Intelligent Solution to Urban Mobility Problems

Lester Gregory Rodrigues
Department of Computer Engineering
St. Francis Institute of Technology
Mumbai, India
rodrigueslester2001@gmail.com

Domingos Julio Pinto
Department of Computer Engineering
St. Francis Institute of Technology
Mumbai, India
domingos.cena@student.sfit.ac.in

Somnath Pratik
Department of Computer Engineering
St. Francis Institute of Technology
Mumbai, India
somnathpratik19@student.sfit.ac.in

Prof. Shamsuddin S. Khan
Department of Computer Engineering
St. Francis Institute of Technology
Mumbai, India
shamsuddinkhan@sfit.ac.in

Joel Dsouza
Department of Computer Engineering St.
Francis Institute of Technology
Mumbai, India
dsouzajoel106@student.sfit.ac.in

***Abstract*—In this era of rapid urbanization fostering faster, efficient and cheaper transportation options for common people as well as organizations has become a crucial concern all over the world. The proposed solution solves commuting problems faced by citizens and improves the efficiency and effectiveness of an organization's logistics operations. The users would be able to find the optimal route to travel depending on their location constraints and thus plan their journeys effortlessly. The administrators of logistics firms could manage their fleets, know where the packages are at all times and keep a track on their goods delivery partners. Customers would get the delivery status, the anticipated delivery time, delivery partner's details. Thus, the application would act as a bridge between the organization and the customers. JHipster, Angular, Ionic, OptaPlanner, and Leaflet are used together to create sophisticated and optimized progressive web applications with a seamless user experience across multiple platforms and devices. The proposed solution in the paper addresses urban mobility issues that hinder convenient and seamless movement, promoting quick and efficient transportation of people and goods.**

***Keywords*— Smart Mobility, efficient transportation, optimal route, progressive web application**

I. INTRODUCTION

Urban mobility encompasses all aspects right from transportation modes, walking, to proximity of services to people. The two categories into which Urban Mobility is classified into are: individual transportation, freight transportation. Individual mobility is subjected to the various decisions which the individual may have to take while travelling. This again is dependent on various rationales. Whereas freight transportation is dependent on the customers, cargo companies and the transportation service providers. In the recent years the way urban mobility is being perceived has changed tremendously. Earlier People's needs were limited to accessing urban areas for jobs, education, entertainment etc. Now Urban mobility is not limited to moving individuals from source to destination. The quick and efficient accessibility to these urban mobility services is also a matter of prime concern.

When the transport systems are not capable to cater to this fundamental idea of urban mobility, plenty of difficulties and problems arise. Delay due to traffic congestion, longer commuting, high maintenance costs, environmental impacts and energy consumption, efficient freight distribution are some of the key challenges of urban mobility. The approach presented in this paper aims to solve some of the problems that disrupt convenient transportation and provide a pleasant transportation experience to the system's user.

The evolution of the cities is a product of various public policies and private initiatives driven by various interests. An efficient urban mobility solution would ensure transportation that is more sustainable, optimum and universal thus raising the standard of living of the people, minimize transportation costs and ensure more areas evolve into being identified as urban. This has been a driving force for the proposed solution. The system makes an effort to make transportation easier for common people who live in and around us.

The proposed solution uses JHipster as the development platform for the project which combines several popular technologies like Angular, Spring Boot to create modern web applications and provides a robust backend. For an attractive user interface and for ease of building of the application Ionic and Angular frameworks are used. Ionic and Angular frameworks give an edge in cross-platform compatibility making it a perfect choice for building mobile applications. For solving the vehicle routing problem OptaPlanner is used. OptaPlanner utilizes optimization techniques to find optimized solutions for complex vehicle routing problems, including various constraints such as time windows, capacity limitations, and vehicle availability. OptaPlanner can help organizations save time and money by optimizing their fleet's routes and improving their operational efficiency. With the use of Leaflet and OpenStreetMap there is flexibility and control to create customized maps that meet the specific needs of the application

II. PROBLEM DEFINITION

By virtue of a progressive web application, the aim is to provide real time fully personalized transportation guidance for individual and freight transportation considering constraints like location and ensure efficient mobility and without unnecessary delay; thus, ensuring prime experience for all the stakeholders involved. People are struggling with matters like determining the best route to reach their destination and getting there within the shortest amount of time possible. The user should be able to put his or her constraints of location and must be recommended an optimal route considering these constraints along with the real time traffic conditions. A logistics company must be able to manage it's vehicles and enable its goods delivery partners to utilize the fastest routes for efficient delivery of products.

III. LITERATURE SURVEY

Paper [1] tries to focus on different types of algorithms that are available, gives their drawbacks and based on these drawbacks it proposes new path planning algorithms. Initially, it gives a walk through regarding the traditional algorithms for path planning and dynamic algorithms for path planning. It also mentions that these algorithms do not consider when large amounts of requests are sent to these algorithms all at once. For example, let's say if we take the shortest path from Point A to Point B. There are 3 possible paths, one is 15 kilometers, second is 10 kilometers and the third one is 12 kilometers. It's easy to identify that the shortest path is 10 kilometers. But when we actually look at that path it is the peak path that most people use and overcrowded most of the time. So not only choosing the shortest path, time efficiency also plays a major role. This paper gives importance to time. Traffic path planning algorithm based on data prediction (TPPDP) and Traffic path planning algorithm based on data prediction with load balancing (TPPDP-LB) are proposed in this paper. These two algorithms with the help of past and present data would help to plan optimal routes. They also help to create hot zones that show crowded areas. This paper proves to be the base for deciding the optimal path in the proposed solution.

Paper [2] focuses on how efficient the services of metro stations are and how they can be increased. The service levels of a metro station were indicated by two important parameters. These two parameters for measuring the service levels of the metro stations in this research were passenger's walking time and passenger's waiting time. The research was conducted by analyzing different types of passengers by using their Smart Card Data. These various types of passengers were categorized into different categories based on 3 different factors that were considered namely the passenger type, travel time and type of cardholder. The waiting time for a passenger at a particular metro station was mostly related to the adjacent train arrival time and the volume of passengers present there at a certain time interval. The evaluation would help to assign passengers to trains and help to increase the operating efficiency of trains especially at peak hours. Thus, the data collected from the passengers' cards and its analysis can help to optimize the timetables at the metro stations.

Paper [3] talks about the importance of big data in transportation of smart cities. Big Data Analytics can be used to find traveling patterns of people within the city. This in turn can help providers for public transportation to make better business-related decisions regarding the service quality for the passengers who will be traveling through these public transportation facilities within the city. This paper has also stated that through mobile phone data triangulation from millions of people and anonymous users, these traveling companies can predict those who travel frequently. The traffic on the roads can be determined by using the Big Data approach since tons of data will be available to the data analyst and using one of the big data associated algorithms, he or she, on behalf of the transportation company, can analyze and predict the future traffic that would occur due to various reasons. This paper also makes one aware about the direct influence it has on supply chain management. The underlying sensor technologies that can be embedded in different vehicles have been mentioned in this paper. Gaining data from these sensors and using this data analysis for ease of transportation from a business perspective and normal user perspective contributes to smart mobility.

Paper [4] gives information about the development of an intelligent system that gives information of vehicles on a real time basis which focuses on achieving high standards of performance. This system would help in fleet management for a transportation company as well as logistics services company since it will give information where their vehicle is at the moment on a real time basis. This system would allow all the functions that a fleet management system should have that too on a real time basis. The real time data that it receives is processed that helps in building a service that's highly reliable. Initially, by giving informative knowledge regarding them, this paper compares the previously known fleet management or fleet tracking systems. It also shows how the typical UI of these systems look. Thereafter this paper talks about the system requirements they needed to build their system. It also gives information on how this system was being developed and how this system was being implemented. Finally, this paper showed the experimental results of this system that was built. In the end this paper believes that even though there are many such fleet tracking systems available in the market, there's a need of optimization according to the need of the specific category of users which is also known as customization according to the user.

Paper [5] solves real time vehicle routing problems by using parallel computing strategies. This paper explains in detail what is a vehicle routing problem and the dynamic factors that contribute to vehicle routing problems such as customer requests. It also indicates where the significance of this problem can be seen. For example, it can be seen to a great extent in fleet management, logistics and transportation. It gives ideas regarding the various different approaches that are available to solve the vehicle routing problem especially when it's dynamic. This paper also gives information regarding what happens if the system is more towards the static side and what happens if the system believes more towards a dynamic approach. It gives detailed information regarding the requirements and setup for the experimental process. Thus, to solve this vehicle route problem, this paper uses the Tabu search algorithm. However, this algorithm doesn't consider the future predictions and handles current predictions only. In the end this paper believes that the vehicle routing problem can be solved much faster using parallel computing rather than a single processing environment.

IV. PROPOSED SOLUTION

The following is the proposed solution for the problem discussed. The project is an intelligent transportation system consisting of an administrator application, user application and a driver application (for the delivery partner). This model could be scaled to business to business or government to citizens point of view as per the client's requirements. The primary users of the application are general public and a logistics company in need of a transportation guide. The application is developed as a progressive web application so that it is accessible anywhere and can be accessed by anyone without any strict hardware requirements

For the users using the travel planning module, OptaPlanner is used to solve the vehicle routing problem. OptaPlanner uses a range of algorithms to solve the problem, including constructive heuristics, metaheuristics, and local search algorithms. Constructive heuristics build feasible solutions from scratch by iteratively adding customers to routes, while metaheuristics like simulated annealing and genetic algorithms explore the solution space to find better solutions. Local search algorithms like tabu search and simulated annealing are used to refine solutions by making incremental changes to improve the objective function value. OptaPlanner also employs a variety of optimization techniques such as pruning and parallel processing to speed up the search process and find optimal or near-optimal solutions to the vehicle routing problem. The user is asked the source and destination to be given as input. For constraints, the budget and the time requirements would are also considered. The user is also be asked to input his or her preferred mode of transportation. The best path considering the input constraints is shown. The traffic status is also be considered for determining the optimal estimated time of arrival for reaching the location. Through APIs the user is directed to appropriate websites for ticket booking purposes. The project also includes starred routes features, so that users can access the routes which they travel on a frequent basis in a quick manner. For logistics company the primary objective is to manage the company's fleet of vehicles, drivers and optimize delivery. For the goods to be delivered to a location priority is given on the basis of hard and soft constraints like perishable things should reach on time, certain things have lesser priority etc. Then the routes are determined by OptaPlanner which would deliver the most important goods first and so on. The driver's location is tracked via the administrator application so that the company can determine where the driver is and get real- time updates about the delivery status. The estimated time of delivery would thus be determined by OptaPlanner.

Company Administrator has 2 more modules i.e., fleet management and package management. In the fleet management system, initially he or she enters the fleet details. He or she can modify and view them as well. In the package management system, the administrator gets the package details, the origin of the package and destination as filled by the user. The administrator can assign the package to the driver based on the availability. Thereafter the delivery partner gets the package details, optimal path and real time details. Optaplanner would help to solve the vehicle routing problem thus giving an optimal path to driver and company administrator. The customer would get the delivery status at all times. The delivery partners get the list of jobs assigned to them, they can check their work statistics. They can check and update the status of a particular delivery at all times.

Following is the flow of the proposed solution:

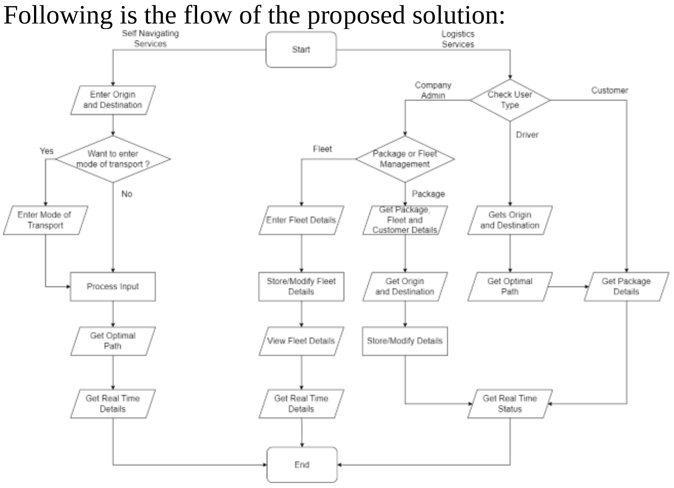


Fig. 1. Flow of the Proposed Solution

Following is the description about the methodology used to solve the problem statement's key areas and achieve the objectives decided:

For getting the optimal route to travel:

OptaPlanner:

Optaplanner is an open-source Java-based optimization engine that helps developers solve complex planning problems by finding the best possible solution in a reasonable amount of time. Optaplanner provides a flexible, configurable, and extensible framework that can be integrated into various types of applications. It also supports real-time planning, which allows for quick and dynamic changes to plans as needed. Following is the explanation of how it was used in the solution.

The user can put constraints like source, destination, budget, time, preferred mode of transportation, and an optimal path is shown. The steps taken to achieve this are mentioned below:

1. Define the problem domain:
   In order to use Optaplanner to solve a problem, the problem domain needs to be defined. This includes defining the entities, constraints, and optimization goals of the problem.

2. Implement the Optaplanner solver:
   The Optaplanner solver is responsible for finding a solution to the problem domain defined in step 1. This involves defining the solver configuration and implementing the solver algorithm using Java programming language.

3. Integrate the Optaplanner solver with JHipster:
   JHipster is used to develop the user interface and business logic of the application. In order to integrate Optaplanner with JHipster, a REST API is implemented that communicates with the Optaplanner solver. This involves defining the API endpoints, request and response formats, and handling the communication between the two systems.
4. Implement the user interface:
   The user interface is implemented using the Ionic and Angular frameworks. This involves defining the UI components, views, and controllers to allow users to input constraints and display the optimal path.

5. Implement the driver constraints:
   For the driver, goods are given priority based on location and constraints such as whether the good is perishable or not. This involves implementing the driver constraints in the Optaplanner solver and the REST API.

6. Test and deploy the application:
   The application is tested to ensure that it functions as expected and meets the requirements. Once it passes the testing phase, it is deployed to the production environment for users to access.

For getting the live location:

1. Request permission to access location data:
   In order to collect the delivery partner's location data, the application needs to request permission to access their location data. This is done using the HTML 5 Geolocation API's navigator.geolocation method.

2. Implement the getCurrentPosition() method:
   The getCurrentPosition() method is used to retrieve the delivery partner's current location. This involves defining a success call back function that retrieves the latitude and longitude data and saves it to variables.

3. Define the Leaflet map:
   The Leaflet map is defined in the front-end code. This involves creating a new L.map() object and setting the view to the delivery partner’s current location.

4. Create a marker:
   In order to display the delivery partner’s location on the map, a marker needs to be created. This can be done using the Leaflet L.marker() method.

5. Add the marker to the map:
   Once the marker is created, it needs to be added to the map using the Leaflet marker.addTo(map) method.

6. Update the marker position:
   As the delivery partner's location changes, the marker position needs to be updated in real-time. This can be achieved by continuously retrieving the driver's location using the getCurrentPosition() method and updating the marker position using the Leaflet marker.setLatLng() method.

For tracking of the goods delivery partner’s live location:

1. Implement Leaflet and Open Street Map:
   Leaflet is a popular open-source library for interactive maps, and Open Street Map is a free and open-source map provider. In order to use Leaflet and Open Street Map in the application, the libraries are implemented in the front-end code. This involves installing the libraries, defining the map views, and adding markers and layers to the map.

2. Collect their location data:
   In order to live track the delivery partners, their location data is collected.

3. Publish location data to Apache Kafka topic:
   Apache Kafka is a distributed streaming platform that is used to handle high volumes of data in real-time. The partner's location data is be published to a Kafka topic using a Kafka producer. This involves defining the topic, serializing the data, and sending it to the Kafka cluster.

4. Consume location data from Apache Kafka topic:
   The administrator application needs to consume the partner's location data in real-time. This is be done using a Kafka consumer. The consumer needs to subscribe to the Kafka topic, deserialize the data, and process it.

5. Display their location on the map:
   Once the location data is consumed by the administrator application, it can be displayed on the map. This involves adding a marker or layer to the map to show their location.

6. Update the location in real-time:
   As their location changes, the location data needs to be updated in real-time. This isa chieved by continuously publishing their location data to the Kafka topic, and the

administrator application consumes the latest data to display the updated location on the map.

V. IMPLEMENTATION

The project is implemented in the following steps:

1. Information gathering and analysis of existing systems:
   As the initial step, the stakeholders' expectations were understood and were defined. The existing systems were evaluated on the basis of user friendliness, relevance of features and ease of access. This gave us a clear idea about the missing loopholes and how our project could bridge the gap.

2. Designing the Data Models Database:
   After understanding about the application's scope and the loopholes in the existing systems we developed the data models which comprises of the entities used in the JHipster Domain Language (JDL) file. This file is an important part of the process as the project uses the JHipster Framework. After the designing of data models, the database is designed using JDL(jhipster domain language).

   A JDL (JHipster Domain Language) file is a file format used to define the domain model of a JHipster application. This file can be used to generate the code for entities, services, and repositories using the JHipster code generator. To create a JDL file, you need to define your application's domain model using a specific syntax. The syntax of the JDL file is similar to the YAML format and is used to define entities, their fields, and their relationships with other entities.

3. Creating the interface of the application:
   Once you have created your JDL file, you can use the JHipster code generator to generate the code for your entities, services, and repositories. The generator will read the JDL file and create the necessary files for your application's domain model. Next we run npm install -g generator-jhipster-ionic and also npm install. We need to follow the prompts to configure the Ionic app, including choosing an app name, theme, and other options. We then check if all the entities are generated in the back end accurately.

4. Creating the functionalities of the system:
   The functionalities are developed following the procedures mentioned in the proposed solution. Data will be stored in PostgreSQL which is a free and open-source relational database management system emphasizing extensibility and SQL compliance.

   Functionalities for the user:
   1. Plan their journey entering their source, destination and other constraints and getting optimal path.
   2. Send a parcel by placing an order.
   3. Locate the driver.
   4. Get delivery tracking of the parcel.

   Functionalities for the delivery partner:
   1. Get optimal path for pick up and delivery
   2. View their work statistics
   3. Get list of jobs assigned
   4. Update the status of the jobs assigned

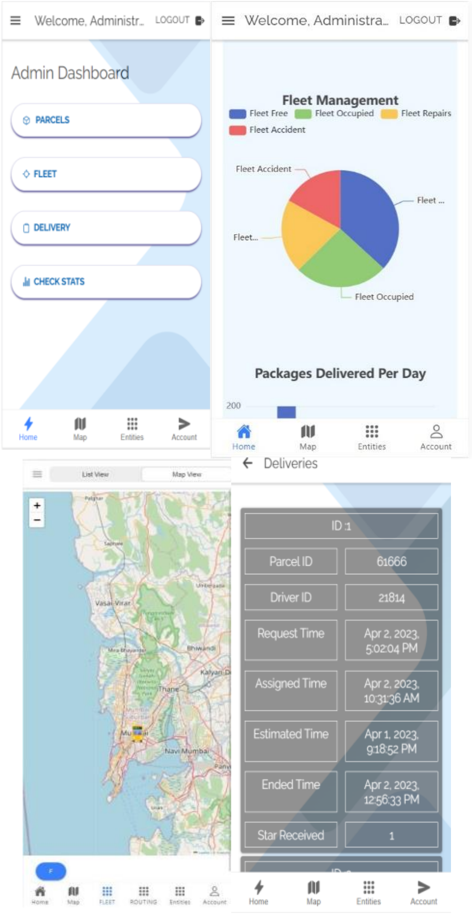
   Functionalities for the administrator:
   1. Have a dashboard with various statistics displayed graphically as per the real-time data
   2. Track the live location of the delivery partners

5. Testing and Deployment:
   The application is tested and then deployed on the suitable cloud platform.

VI. RESULTS

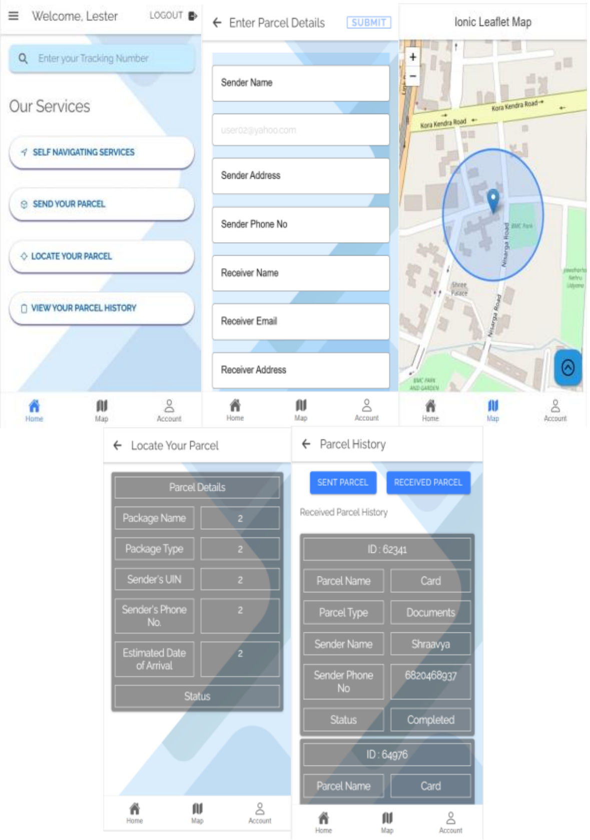
The developed solution is a comprehensive system that provides end-to-end management of the transportation and logistics domain. The system optimizes the delivery of goods and services, enabling logistics companies to prioritize deliveries based on the hard and soft constraints of the delivery. The system ensures the real-time tracking of the driver, maintaining transparency and accountability, and providing users with the track parcel history for review purposes. The developed solution delivers a robust, efficient, and user-friendly system that has the potential to revolutionize the transportation and logistics domain.

The images below are of the implementation done of the proposed solution.



ID 1	
Parcel ID	61666
Driver ID	21814
Request Time	Apr 2, 2023, 5:02:04 PM
Assigned Time	Apr 2, 2023, 10:31:36 AM
Estimated Time	Apr 1, 2023, 9:18:52 PM
Ended Time	Apr 2, 2023, 12:56:33 PM
Star Received	1

Fig. 2. Administrator application Dashboard



Sender Name
user02@yahoo.com
Sender Address
Sender Phone No
Receiver Name
Receiver Email
Receiver Address

Parcel Details	
Package Name	2
Package Type	2
Sender's UIN	2
Sender's Phone No.	2
Estimated Date of Arrival	2
Status	

ID : 62341	
Parcel Name	Card
Parcel Type	Documents
Sender Name	Shraavya
Sender Phone No	6820468937
Status	Completed
ID : 64976	
Parcel Name	Card

Fig. 2. Images of various functionalities of User application

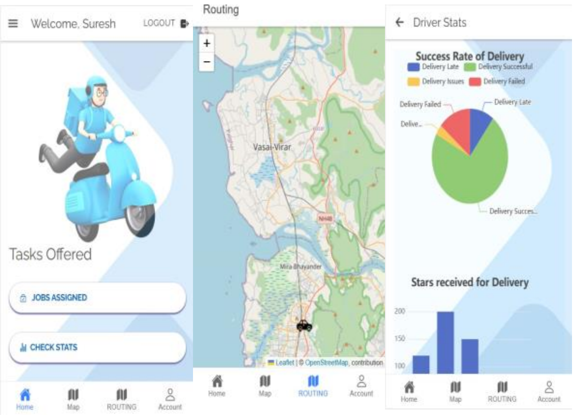


Fig. 3. Driver application user interface

VII. CONCLUSION

Cities are one of the most complex areas where the movement of freight and passengers takes place. By the virtue of the proposed solution, we intend on solving the urban mobility challenges that arise due to lack of technological advancement or proper utilization of available resources.

In the area of mobility, transportation of people and gods. is seen as a key component. Our system could be used by any users for their travel planning. The various constraints of users like time, budget, source and destination, traffic status is taken into consideration before recommending the optimal path.

Tools Technologies like JHipster, OptaPlanner are used keeping in mind low-code, no-code approach. The paper discusses several issues faced in the mobility sector, highlights the most important ideas, and attempts to solve each issue in the best possible way. The majority of the problems are resolved with workable solutions as mentioned in the paper. The aim of this project was to maximize societal impact which stands fulfilled.

VIII. FUTURE SCOPE

With regards to the future scope, this solution could be scaled at a larger level to suit the needs of states and countries at large. More and more modes of transportation could be considered and this solution could be modified and applied according to their constraints. If a scaled solution is being developed, appropriate care and attention could be focused towards improving the security and privacy of the user's data in the application. The user interface could be further improved to suit the user's requirements. We could focus on increasing the societal impact of the application by providing latest news related to the available mobility options in the location.

REFERENCES

[1] N. Sun, H. Shi, G. Han, B. Wang and L. Shu, ”Dynamic Path Planning Algorithms With Load Balancing Based on Data Prediction for Smart Transportation Systems,” in IEEE Access, vol. 8, pp. 15907-15922, 2020, doi: 10.1109/ACCESS.2020.2966995

[2] W. Li, X. Yan, X. Li and J. Yang, ”Estimate Passengers' Walking and Waiting Time in Metro Station Using Smart Card Data (SCD),” in IEEE Access, vol. 8, pp. 11074-11083, 2020, doi: 10.1109/ACCESS.2020.2965155

[3] Denis Ushakov, Egor Dudukalov, Ekaterina Mironenko, Khodor Shatila, “Big data analytics in smart cities' transportation infrastructure modernization”, Transportation Research Procedia, Volume 63, 2022, Pages 2385-2391, ISSN 2352-1465. https://doi.org/10.1016/j.trpro.2022.06.274.

[4] Husak, Vitalii, Lyubomyr Chyrun, Yurii Matseliukh, Aleksandr Gozhyj, Roman Nanivskyi and Mykhailo Luchko. “Intelligent Real-Time Vehicle Tracking Information System.” MoMLeT+DS (2021)

[5] R. Kannadasan, R. S. Kumar, K. S. Devineni, N. Prabakaran, A. S. Anakath and A. Jayanthiladevi, ”Vehicle Navigation Protocol in RealTime Fleet Management,” 2020 International Conference on Emerging Smart Computing and Informatics (ESCI), 2020, pp. 248-251, doi: 10.1109/ESCI48226.2020.9167608