A Survey Paper on Ride-Sharing Application

Shubham Sakhale ,Sahil Kadam ,Shreyas Joshi ,Akanksha Khedkar

UG Scholar,,Dept. of Computer Engineering, Smt Kashibai Navale College Of Engineering, Pune, India

---

Abstract - *The carpooling web application is a platform designed to connect drivers with available seats in their vehicles to passengers who are seeking rides in the same direction. The application aims to optimize transportation resources by facilitating ride-sharing, reducing traffic congestion, carbon emissions, and travel costs. The application functions by allowing users to create accounts and provide necessary personal details. Drivers can create trips by specifying their starting point, destination, date, time, and available seats. Passengers can search for trips based on their desired criteria and request to book seats. Drivers have the option to accept or decline booking requests based on their preferences and seat availability. Communication and coordination between drivers and passengers are facilitated through an in-app messaging system or by sharing contact information. After the trip, both drivers and passengers have the ability to rate and review each other, establishing a feedback system that enhances trust and reliability within the carpooling community. The carpooling web application may offer additional features such as advanced search filters, rating-based matching algorithms, trip update notifications, and integration with other transportation services.*

Key Words*: *Carpooling, optimize transportation, app messaging system, matching algorithm

1.INTRODUCTION

Carpooling is a transportation arrangement where several people travel together in the same vehicle, usually to a common destination or along a similar route. It is an environmentally friendly and costeffective solution that promotes resource efficiency and reduces traffic congestion on the roads. The concept of car sharing revolves around maximizing the use of the vehicle by giving people traveling in the same direction the opportunity to share a ride. This practice not only reduces the number of cars on the road but also reduces fuel consumption, carbon dioxide emissions, and the overall environmental impact of traffic. Carpooling offers several benefits to both individuals and communities. For commuters, it provides a convenient and affordable alternative to driving alone. By sharing the costs of fuel, tolls, and parking, carpooling helps save money and reduces the burden of transportation expenses. It also allows passengers to relax, work, or socialize during the commute, making the journey more productive and enjoyable. From a broader perspective, carpooling contributes to the reduction of traffic congestion, especially during peak hours, by reducing the number of vehicles on the road. This leads to shorter travel times, improved traffic flow, and less frustration for commuters. Additionally, carpooling helps alleviate parking space shortages in crowded areas, as fewer parking spots are needed when multiple passengers share a vehicle. With the advent of technology, carpooling has been revolutionized by web and mobile applications that connect drivers and passengers in a convenient and efficient manner. These platforms facilitate the matchmaking process, allowing users to find suitable carpooling partners based on their travel preferences, schedules, and routes. Such applications provide a user-friendly interface, secure payment systems, and features like real-time tracking, ratings, and reviews, enhancing the overall carpooling experience. Carpooling not only provides practical benefits but also fosters social connections and community building.

1.1 OBJECTIVES

The primary objectives stated are, Promoting alternative modes of transport. Reducing single occupancy trips by implementing- a Ride system and reducing the number of cars on road. Reducing pollution and carbon dioxide emission. State-of-the-art analysis of ride-sharing online platforms; Identification of factors affecting current and potential ride-sharing passenger and drivers.

1.2 MOTIVATION

The Motivation is to find people to share a ride with is the challenge of carpooling as it is difficult to find a person going to the same place as you at a given time.

The downside of many such applications available in the market is regarding the limitation of location and language barriers in specific regions.

The Application created must be generic, easy to comprehend and use in our day-to-day activities. It establishes an eco-friendly and a active environment for a sustainable ecosystem.

2. LITERATURE SURVEY

Mayur K. Thorat and Rahul M. Lahakare [1] have provided a summary of the Carpooling system with SMS warnings with an emphasis on how to fix existing problems and increase security. It was suggested that you may use it for both intercity and intracity travel. They made an effort to increase the number of users by including blind persons, who can utilize speech recognition technology to accurately determine their location at any moment.

R. Manzini and A. Pareschi [2] have provided a system to aid in decision-making for the use of carpooling. Passengers will use this to help them choose which vehicles to use. Swati. R. Tare, Neha B. Khalate, and Ajita A. Mahapadi [3] have contributed by offering suggestions for how to improve this application's usability for users other than drivers.

They focused particularly on the security of female travelers and the dependability of real-time systems. The greatest long-distance ridesharing network in the world is called BlaBlaCar. [4]. Frédéric Mazzella came up with the idea for BlaBlaCar in December 2003, and the company was established in 2006.

This program has a significant drawback in that it only provides intra-city carpooling choices, which is something that our service seeks to address. A community-based system called Volkswagen enables users to share rides with others. The driver receives a portion of the charge while the passengers pay far less than they would for a traditional taxi service. It is exclusively available to corporate clients as registration requires a company email and payment is made through a prepaid account or online wallet system

. The first car-sharing software to promote carpooling for "vacationers" was the well-known taxi-hire app "taxifares" [5] on the Android platform. i.e., for people who are on vacation and wish to save money by traveling less. They first only offered it on a few specific routes, such as Chandigarh-Delhi and MysoreManali, but they hope to eventually make it available

to everyone. Do not use abbreviations in the title or heads unless they are unavoidable.

"Optimizing the Efficiency of Carpooling Systems through Pickup and Delivery Point Re-arrangement" by Guo et al. (2019) - This paper proposes a model to optimize the efficiency of carpooling systems by rearranging pickup and delivery points. "Building a Scalable Web Application using Node.js and MongoDB" by Hage et al. (2016) - This paper presents a case study on building a scalable web application using Node.js and MongoDB and discusses the challenges and solutions encountered during the development process. "Evaluating the Performance of Node.js for Real-time Web Applications" by Neuman and Mugambi (2018) This paper evaluates the performance of Node.js for real-time web applications and provides insights into the scalability and reliability of the technology.

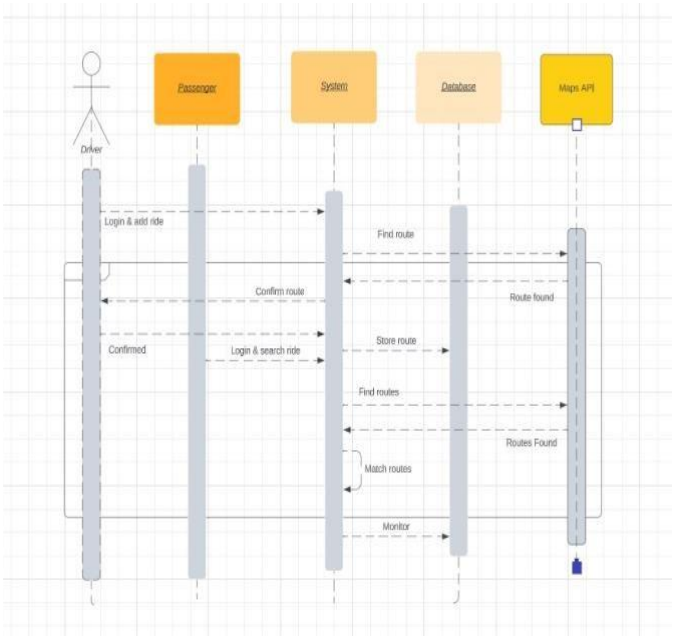

Fig – 2.01 Sequence Diagram

3. METHODOLOGY

The methodology for developing a carpooling web application using Node.js, React.js, and MongoDB includes the following steps:

Wireframes, user stories, and flowcharts to guide the development process.

- Requirements gathering : The first step is to gather requirements for the application, which can include user requirements, business requirements.
- Development: The third step is to develop the application using Node.js, React.js, and MongoDB. This step involves coding the frontend and backend of the application, integrating third-party APIs, and creating test cases to ensure the application's functionality and performance.
- Testing: The fourth step is to test the application to make sure it meets the requirements and is bug-free. This phase includes unit testing, integration testing, and system testing to validate the functionality and performance of the application.
- Deployment: The fifth step is to deploy the application to a production environment. This step involves configuring the application's servers, databases, and other resources to ensure that the application is available to users.

- Maintenance and support: The final step is to maintain and support the application post deployment. This step involves monitoring the application's performance, fixing bugs, and adding new features to meet evolving user requirements.

- Market trends, and identifying the scope of the project. Design and architecture: The second step is to design the application's architecture and create a high-level design that includes the user interface, data models, and backend services.

4. CODING & IMPLEMENTATION

Develop the application using Node.js, React.js, and MongoDB. This step involves coding the frontend and backend of the application, integrating third-party APIs, and creating test cases to ensure the application's functionality and performance.

1. Design the front-end
2. Build the front-end
3. Build the persistence layer (back-end database and data models)
4. Build the API (back-end application)

4.1 Frontend

A Real-time dynamic result design generalities as was noted in section II, real-time dynamic carpooling and lift-participating results are getting more common among the current carpooling and lift- participating results, although it takes further designing trouble to achieve real-time dynamic capabilities than for bare stationary carpooling and lift- sharing. The reason for the recent increase is obviously because real-time dynamic results are more accessible, and therefore more likely to be used in lesser figures by end druggies, but also because some technologies preliminarily used for putatively real-time communication on the web, have only lately progressed and have been formalized. In the soliciting of the so-called Web2.0, at the time when real-time streamlining websites were only just starting to appear, utmost of those websites used Asynchronous JavaScript and XML(AJAX) (10), which is a group of interrelated webreal-time web operations. Utmost of those ways reckoned upon regular HTTP, a simple request response and stateless protocol. Having to achieve what was generally two-way communication took some trouble for websites and web operations, using colorful workarounds, ways involving the use of the cybersurfed Xml Http. Request object or some other web cybersurfed plugins. The first workarounds developed into ways known as frequent polling, long-polling and the so called ever frames. Although all of those ways were, and still are, veritably much usable for putatively real- time web runner updates without taking full runner refreshes, they had downsides.

Their primary debit was, notwithstanding customer-side perpetration difficulties, the quantum of garçon-side and network coffers they consume. The garçon is more over forced to respond to a large number of frequent requests, or it opens up a number of long handling responses, which also enthrall its tackle coffers. On the other hand, using workarounds similar as colorful cybersurfed plugins, although lower network and garçon- side resource demanding, turned out to be non-practical, because of the lack of plugin support on the current mobile bias. For similar reasons, new ways were developed and lately formalized by the W3C. As part of the HTML5 specification Garçon- transferred DOM Events (SSE) were formalized in 2011 (11), but haven't yet been enforced by all desktop cybersurfs, videlicet, and Internet Discoverer. still, Web Sockets API (12), drafted a protocol back in 2009 presently supported by all major web cybersurfs. Web Sockets give a fullduplex communication channel over a single TCP connection, therefore allowing for a lower network quiescence time due to lower business outflow compared to HTTP. Compared to SSE and other polling ways Web Sockets give the stylish option for erecting real- time communication on the web, and that's why such a protocol is part of our proposed design conception. So, to achieve ubiquity, we propose the use of combined HTML5/ CSS3 for the stoner interface (UI) picture. erecting the UI around streamed realtime data flows of state changes created by passenger and vehicle driving stoner events (druggies requesting lifts, stoner driving busy, stoner driving free, etc.) is why the paradigm of reactive programming seems to be the perfect choice. Reactive programming isn't to be confused with responsive web design, which is also

4.2 Backend

As this application's frontend was made using REACTJS which is a JavaScript Framework so NodeJS would be an exceptional option for the backend code Nodejs is a server-side platform based on the JavaScript Engine in Google Chrome. It was created by Ryan Dahl in 2009, and the most recent version is v0.10.36. This is a cross-platform runtime environment for developing server-side and networking applications that are open source. Node.js programs are written in JavaScript and run on the Node.js runtime on OS X, Microsoft Windows, and Linux. Node.js also comes with a big library of JavaScript modules, which makes developing Node.js web apps much easier. The backend design of a carpooling web application using Node.js, React.js and MongoDB will typically follow a Model-ViewController (MVC) architecture, which separates the application into three main components:

1. **Models:** Models represent the data and define the application's data schema. In the case of a carpooling web application, the models could include user profiles, ride

requests, ride offers, and booking details. 2. **Views:** Views represent the frontend user interface, which displays the data to the user and enables them to interact with the application. In the case of a carpooling web application, the views could include the landing page, the ride search page, the ride offer page, and the user profile page.

3. **Controllers:** Controllers act as an intermediary between the models and views and handle user input and application logic. In the case of a carpooling web application, the controllers could include the user authentication controller, the ride search controller, the ride offer controller, and the booking controller.

4. **Data models:** The data models could be defined using a MongoDB schema and include models for users, ride requests, ride offers, and booking details. 5. **Express.js server:** The backend server could be built using Express.js, which is a popular Node.js web framework. The server could define routes for handling user authentication, ride search, ride offer, and booking requests.

6. **Mongoose ORM:** Mongoose is an ObjectRelational Mapping (ORM) library for MongoDB that simplifies the data modelling process. The application could use Mongoose to interact with the database and perform CRUD (Create, Read, Update, Delete) operations on the data models.

7. **Authentication middleware:** The application could use Passport.js, a popular authentication middleware for Node.js, to handle user authentication and authorization.

8. **API endpoints:** The application's API endpoints could be defined using RESTful API principles and return data in JSON format. For example, the ride search endpoint could return a list of available rides that match the user's search criteria.

The Database Class diagram for this application would be:

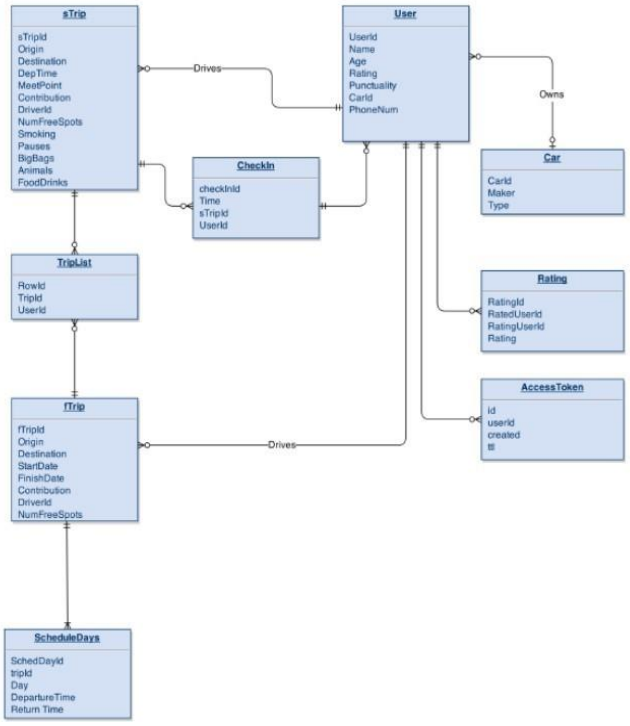

Figure:- 4.01

4.3 Database

Node.js (Javascript) and NoSQL format data the same way using JSON, which improves performance because there is no need to reformat data. Reformatting will be the case if we use an SQL database since the result need to be processed to meet the JSON format. As stated before, an application that is expected to be scalable should use a NoSQL database. NoSQL provides a flexible way to store data where the rules are stated by the programmer and not by a relational model. Also, the choice of Node.js as an application server pushes the choice of a NoSQL database as it is more supported than a traditional SQL one.

4.4 User interface

1.Login/Register:

If the user already has a account, they can login or else opt for Registration to create account

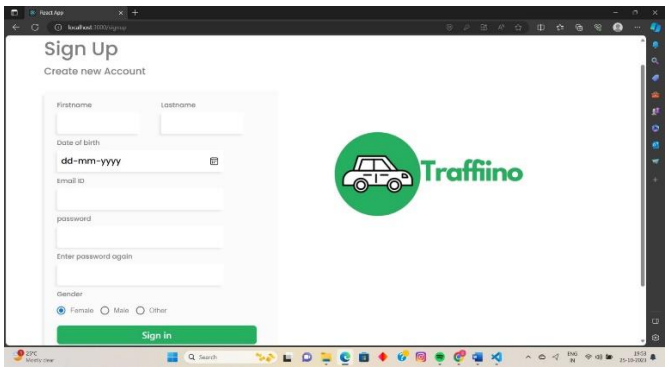

Figure:- 4.02

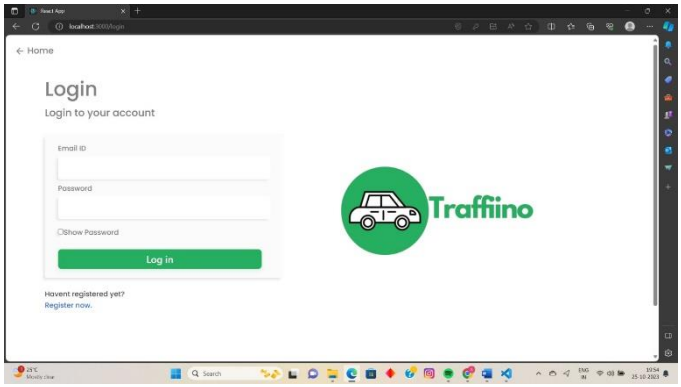

Figure:- 4.03

2.Post ride:

As a rule, platforms offer rental opportunities in various cities and countries. Choose a location that is convenient for you and decide on the time.

After specifying the details of your planned trip, you can move on to the most enjoyable part - selecting a car. Approach this matter unhurriedly: examine detailed characteristics of the vehicles, including the model, price, and rental conditions. Also, do not ignore feedback from other users about their driving experience to get a complete picture of your future trip.

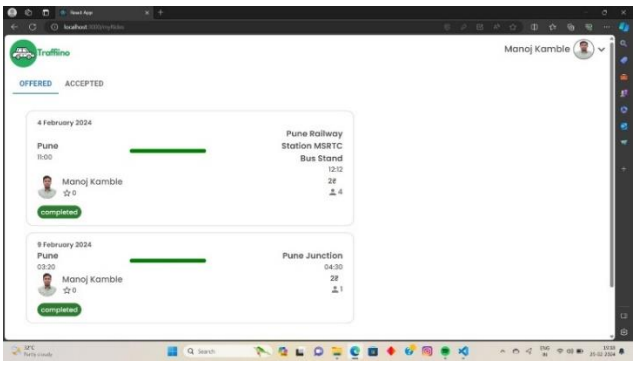

Figure:- 4.04

3.Ride status:

 Ride status This will show us the current status of the ride which is accepted. Once completed the user will click complete to officially end the trip.

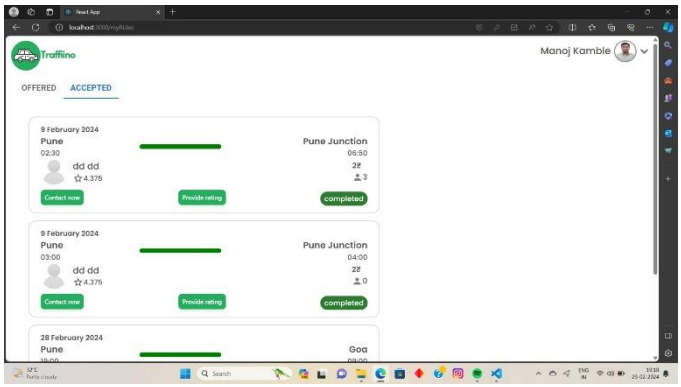

Figure:- 4.05

4.Chat page:

a chatpage might be a component within an website that facilitates communication between users.

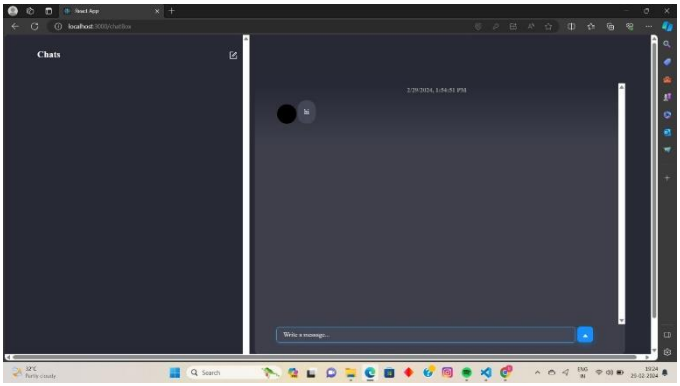

Figure:- 4.06

5.MERITS AND DEMERITS

Advantages:

1. **Cost savings:** Sharing allows users to share transportation costs, which reduces their overall costs. By providing a platform for users to search for and offer rides, a carpooling app can help users save money on transportation costs.

2. **Reduced traffic congestion:** Sharing can help reduce traffic congestion by reducing the number of cars on the road. This can shorten commutes and reduce air pollution. o Environmental benefits, Sharing can reduce the environmental impact of traffic by reducing the number of cars on the road and reducing greenhouse gas emissions.

3. **Convenience:** The carpooling website can provide users with a convenient way to find and offer a ride, facilitating transportation arrangements.

4. **Increased social interaction:** Carpooling can offer users the opportunity to meet new people and interact with their fellow passengers.

5. **Real-time tracking:** The carpooling website can provide real-time tracking of ride requests, offers, and reservations, allowing users to track their rides and make changes as needed.

Disadvantages:

1. **Different Time Requirements:** People on the bus may run errands at different times, or they may need to take a different route to reach their destination. This becomes a problem when sharing.

2. **Less Privacy:** Some people like privacy when they travel. These individuals must compromise their privacy.

3. **Maintenance and support:** The program requires ongoing maintenance and support to keep it updated and updated. This can lead to a waste of resources and time.

6. Future Scope

Future carpooling apps may incorporate more sophisticated matching algorithms that consider factors beyond just location and timing, such as preferences for conversation, music, or driving style. Machine learning techniques could help improve the accuracy of matches and enhance user satisfaction.

Integration with smart city infrastructure and transportation systems could enable real-time data sharing to optimize routes, reduce congestion, and provide incentives for carpooling. This might involve partnerships with municipalities or transportation authorities. As autonomous vehicle technology advances, carpooling apps could integrate with self-driving cars to offer on-demand ridesharing services. This could lead to more efficient use of vehicles, reduced traffic, and increased accessibility to transportation for those unable to drive.

Future carpooling apps may provide users with insights into the environmental and social impact of their ridesharing activities, such as CO2 emissions saved, kilometers reduced, and community contributions. This could help promote sustainability and encourage more people to participate in carpooling.

Carpooling apps may evolve to integrate with other modes of transportation, such as public transit, biking, or ride-hailing services, to offer seamless multi-modal journeys. This could provide users with more flexibility and convenience in planning their trips.

Future carpooling apps may prioritize safety and trust-building measures, such as enhanced background checks, identity verification, real-time tracking, and emergency assistance features. This could help alleviate concerns about traveling with strangers and foster a greater sense of security among users.

7. CONCLUSIONS

In conclusion, the carpooling web application offers a convenient and sustainable solution for users to share rides and reduce their carbon footprint. By providing a platform for passengers to find and connect with drivers who are traveling in the same direction, the app promotes efficient resource utilization and fosters a sense of community.

Throughout the development process, various technologies and frameworks were utilized to create a robust and user-friendly application. The frontend, built with HTML5, CSS, and JavaScript libraries like React, ensures an intuitive and responsive user interface. The backend, powered by Node.js with Express.js, manages the server-side logic and data storage, facilitating seamless communication between users and the application.

In conclusion, the carpooling web application serves as a comprehensive solution that addresses the challenges of commuting and transportation. It offers a user-friendly platform for individuals to connect, share rides, and contribute to a more sustainable and efficient way of commuting.

8.ACKNOWLEDGEMENT

We want to specially thank our respected internal guide Prof. Barkha Kasab for his guidance and encouragement which has helped us to achieve our goal. His valuable advice helped us to complete our project successfully. Our Head of Department Prof. R.H. Borhade has also been very helpful and we appreciate the support he provided us. He also gave us valuable inputs during our work.

We would like to convey our gratitude to Dr. K.R. Borole (Vice Principal) and Dr. A.V. Deshpande (Principal) all the teaching and non-teaching staff members of the Computer Engineering Department who gave us the freedom to explore and guided us the right way, also our friends and families for their valuable suggestions and support.

9.REFERENCES

1. Riccardo Manzini, Arrigo Pareschi Department of Industrial Engineering, Bologna University, Bologna, Italy. A Decision-Support System for the CarPooling Problem.

2. S Abutaleb, N El-Bassiouny, S Hamed Management of Environmental Quality: An International Journal, 2020 . Sharing rides and strides toward sustainability: an investigation of carpooling in an emerging market

3. DW Massaro, B Chaney, S Bigler, J Lancaster, S Iyer, M Gawade, M Eccleston, E Gurrola. Just-in-Time carpooling without elaborate preplanning

4. S Carrese, T Giacchetti, SM Patella, M Petrelli 2017 5th IEEE International Conference on Models and Technologies. Real time ridesharing: Understanding user behavior and policies impact: Carpooling service case study in Lazio Region, Italy

5. J Ferreira, P Trigo, P Filipe - ... Journal of Humanities and Social Sciences, 2009 . Collaborative car pooling system

6. MS Do, HY Jung .Journal of Open Innovation: Technology, Market, and Complexity, 201 . The socioeconomic benefits of sharing economy: Colleaguebased carpooling service in Korea and used for open innovative techologies

7. BP Bruck, V Incerti, M Iori, M Vignoli - Computers & Operations Research, 2017 . Minimizing CO2 emissions in a practical daily carpooling problem

8. N Bicocchi, M Mamei - Pervasive and Mobile Computing, 2014 . Investigating ride sharing opportunities through mobility data analysis

9. P Julagasigorn, R Banomyong, DB Grant... - Transportation Research ..., 2021 . What encourages people to carpool? A conceptual framework of carpooling psychological factors and research propositions

10. YT Chen, CH Hsu International Journal of Social Science and Humanity, 2013 . Improve the carpooling applications with using a social community based travel cost reduction mechanism