AUTOMATIC TRANSPORTATION SYSTEM

Kirti Rathi[1], Bhumika Gupta[2], Anushka Aggarwal[3], Geetika Chhabra[4]

[1] *Student, ECE, BVCOE, New Delhi, India*
[2] *Student, ECE, BVCOE, New Delhi, India*
[3] *Student, ECE, BVCOE, New Delhi, India*
[4] *Student, ECE, BVCOE, New Delhi, India*

Abstract

In today's era need arises to develop a cost effective transportation system which would allow people to travel from one place to another in minimum time period. In this paper, we have proposed one such wireless locomotive. The locomotive will change its path dynamically [1], based upon stations to be traversed, as per passengers present on respective stations. It comes up with an advantage that it will go to only those stations where passengers are present, thereby saving time and energy and hence being cost effective. The simulations are carried out using python code, local server and an android app. This locomotive makes sustainable use of resource. The proposed automatic transportation system is an intelligent transportation system. It works without any manual assistance which means it is a driverless transportation system.

Key Words: *Android Application, Server, Python, Smart Locomotive*

1. INTRODUCTION

Transportation recently used is schedule based, irrespective of whether passengers are there or not at different station. Present systems have a fixed route & the transport (locomotive) runs accordingly [1]. But the system that we proposed is used for short distances where the locomotive instead of running on same path changes its path according to the availability of passengers on different stations which further save the time of passengers and other resources. In [2], authors have proposed a locomotive which works on Global Positioning System. The use of a server for deciding the path of the locomotive has been a suggested in [3]. A fast and efficient method of transportation system for short distances has also been proposed in [4].

2. PROPOSED METHOD

The principle behind the proposed automatic transportation system (locomotive) is basically that we are sorting all our stations in a manner such that our locomotive covers shortest distance and still be able to cover all the stations on which passengers are present and waiting to board. The list of stations which will be sorted in our program will be made based upon those stations on which passengers are present and wish to board. Moreover to this list we will also be adding the destination station of the passengers and sort the list again to calculate shortest path for passengers to board and deboard in a single go. The process of sorting will be continuous and will be performed on reaching every station so that shortest path can be calculated from that particular station. The list of current station of the passenger and the destination station of the passenger will be made in the database of our local server (wamp). The list will be made dynamically, as and when passengers enter their current station, through an android application. The process of sorting the stations to calculate the shortest path will take place in the python simulator as python program receives data from our local server.

2.1 METHODOLOGY

The operation of the smart transportation systems has been shown in the figure 1. The system will start when a passenger inputs the data of their current station and their destination station using an android application. Data from the android application is sent to a local server which is a private server

designed only for the transportation system. The local server collects all the data and arranges it into a database such that it makes two lists one of the current stations of the passengers and the other one for the destination stations of the passengers. The data from the local server's database is sent to the python simulator. Here the python code calculates the shortest path that the locomotive shall be covering based upon the two lists of data sent by the local server. And then it shows the simulation of how the locomotive will be moving along the shortest path.

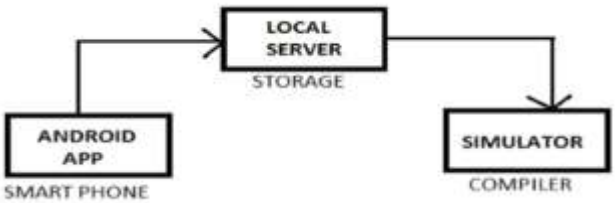

Fig 1 : Methodology

2.2 WORKING OF THE ANDROID APPLICATION

The passenger will use the android application to enter the current station which is the station from where the passenger wants to board the locomotive. Figure 2 shows the front panel of the application. Suppose that the current station by default is set to be station 1, which means the locomotive will start its motion from station 1 each time and come back to the same station. This data is sent to the wamp local server where it stores the data in form of a list.

Fig 2 : Front panel of Android Application

The data is sent to the python simulator from the wamp server. Thereafter sorting of the list of stations is performed. Accordingly, the next station is calculated. And the simulation goes from station 1 to the next station. During this process simulation goes to sleep. After this the user is asked to enter the destination that is the station where the passenger wants to de-board. This data goes to the database of wamp local server and the list stored there is updated accordingly. This updated data is sent to the python simulator where the process of sorting is done again. And the next station is calculated. Further the data is stored in the wamp database which is then received by the android application. The python simulator requests a call from the wamp server to send the list of stations.

After receiving this data, the python code calculates the next station by sorting the stations according to the shortest path.

3. SIMULATION & RESULTS

To validate the working of the locomotive, we had considered 4 different stations & a python code was written. After interfacing the python code, android application & wamp server inputs were taken from the passengers, through android application, on different paths & observe the paths the locomotive is taking according to the availability of passengers on different stations. The simulated output observed was that, the locomotive took the shortest path according to availability of passengers on different stations. This has been depicted in the figure 3.

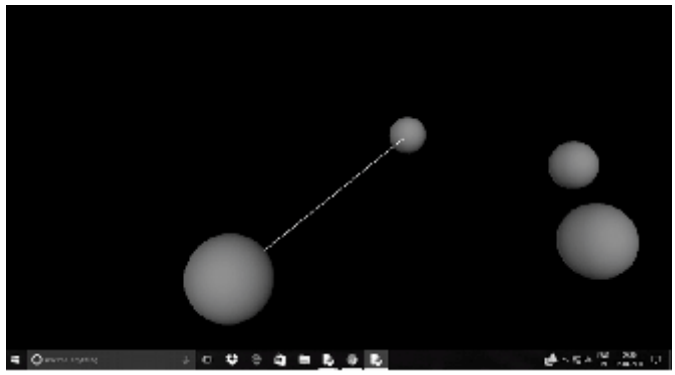

Fig 3: Simulated Output

4. CONCLUSION

The simulations show how the proposed transportation system will work in real. The current station and the destination of the passenger will be input taken from the passenger through an android application. This application will be connected to a

local server which will store this data. The local server designed here is a private server made only for the use of the transportation system. The data stored in the local server will be retrieved by the program and it will show through simulations as to how the locomotive is to calculate and traverse the shortest path. Every passenger will use an Android App which will take current location and destination of the passenger. The benefit is that if 5 stations are chosen out of 10 stations then the locomotive will traverse the shortest path so that it doesn't have to visit all 10 stations. Hence the proposed smart Locomotive transportation system proves beneficial both in time saving as well as resource conservation.

REFERENCES

[1] Zhenghao Chen, Jianlong Zhou, Xiuying Wang, Jeremy Swanson, Fang Chen, Dagan Feng, "Neural net-based and safety-oriented visual analytics for time-spatial data", Neural Networks (IJCNN) 2017 International Joint Conference on, pp. 1133-1140, 2017, ISSN 2161-4407.

[2] Jakub Młyńczak, Rafał Burdzik, Ireneusz Celiński, Springer Proceedings in Mathematics & Statistics, vol. 182, pp. 279, 2016, ISSN 2194-1009, ISBN 978-3-319-42407-1.

[3] Yu-Chia Hsu, Chih-Sheng Wang, Shih-Fan Chou, Ai-Chun Pang, Pi-Chen Chiu, Shu-Jye Chang, "Design and implementation of Taxi advertisement system", Electrical Engineering and Computer Science (ICEECS) 2014 International Conference on, pp. 1-6, 2014.

[4] Jecinta Kamau Andrew Rebeiro-Hargrave Nazmul Hossain Zahidul HosseinH i r o s h i OkajimaAshir Ahmed "Hybrid Multiservice Demand Responsive Mobility Service for Developing Countries" Proceedings of eChallenges e-2015 Conference pp. 83 2015.