$d(v) \leftarrow \min \{d(v), d(u) + w(u, v)\}.$$

### III. METHODOLOGY

Customers who like to book a taxi for their ride, enter the website for booking. The first page of the booking collects all the required details of the passengers in the form. The information collected are pickup and drop off post-codes, number of passengers and luggage's, date and time of the journey. This information is collected to check various details for the journey. It is also used for getting journey quote for their desired rides without actually booking a ride.

After the initial filling of the passenger details for the journey, the passengers are re-directed to the availability checking page. It is here in this page the passenger gets quotation for the specified journey by various cab or taxi providers. Here the passenger can browse through the list of providers to see the best fare for that journey. Also additional details like the cab type, the provider details are listed along with the book now button to book the cab.

Once the passenger goes through all the details he/she can press the book now button near each provider. Now the passenger will be re-directed to another page where he/she needs to fill up the details in the form. This form contains all the details which are given by the passenger in initial submission of quote plus additional details like full address of the passenger, mobile and email address for communication, payment method, etc. This module is reached after the booking is confirmed by the passenger. Here the admin first checks the availability of the drivers to allocate them for a particular journey. Admin checks the booking for the information such as location of the job, with that information the admin sorts a list of drivers in that location for checking whether the job is feasible to do at that particular time of availability of drivers. Once the drivers are sorted for that job, the admin puts that in the bidding bin in which the drivers can see the jobs for bidding. All or some of the drivers then bid for that job. The drivers bid for the job using a web application which is used by them for other purposes like logging into the dashboard, checking for a new job, invoice for the completed job, etc. Using this application the drivers bid with other drivers who also use the same web application for taking a new job. After the drivers bid for a particular job, the admin can see the details like which drivers have bid, for how much they bid, etc. Now the admin will allocate the drivers who have bid for a minimum amount for that particular journey so that the company can see more profits. The admin now chooses the driver and enters the amount he bid and presses the allocate button to allocate that particular job. Once the driver has been allocated for a particular job, now the details about the journey must be sent to both driver and passenger. This is done by the admin, the admin now sends the driver details to the passenger and passenger details to the driver for that particular job. Also an email notification is sent to the both parties regarding the journey for the confirmation. After the job details are sent to both passenger and the driver, the admin keeps track of that job for completion. Here various activities take place like confirming job, cancelling a job due to driver is of no-show or because of flat tyres. Also if the job is successfully completed the admin marks that job as completed for future references.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Special Issue 4, April 2018

## IV. ALGORITHM

### COUNT FOR ARRANGING THEM IN ORDER

1.  $d = 0$
2. If read  $\langle tr \rangle$  then  $d = d + 1$
3.  $RsTot = d$
4.  $CsTot = 0 // \text{Jum} \langle td \rangle = \text{count tag } \langle td \rangle \dots \langle /td \rangle$  in first tag
5. For  $i = 1$  to  $\text{jum} \langle td \rangle$
6. Read  $nilaics(i)$
7.  $CsTot = CsTot + cs(i)$
8. Next  $i$  ;

### COUNT FOR PAGINATION AND ROW-COLUMN SPAN

1.  $Datas(0) = 1$
2. while  $s = 1$  do
3.  $rsMax(0) = 1$
4. Count  $\text{jum}$
5. For  $i = 1$  to  $\text{jum}$
6. If  $rs(i) > 1$  then
7. If  $rs(i) \geq rsMax(i-1)$  then  $rsMax(i) = rs(i)$  else
8. If  $rs(i) < rsMax(i-1)$  then  $rsMax(i) = rsMax(i-1)$  Next  $i$  ;
9.  $Datas(s) = rsMax(i) + s - 1$
10. if  $Datas(s) < Datas(s-1)$  then  $Datas(s) = Datas(s-1)$  until  $Datas(s)$  ;
11.  $rowmax\_pro = Datas(s)$  ;

After obtaining the actual table size (row x columns) and the boundary row of the table property, then table extraction into a database is easy to do. Illustration of implementation of the approach can be implemented in many areas. For instance in manufacturing area, the main issues is to find a raw material. Currently many suppliers provide the information in Internet by using table form. Automotive industry need to have spare parts, such as tire, engine audio system. There are many products of audio system that inform by table in Internet. To find the appropriate spare parts the manufacturing will copy many tables from many suppliers of audio system. The next step, the content of table will merge to be one table. This effort is acceptable for number of data and sources are limited. If number of data and sources are huge, it can be hard effort and difficult to avoid an error. By implementation the two algorithms, the automatic process data harvesting process in table of Internet can be performed.

## V. CONCLUSION

In this paper the system present an application, that is used to book an online taxi to have an efficient journey on the roads. The driver and passenger are given equal importance and handled carefully to get the efficient result. The bidding system for driver and job crabbing system helps them to have a good understanding with the system. Passengers are assisted with the live chat system to clarify the queries as soon as possible. Also the admin having the overall control to maintain the system good and efficient.

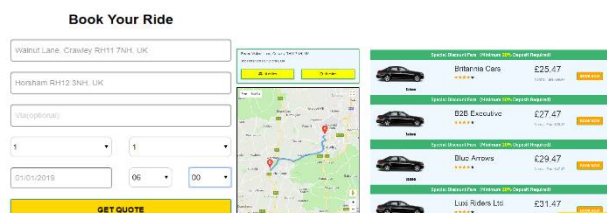


Figure No.1 Output of the Implementation

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 7, Special Issue 4, April 2018

## REFERENCES

1. Embley D. W., Hurst M., Lopresti D., Nagy G. 2006 Table-Processing Paradigms : a Research Survey International Journal of Document Analysis, page 66-86
2. Gultom R., Sari R. F., Budiardjo B. 2011 Proposing the new Algorithm and Technique Development for Integrating Web Table Extraction and Building a Mashup Journal of Computer Science 7 (2), page 129-142
3. Krupl B., Herzog M., Gatterbauer W. 2005 Using Visual Cues for Extraction of Tabular Data from Arbitrary HTML Documents Proceeding WWW '05 Special interest tracks and posters of the 14th international conference on World Wide Web
4. Lerman K., Knoblock C., Steven M. 2001 Automatic Data Extraction from Lists and Tables in Web Sources Proceedings of Automatic Text Extraction and Mining Workshop ({ATEM}-01)
5. Lin J., Wong J., Nichols J., Cypher A., Lau T. A.2009 End-User Programming of Mashups with Vegemite Proceedings of The 13th International Conference on Intelligent User Interfaces.
6. Liu Y., Mitra P., Giles C.L. 2008 A Fast Pre processing Method for Table Boundary Detection : Narrowing Down the Sparse Lines using Solely Coordinate Information The Eighth IAPR International Workshop on Document Analysis Systems, page 431 – 438
7. Tengli A., Yang Y., Ma N. L. 2004 Learning Table Extraction from Examples Proceedings of The 20th International Conference on Computational Linguistics
8. Wong W., Martinez D., Cavedon L. 2009 Extraction of Named Entities from Tables in Gene Mutation Literature Proceedings of The Workshop on Current Trends in Biomedical Natural Language Processing, BioNLP '09.
9. H. Wang, "Improving taxi dispatch services with real-time traffic and customer information," Ph.D. dissertation, National University of Singapore, 2004.
10. <http://ztzrb.hangzhou.com.cn/system/2013/01/10/012277204.shtml>.
11. <http://news.mediainan.com/anhui/2012/0505/1143638.shtml>.
12. <http://news.yntv.cn/content/15/201205/24/15553624.shtml>.
13. <http://cd.qq.com/a/20120611/000004.htm>.
14. J. Yuan, Y. Zheng, L. Zhang, and X. Xie, "T-finder: A recommender system for finding passengers and vacant taxis," IEEE Transactions on Knowledge and Data Engineering (TKDE), 2011.
15. S. Ma, Y. Zheng, and O. Wolfson, "T-share: A large-scale dynamic taxi ridesharing service," in 29th IEEE International Conference on Data Engineering, 2013.
16. J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun, "Where to find my next passenger ?" in The 13th ACM international conference on Ubiquitous computing, UbiComp 2011.
17. H. Yang, S. C. Wong, and K. I. Wong, Demand-supply equilibrium of taxi services in a network under competition and regulation. Transportation Research, 2002, vol. Part B 36, no. 799-819.
18. C. F. Manski and J. D. Wright, Nature of Equilibrium in the Market for Taxi Services. Transportation Research Record, 1976, vol. 619, no. 11-15.
19. D. H. Lee, H. Wang, R. L. Cheu, and S. H. Teo, "A taxi dispatch system based on current demands and real-time traffic conditions," in Transportation Research Record, vol. 1882, 2004, pp. 193-200.
20. P. D. Taylor and L. B. Jonker, "Evolutionary stable strategies and game dynamics," Mathematical Biosciences, vol. 40, pp. 145 – 156, 1978.