

Geoinformation system for managing non-regular passenger transportation

*Andrii Yefimenko, Alex Kuzmenko, Halina Marchuck, Roman Petriv, and Inna Suhoniak**

Zhytomyr Polytechnic State University, Department of Computer Science, 103 Chudnovska Str., Zhytomyr, 10003, Ukraine

Abstract. Today there are a large number of different companies involved in non-regular passenger transport. Most of them use special software to reduce the speed of order processing and for better service management. Such programming tools often provide a convenient system for processing orders and a user interface for the contractors. Therefore, such companies have already forced out other competitors from the market of passenger transport. The purpose of the research is to develop and implement an algorithm of driver's path optimization and to develop a decentralized on-line service, oriented on support of the management of non-regular passenger transport, consisting of a service for car ordering, a service for processing and executing orders and a service for managing internal private organizations. Each service is an application accessible from both a web browser and a mobile device. In the paper the theoretical basis of the management of passenger transport has been analyzed and the basic problems have been defined. The route optimization task, which was reduced to the asymmetrical task of the salesman, was initiated. The Little algorithm for the resolution of this task was used.

1 Introduction

Modern management services of passenger transportation provide a convenient interface and ease of using. But they don't solve all user problems.

There are many different schemes for managing passenger transportation. Some companies have their own vehicle fleet, hired drivers and a database of customers. The other companies provide a centralized service with a database of drivers, customers and orders, and the client applications, used by drivers and passengers for their interactions.

There are a lot of advantages in such services. The driver himself decides the time and the place of his work. He may choose a favorable order for him. Passengers may also choose drivers and share impressions about the trips with them. But also, there are some problems. Trust and security are some of them. Even a rating system and reviews will not always help to solve these problems.

The problem of trust is suggested to resolve with the help of private groups, the drivers and the passengers will be joined in. These groups will be closed and participants will have an access to activate the private mode, thus any other driver or a passenger could not see their order in the system. So, the passenger transportation will only be done by trusted drivers. Drivers and passengers will also be able to combine a private and public mode to achieve the best efficiency of the work.

That is why the development of a service that solves these problems and provides a convenient user interface for passengers, drivers and, most importantly, private

groups (organizations) is a very actual problem of modernity.

Also, one of the most important problems in sustainable city development is road traffic. Optimal routing is perspective direction to solve this problem. Minimizing of path's length reduce fuel consumption and tail pipe emissions of carbon dioxide.

2 Problem analysis and the goal of research

Taxi is an important vehicle in urban transport systems. Encouraging taxi passengers can increase transportation efficiency, reduce overall distance, a time, a congestion and emissions. Today many researchers study the taxi problems. Particular attention is paid to the carpooling and its problems and optimization. The analysis of the most famous publications on the problem of taxi and carpooling allowed to determine the main directions of development of carpooling research and their peculiarities.

S. Ardekani represent a method for carpooling when the driver wages, vehicle operating costs and reasonable compensation of other participants depend on the travel time or the distance based on fuel consumption of the vehicle was proposed [1]. It was a first try to find optimal routers for carpooling problem.

In [2] a static optimal control model with a flexible starting price to satisfy market demand was constructed.

Zou proposed to use a transport network model to calculate the shortest path for a car and analyzed the

* Corresponding author: isugonyak@gmail.com

influence of road factors on the shortest path [3]. Next researcher developed a model for choosing a taxi route and methods of speed optimization based on the principle of fairness, which provides a minimum time of passenger trip and cost as a target function and guarantees a reasonable income of a driver, taking into account the interests of both drivers and passengers. The genetic algorithm was used to solve this problem [4].

M. Friedrich, M. Hartl, C. Magg represent the static and dynamic modes of working with a car, separating the carpooling mode into several types, such as one to one, one to many, and many to many, and developed the path choosing method and speed optimization model, was investigated [6].

In [5] an efficient model of route planning was proposed and a method for solving the problem associated with the taxi management problem was proposed.

M. Tamannaei & I. Irandoost presents a matching algorithm, which works with typical for agent-based technologies integer and non-integer demand occurring in travel macroscopic model. This approach compares two datasets of suppliers and demanders. Optimal path in the road calculated from sequence of links to a sequence of zones [7].

In [7] minimizes the costs of travel times. Branch-and-Bound (B&B) algorithm find optimal solution of the problem. The large-scale problems were solved by near-optimal algorithm. This model tested on real data from transportation network of Isfahan city, Iran.

In [8] authors present a new heuristic, formulated and tested on real-world, and simulated carpooling problem instances, that mimics a decentralized carpool self-organization process. These findings reveal system-wide savings similar to centralized models, and a potential strategy for improving carpool utilization.

The goal of the research was to develop an on-line service oriented for management of non-regular passenger transportation for different categories of users: drivers, passengers and organizations. Service must also resolve the problem of trust between drivers and passengers and implement an algorithm of driver's path optimization.

The defined purpose determines the following tasks:

- research and optimization of taxi tasks;
- construction a model of route optimization for a multi-route taxi;
- building a carpooling club model;
- optimization the driver's route;
- developing a decentralized model of system functioning and the mode of private passenger transportation;
- designing algorithms for the non-regular transportation management system;
- designing the structure of the non-regular transportation management system.

The main results of the on-line service implementation are car amount optimization and minimization path's length – to prevention traffic jam and protect environment in the big city.

3 Optimization for carpooling routes

3.1 Carpooling club model

There are a large number of people who have travel schedules for work enough stable, but there is also a significant and growing number of travels with multiple destinations and directions that tend to not relate work and non-work travels. People who have stable travels can easily create stable groups of cars for carpooling, and we must also take into account the fact that if a person has to go elsewhere than the destination, this must be included in the group route.

The concept of carpooling clubs is to manage both traditional and dynamic systems in one structure, using their compatibility, and also providing a minimum level of trust between their users through filtering and supporting all groups. The traditional flow of groups is the basis of the system, but when free space is appeared within a group, it can be dynamically distributed to random demand by people who normally belong to another flow in the club (Fig. 1). This system is based on users who can act as passengers, as well as drivers. No one is allowed to act solely as a driver or as a passenger; this helps to increase tolerance and avoid criticism.

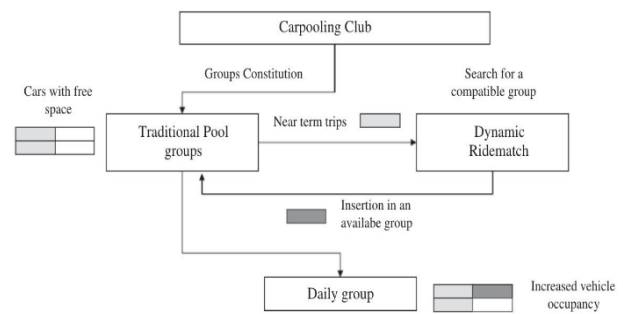


Fig. 1. Carpooling club model.

3.2 Taxi route optimization model

Model conditions:

- Taxi route is independent of the others.
- The capacity of each taxi is 3 passengers.
- Passengers are prohibited from changing the route.
- The taxi speed remains unchanged for carpooling and one-way travel.
- The taxi driver chooses the shortest route.

For a taxi route optimization model, we need to define a target function.

A detour can increase the travel time of passengers in carpooling mode, so a minimum taxi distance can be selected as a target function and optimize the taxi route. The objective function is defined by (1).

$$Z_1 = \sum_{k=1}^n \sum_{i=1}^A \sum_{j=1}^A y_{ij}^k q_{ij}^l d_{ij} \rightarrow \min \quad (1)$$

where $k \in n$, n is a set of taxis; d_{ij} – distance from i and to j ; i and j are adjacent points, $i, j \in A$, A is the number of nodes in the road network; $l \in p$, p is a set of paths for carpooling;

$$y_{ij}^l = \begin{cases} 1 & \text{if carpooling, } ij \in l \\ 0 & \text{if carpooling, } ij \notin l \end{cases}$$

$$q_k^l = \begin{cases} 1 & \text{if } l \text{ carpooling path of the vertex } k \\ 0 & \text{in other cases} \end{cases}$$

After defining the target function for the model, it is also necessary to determine the conditions of the delay restriction.

Carpooling can increase the length of travel for some passengers. For the whole system, total travel distances may not exceed distances when using a standard single-route taxi (2).

$$\sum_{k=1}^n \sum_{l=1}^p \sum_{i=1}^A \sum_{j=1}^A y_{ij}^l q_k^l d_{ij} \leq \sum_{k=1}^N \sum_{l=1}^p \sum_{i=1}^A \sum_{j=1}^A y_{ij}^l q_k^l d_{ij} \quad (2)$$

where $k \in N$, N is a single-way taxi set; $L \in P$, P is a set of single-way taxi routes;

$$y_{ij}^l = \begin{cases} 1 & \text{if taxi with single way route, } ij \in L \\ 1 & \text{if taxi with single way route, } ij \notin L \end{cases}$$

$$q_k^l = \begin{cases} 1 & \text{if } L \text{ taxi path with single way of the vertex } k \\ 0 & \text{in other cases} \end{cases}$$

The restriction of the driver income is also needed to be add to the model. Driver income is another condition that follows from this model. It always exceeds the cost of the travel with the longest distance between the passengers in the taxi (3).

$$r_{ij}^m \sum_m \sum_i \sum_j y_{ij}^l q_k^l d_{ij} \geq \max\{c_m q_m^k, \forall m\}, \forall k \quad (3)$$

where $m \in S$, S – the set of passengers.

In the modelling process above, it is assumed that all taxis are traveling with a constant speed. However, since urban transport increases, traffic jams occur, so taxi speed cannot be kept constant at all times. In this case, the target function can be defined by (4).

$$Z_2 = \sum_{k=1}^m \sum_{i=1}^A \sum_{j=1}^A y_{ij}^l q_k^l c_{ij} \rightarrow \min \quad (4)$$

where c_{ij} – the distance from i to j .

The problem of taxi carpooling belongs to the combinatorial optimization problems and to the class of NP-complete tasks, so classical algorithms cannot be used.

Since optimization methods are designed to find the optimal solution that satisfies more than one target function, the concept of optimal Pareto solution is introduced in the theory of optimization tasks to find a compromise solution in multicriteria models. The formal definition of the Pareto optimal solution of the problem is defined as follows

$$\min_{x \in X} (f_1(x), \dots, f_p(x))$$

A valid solution $\tilde{x} \in X$ is called Pareto optimal solution, if there is no other solution such that $f(x) \leq f(\tilde{x})$ for all $k=1, \dots, p$, and at least for one of

them this condition is strictly fulfilled $f_i(x) < f_i(\tilde{x})$, where p is the criteria of evaluation.

Since genetic algorithms proved themselves well as search techniques in large areas with little or no information on the target function properties and constraints, several methods and approaches for using genetic algorithms to solve multicriteria optimization have been developed.

Among representative decision algorithms are the Niched Pareto Genetic Algorithm (NRGA), the Non-dominated Sorting Genetic Algorithm (NSGA-II), and the Strength Pareto Evolutionary Algorithm (SPEA2). All these algorithms have favorable solution efficiency in a particular problem.

The multi-criteria genetic algorithm should obtain the maximum approximation of the set of solutions found to the true Pareto front, at the same time preserving and stimulating the variety of the solutions found, in order to completely obtain the entire Pareto front.

On the flowchart of the algorithm (Fig. 2), Pop represents the population, and Pop is the scale of the population expressed by A . Paretos is the optimal Pareto solution, where Paretos is the number of optimal Pareto solutions in the set of solutions, which is expressed by B . The final conditions have certain restrictions.

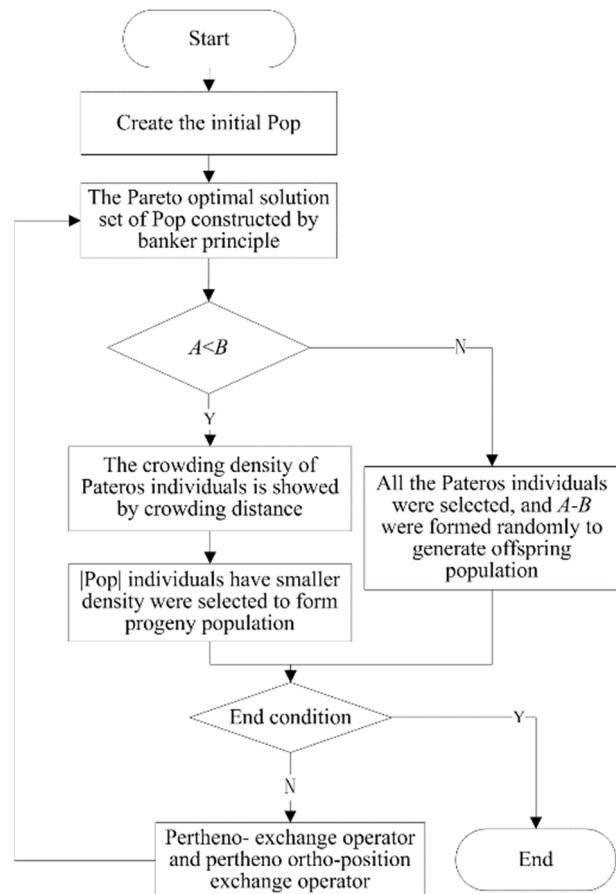


Fig. 2. Block diagram of the algorithm.

4 Design and implementation of separate modules of the system

4.1 Optimizing the driver path

In order to save fuel and a driver time, it is necessary to build the optimal route to be followed by the driver during the passenger transportation. Since an order may contain several destinations, it is possible to describe a model as a graph $G(V, E)$, where the points of destination and a driver location will be the set of vertices (V), and the shortest distances between all vertices will be the set of edges (E). The shortest distances between any two vertices can be obtained by using Google Maps API.

It must also be stated that the shortest distance from vertex A to vertex B will not always be equivalent to shortest distance from vertex B to vertex A (shortest_dist($A \rightarrow B$) \neq shortest_dist($B \rightarrow A$)), so the graph can have two edges from each vertex to each and should be oriented.

The solution of this problem can be the shortest closed route that passes through all the vertices of the graph exactly one time. In such formulation the problem of the optimal route for the driver is reduced to solving the asymmetric salesman problem. Since the problem is expressed by an oriented graph, it is asymmetric, so there $(n-1)!$ possible routes exist.

Since the problem of finding the optimal route where can be no large dimension is necessary to solve, it is suggested to use one of the exact algorithms, the Little's algorithm.

If we take the driver destination and location as the vertices of the oriented graph $G(V, E)$, and the shortest distances between them as the graph arcs, and impose an additional acyclic condition for the previously set task, the problem of the optimal driver route with no need to return to the starting point can be reduced to a closed variant of the salesman problem (see Fig. 3).

The unclosed variant of the salesman problem is reduced to the closed one by changing the weights of arcs that enter to the start vertex with 0. The optimal closed route of a salesman in such graph corresponds to the optimal closed route of the outgoing graph.

The minimum acyclic Hamilton route, which will begin at the starting point and pass through all destinations, specified by the client exactly once, will be the solution of this problem.

Following such a route, the driver will be able to fulfill the client order for minimal time without returning to the starting point. However, it is often necessary to visit one destination $A(0)$ first (start stop) and another destination $A(n-1)$ last (end stop) among the n certain destinations. In this case, the points between $A(0)$ and $A(n-1)$ should be visited in the most optimal order. Solving a previously set task will not satisfy the conditions of the new task.

As in the previous case, it is possible to formulate a new problem using the oriented graph $G(V, E)$, where the set of all client destinations is the set of vertices V , and the set of shortest distances between each location is the set of arcs E . In this case the location of the driver does

not need to be specified in the problem, since his first path will always be the path to the starting point, assigned by the client. After that he will need to follow the route, which will be the solution of the problem. And a minimal acyclic Hamilton route that starts strictly at a point A (client-assigned starting point) and ends at a point Z (client-assigned end point) will be the solution (Fig. 3).

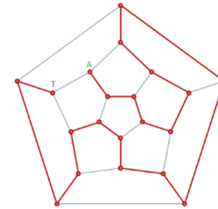


Fig. 3. The minimal acyclic Hamiltonian route, which starts at a given point and ends at the last one.

Since the route must be acyclic and begin and end at the certain vertices A and T correspondingly, the problem can be reduced to the closed salesman problem by adding a new vertex $V(n)$ into the graph. In this case, all arcs leading to vertex $V(n)$, except the arc from the vertex T , must be assigned a weight greater than any possible distance of any Hamiltonian route of the original graph. The arc from the vertex T must be assigned with the weight of 0. This is required to ensure that there is a guaranteed path from the vertex T to the vertex $V(n)$ in the result. Since, if in the cycle obtained during the of solution the vertex $V(n)$ with its edges is rejected, then the vertex T becomes the last in the optimal non-closed Hamiltonian route.

All the arcs coming out of the vertex $V(n)$, except the arc, that enters the vertex A , must have a weight greater than any possible distance of any Hamiltonian route of the original graph. The arc that enters the vertex A must be set its weight to zero. This is required in order to guarantee a path from the vertex $V(n)$ to the vertex A . As a result, if in the cycle obtained during the of solution, the vertex $V(n)$ with its edges is rejected, then the vertex A becomes the first in the optimal non-closed Hamiltonian route.

Based on the research, a problem was set, the solution of which completely satisfies the purpose of the work. This is a problem of minimal acyclic Hamiltonian path that begins at a given point and ends at the last one (see Fig. 3), which can be reduced to a non-closed asymmetric task. Since the solution of this problem is the optimal route, which begins at the first client-assigned point and ends at the point of the last destination in the client order, the given solution completely satisfies our purpose - to find an optimal route for the driver.

4.2 Designing algorithms for the non-regular transportation management system

As the geoformation management system for non-regular passenger transportation is oriented to two types of actors: drivers and passengers, the logic of processing orders, work with other actors and the map was separated on the corresponding modules. Among the main modules

in the system are the module for constructing and displaying the route of the user, the module for creating orders, the module for sending orders and managing them and the module for working with organizations.

All algorithms of the system were implemented in JavaScript language with using the additional libraries. The Google Maps API service was used for geocoding, reverse geocoding and for finding and constructing the shortest path between two specified addresses. The calculations of distance and cost are performed by a separate module based on a path built with the help of Google Maps API.

The algorithm of optimal driver path was obtained by the way of reducing the optimal path problem with fixed start and end points to the salesman problem. The algorithm for solving the salesman problem was implemented using the Little method. Using of Little's algorithm with Google Maps traffic jam detection technologies should reduce fuel consumption more than 30%.

4.3 Designing the structure of the non-regular transportation management system

An event-oriented approach for development was used using RAM cache based on Redis and using Sosket.io for sending notifications.

The non-regular passenger transportation management service (Fig. 4) physically is composed of a Web API service component and a client single page application that can be deployed with server-less architecture. The client part is not required a server to deploy. Amazon S3 or similar by the functionality platforms for the static content is sufficient to use.

To deploy a smartphone client application, corresponding package is required to install. Structural diagram of the system is shown in Fig. 5.

The business logic of the application is implemented in JavaScript programming language. A monk framework was used for work with MongoDB database. Each collection was built with a wrapper that provides a convenient interface for working with MongoDB.

Redis was used to cache orders, store Sosket.io customer IDs and other operations requiring quick access. To work with Redis on NodeJS, corresponding library node-redis was used, and an additional wrapper was implemented to adapt methods to the system.

To work with Web-sockets Socket.io library was used and a number of classes were written to handle events and send messages. The ConnectionHandler class is responsible for connecting and disconnecting users which open or close an application.

5 The results of the research

After the successful login, the user will see the main page. In driver mode, he will see a map with the locations of other drivers and a block displaying the current active orders from the clients (see Fig. 6).

If the driver mode is switched off, the user will see a page with a map, locations of the active drivers and a

block for building the path for his or her order.

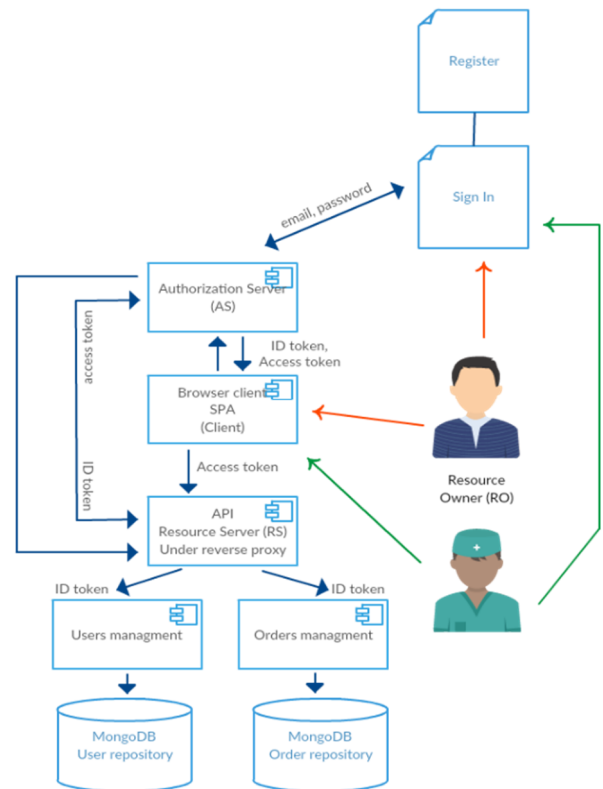


Fig. 4. System services interaction scheme.

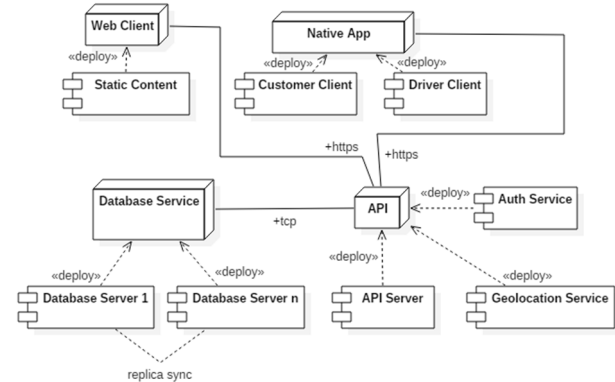


Fig. 5. Structural diagram of the system.

Passengers also have an option to optimize their route and build the shortest possible path. To do this the option “Shortest route” must be selected.

Once the order is created, it is sent to all drivers online, the drivers have 60 seconds to accept the order or decline it. After the first driver accepts the order, it becomes disable for all other drivers, and the passenger will be notified about the accepted order. The passenger must choose: to agree or to refuse to go with the driver who accepted the order. If the passenger agrees to go with the driver – the driver will receive a message about the confirmed order (Fig. 7) and has to go to passenger. Otherwise the driver will receive a rejection message, and all other drivers will again be able to accept the order.

After the order has been completed, the driver confirms its completion and optionally may write a

review of the passenger and evaluate him. The passengers have similar functions. Later the driver and the passenger may again create and accept new orders.

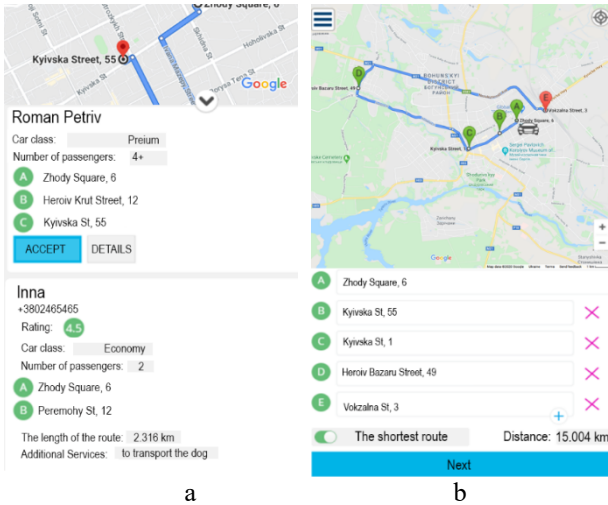


Fig. 6. Main page (a) for the driver and (b) for the passenger with the built path.

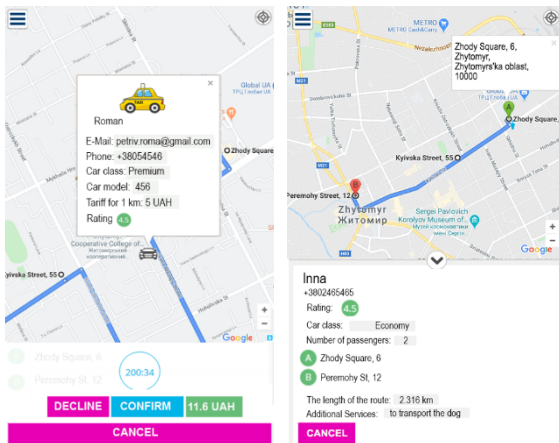


Fig. 7. The order received by the driver and the process of confirmation.

According to the results of the load testing, the service is able to support a large number of users while consuming a small amount of resources. As a result of the security testing, no critical vulnerabilities were identified, and measures were taken to protect against the most common attacks.

6 Conclusions

The passenger transportation management service was designed and implemented in the work.

To solve the set problem, a number of services for performing some useful for the work functions were analyzed.

The main functions of the service are building of paths on the map in real time, the modes of the driver and the passenger, the calculation of the length and the cost of the route, the creation of organizations and private modes of the work.

For implementation of fault-tolerant, highly accessible, reliable and scalable system, a number of technologies were analyzed and those that maximally correspond the specifics of the system functioning were selected. The document-oriented MongoDB was used as the database. To optimize data processing speed, data caching was used using Redis. The client and server code were written in JavaScript programming language. Working with Web-sockets is one of the most important functions of the system. Socket.io library was used for this purpose. The system is based on event-oriented architecture, which provides additional advantage in the resources use.

Algorithms for service functioning were developed and Google Maps API was used to implement some of the algorithms for building routes and geocoding.

This system should have positive impact to environment, because optimal transport routers minimize tail pipe emissions of carbon dioxide more than 5%.

References

1. S. Ardekani, B. Jamei, R. Herman. A taxicab fare policy formula based on fuel consumption observation. *Transportation Research Record* **11**(3), 33–39 (1989)
2. T. Chang, T. Chu. Optimal taxi market control operated with a flexible initial fare policy. *Networking, Sensing and Control* **2**(1), 1335–1440 (2004)
3. X. Zou, S. Zheng, X. Ban, X. Lian. Optimal path algorithm for road network with traffic capacity limits. *Journal of Highway and Transportation Research Development* **19**(4), 82–84 (2002)
4. L. Wang, Discussion on static and dynamic sharing mode of taxis in big cities, Master Dissertation, Changsha University of Science & Technology, 2012
5. X. Yun, Research on Traveling Salesman Problem Algorithm. *Advanced Materials Research* **694–697**, 2901-2904 (2013)
6. M. Friedrich, M. Hartl, C. Magg. A modeling approach for matching ridesharing trips within macroscopic travel demand models. *Transportation* **45**(6), 16–39 (2018)
7. M. Tamannaeei, I. Irandoost, Carpooling problem: A new mathematical model, branch-and-bound, and heuristic beam search algorithm. *Journal of Intelligent Transportation Systems* **23**(3), 203–215 (2019)
8. P. Kalczynski, M. Miklas-Kalczynska, A decentralized solution to the car pooling problem, *International Journal of Sustainable Transportation* **13**(2), 81–92 (2019)
9. D. Blystone, The Story of Uber: How a Snowy Night in Paris Created a Behemoth (Investopedia, 2018), <https://www.investopedia.com/articles/personal-finance/111015/story-uber.asp>. Accessed 11 Feb 2019

10. P.M. d'Orey, R. Fernandes, M. Ferreira, Reducing the environmental impact of taxi operation: the taxi-sharing use case, in in: *2012 12th International Conference on ITS Telecommunications*, pp. 319–323 (2013). doi:10.1109/ITST.2012.6425191
11. S.S. Semenov, A.V. Pedan, V.S. Volovikov, I.S. Klimov, Analysis of the Labor Intensity of Various Algorithmic Approaches for Solving the Traveling Salesman Problem. *Systems of Control, Communication and Security* 1, 116–131 (2017), <http://sccs.intelgr.com/archive/2017-01/08-Semenov.pdf>. Accessed 31 Mar 2020