International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Development of Carpooling Website Using Machine Learning

Meduri VNS SRK Sai Somayajulu[1], G. Karthik[2], R. SaiKiran[3], K. Chandu Venkata Satya Sai Ganesh[4], N. Jaswanth[5]

[1]Assistant professor, Dept of CSE, Krishna University College Of Engineering And Technology, Machlipatnam, Krishna (Dt), AP, India.
[2,3,4,5]Final year Student, Dept of CSE, Krishna University College Of Engineering And Technology, Machlipatnam, Krishna (Dt), AP, India.

ABSTRACT:
In the face of urban congestion and environmental concerns, Carpooling has emerged as a sustainable solution for optimizing transportation resources. This abstract presents the architecture and functionality of a Carpooling Website designed to facilitate efficient ride- sharing among users. The proposed web Website offers a user-friendly interface allowing individuals to register, create profiles, and list their travel preferences including departure times, routes, and passenger capacity. Through a sophisticated matching algorithm, the system identifies compatible travel plans and suggests potential matches to users, considering factors such as proximity, departure times, and user-defined preferences. Key features of the Website include real-time tracking, allowing users to monitor the location of their fellow travelers and receive notifications on any changes or updates to their shared rides. Moreover, the system incorporates a secure payment gateway for handling transactional processes, ensuring fair cost-sharing among participants. To enhance user engagement and trust, the Website includes a rating and review system where users can provide feedback based on their Carpooling experiences, thereby fostering a community-driven culture of reliability and accountability.

Keywords: Carpooling, Ridesharing, Price Recommendation.

1. INTRODUCTION

In today's world, where urbanization and population growth continue to strain transportation infrastructure and contribute to environmental degradation, the need for sustainable and efficient commuting solutions is more critical than ever. Car pooling, a practice where multiple individuals share a Single vehicle for a common journey, presents promising avenue for alleviating traffic congestion, reducing carbon emissions, and cutting transportation costs. To leverage the v benefits of Carpooling and address the challenges associated with traditional commuting methods, we propose the development of a sophisticated Carpooling website. This website will serve as a centralized platform, connecting commuters with similar routes and travel preferences, facilitating ridesharing arrangements, and promoting community engagement. With Python as the backend programming language, HTML, CSS, and JavaScript for frontend development and MongoDB as the database management system, our framework of choice will be Flask, a lightweight and flexible web Website framework. In this project,

we aim to combine the power of Python's backend capabilities with the versatility of HTML, CSS, and JavaScript for creating an intuitive and interactive user interface. Flask, known for its simplicity and ease of use, will enable us to build a robust backend infrastructure, handle HTTP requests, manage user sessions, and interact with the MongodB databases ne aimlessly. Through this Carpooling website, users will be able to register, create profiles, post and search for rides, book and confirm ride requests, make secure payments, provide ratings and reviews, and engage with fellow commuters in real-time communication. The platform will prioritize safety, convenience, and efficiency, offering advanced features such as secure payment systems, user verification, real-time messaging, and advanced matching algorithms. By harnessing the power of technology and community collaboration, our Carpooling website aspires to revolutionize the way people commute, promote sustainable transportation practices, and contribute to greener and more connected future. With its user-friendly interface, comprehensive features, and commitment to sustainability, this platform seeks to empower commuters to make environmentally conscious choices while enhancing their overall commuting experiences. Assistance content based discussing of memory pages. Specifically, if several VMs citizen on the same physical server use similar Web pages, content-based discussing allows storing just one duplicate of the distributed webpage. Thus, material based sharing reduces the storage impact needed to host a set of VMs on only one actual server. The concept of content-based discussing was first used in the Disco program, and consequently website lied in VMware ESX where the strategy was proven to preserve as much as 33% of the storage sources of a server.

2. MATHEMATICAL MODEL

Entities:

Users: Represents individuals who are either offering or requesting rides.

Rides: Represents the transportation service from one location to another.

Locations: Represents the pickup and drop-off points.

Costs: Represents the expenses involved in the ride- sharing process.

Attributes:

User: user ID, Name, Email, Password, Location, Preferences (e.g., smoking, pets), Rating, etc.

Ride: Ride ID, driver ID, Pickup Location, Drop off Location, Available Seats, Date, Time, Cost, etc.

Operations:

User Registration/Login: Users can register with their details or log in using existing credentials.

Offer Ride: Drivers can offer rides by specifying pickup/drop-off locations, date, time, available seats, and cost.

Request Ride: Passengers can request rides by specifying pickup/drop-offlocations,date,time,andany preferences.

Matchmaking: The system matches drivers with passengers based on their preferences, locations, and timings.

Payment: If website likable, users can make payments for the ride.

Rating: After the ride, users can rate each other to build trust within the community.

Mathematical Formulation:

Distance Calculation: Utilize mathematical formulas to calculate distances between pickup and drop-off locations.

Cost Calculation: Based on distance, time, and any other factors (e.g tolls, additional stops),calculate

the cost for the ride.

Matching Algorithm: Develop an algorithm to efficiently match drivers with passengers based on their preferences and locations, considering factors such as distance, time, and cost.

Payment Calculation: If there's a payment system, calculate the payment amount based on the agreed-upon cost and any additional charges.

3. SYSTEM ARCHITECTURE

The system architecture for a Carpooling website typically involves multiple components working together to provide the desired functionality. Here's a high-level overview of the system architecture below are the screenshots and explanations for our Carpooling Website

a. Login Page

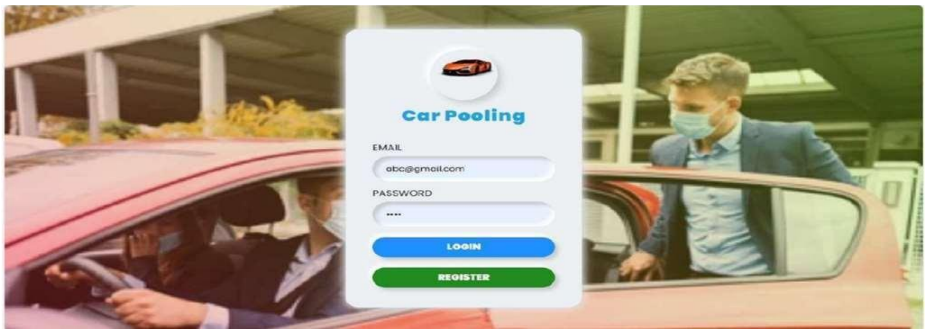

Figure 1. Login Page

This is the login page of the “Car pooling” website, where the user will enter their phone number as user id and password to login into their account. The GUI of this page is simple, two input fields for phone number and password, one login button to login and finally a link at the bottom to register as new users for creating a new account.

Figure 2. Registration page

Registration Page For new users who don't have login credentials, the user is redirected to the registration page. For registration the user requires to enter a name, unique email id, unique phone number and password. Users have to select gender by radio buttons as it acts as the main feature of the website where further listings are filtered using gender. For travelling from source to destination, the user has two options either to be a rider or a driver. Here driver means the person who will be taking a passenger while rider means the person who will ask for the lift. After successful log in users can opt between rider and driver.

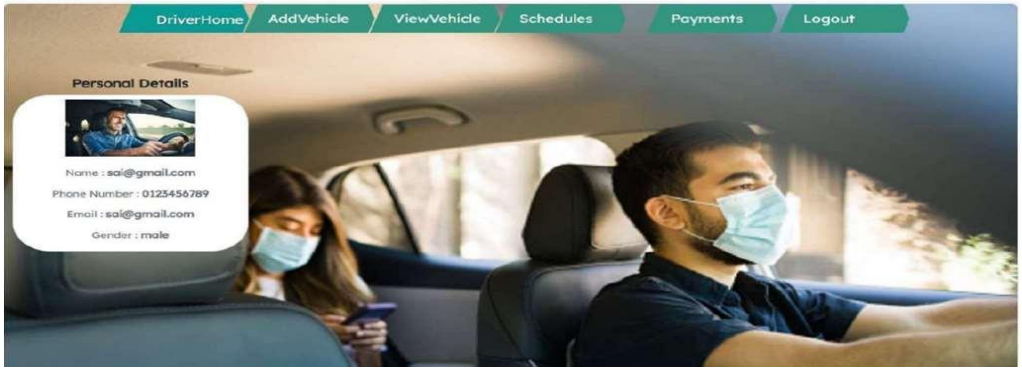

Figure 3. Driver Home Page

Drive Home Page contains Driver Home, Add Vehicle, View Vehicle, Schedules, Payments, Logout.

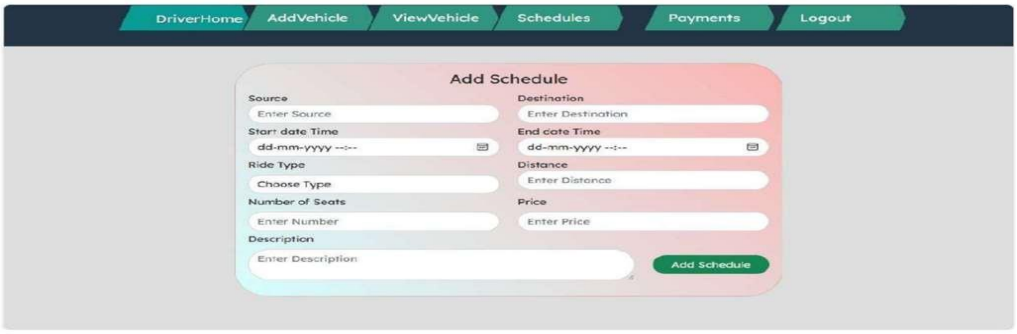

Figure 4. Add Schedules

This Add Schedule Page appears to show a user interface for scheduling a ride or trip using a vehicle.The user is presented with a series of prompts to enter information for the trip, including the source, start date and time, ride type, number of seats needed, and a description. The user can also enter the destination, end date and time, distance, and price.

Figure 5. Rider Home Page

This Figure appears to show a user interface for a ride- sharing or transportation application or website, with a menu of options at the bottom and personal details at the top. The menu of options at the top includes the following options Rider Home, Schedules, Bookings, Payments, Logout.

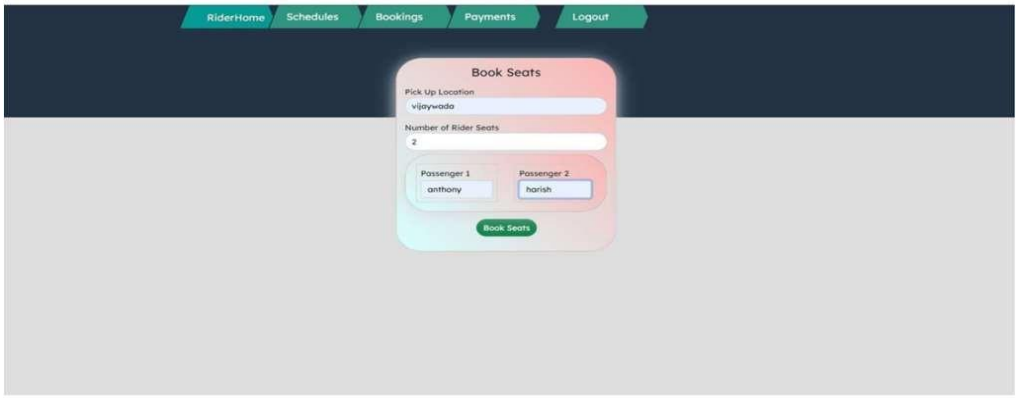

Figure 6. Bookings

Driver and rider. Here after successful login the user selects either to be a rider or a driver. If driver is selected then the driver will post the ride while if rider is selected then the rider will search the ride, get information and finally book the ride. Now if everything matches they will share the ride.

4. SOFTWAREMODELLING

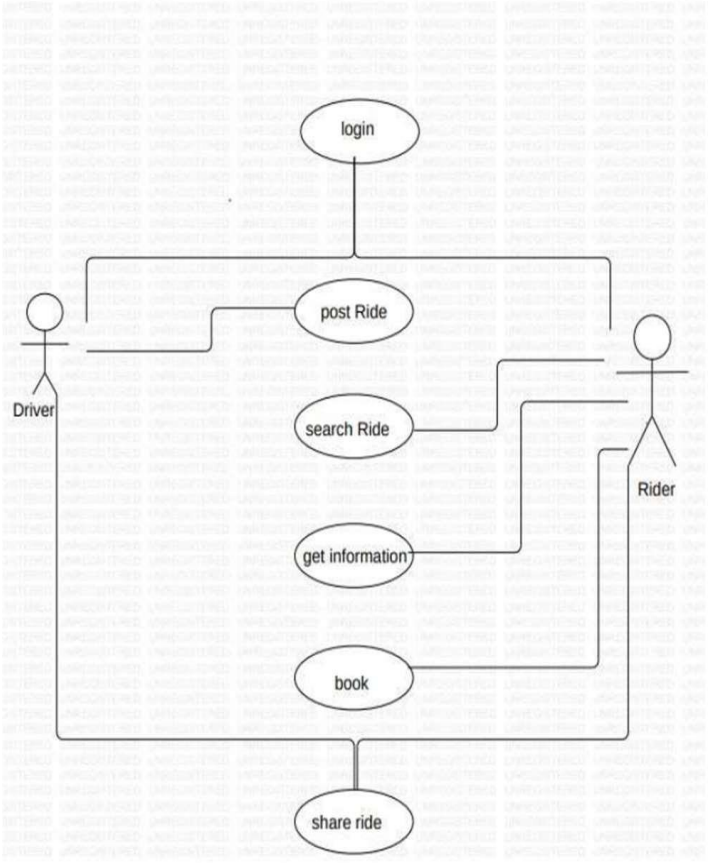

Figure7. USE CASE DIAGRAM

The functionality of a system can be described in a number of different use-cases, each of which represents a specific flow of events in a system. It is a graph of actors, a set of use cases enclosed in a boundary, communication, associations between the actors and the use cases, and generalization among the use cases.

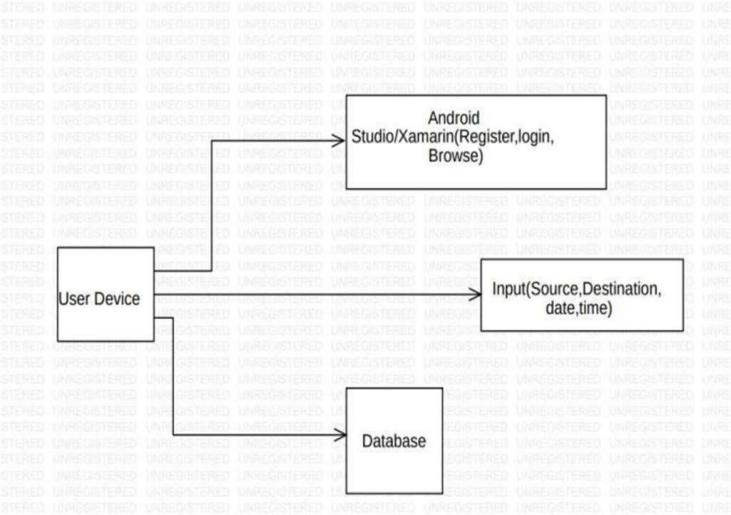

Figure 8. Component diagram

A component diagram displays the structural relationship of components of a software system. These are mostly used when working with complex systems that has many components. Components communicate with each other using interfaces. The interfaces are linked using connectors. Below images shows a component diagram

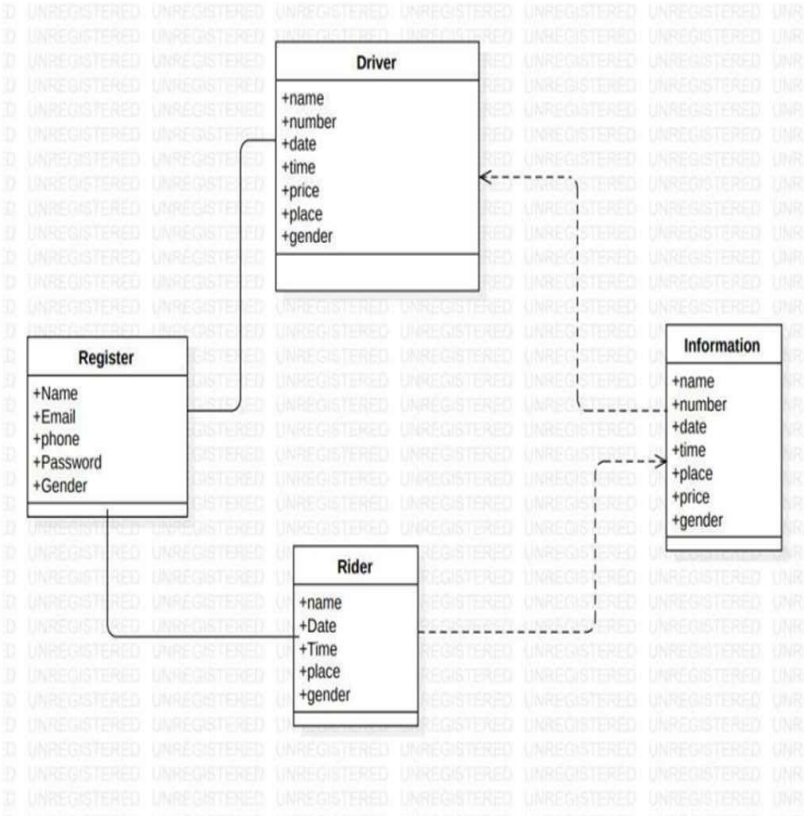

Figure 9. Class Diagram

The purpose of a class diagram is to depict the classes within a model. In an object oriented application, classes have attributes (member variables), operations (member functions) and relationships with other classes. The UML class diagram can depict all these things quite easily. The fundamental element of the class diagram is an icon the represents a class. This icon is shown in the figure above.

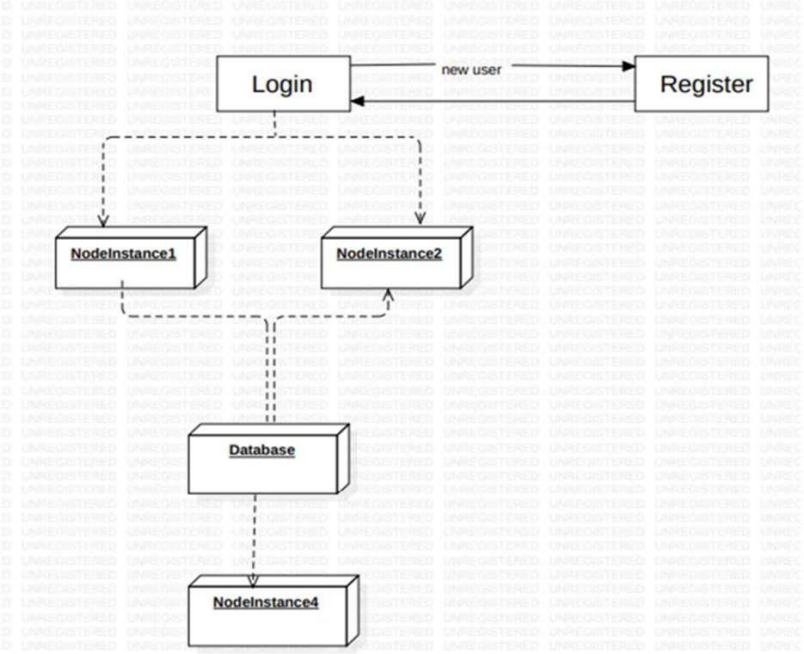

Figure 10. Deployment Diagram

A deployment diagrams shows the hard ware of your system and the software in those hardware. Deployment diagrams are useful when your software solution is deployed across multiple machines with each having a unique configuration. Below is an example deployment diagram. UML Class Diagram with Relationships.

5. SOFTWARETESTING

Software program testing is the procedure of comparing a software program object to come across variations among given input and predicted output. Also to evaluate the characteristics of a software program item. Software testing is a method that ought to be accomplished at some stage in the development process. In different phrases software program testing is a verification and validation method.

Verification: Verification is the process to make certain the products at is files the condition imposed on the start of the development phase. In different phrases, to make sure the product behaves the manner we want it to. Validation: Validation is the technique to make certain the product satisfies the required requirements on the give up of the development phase. In different phrases, to make sure the product is built as consistent with client necessities

Basics of software testing: There are two basics of software testing: black-box testing and white-box testing.

- Black-box Testing: Black box testing only focuses on the output generated and ignores the internal structure of the system.
- White-box Testing: White Box testing or the structural testing focuses on the internal structure of the system and is always used for validation purposes.

Test Case ID 1 Test Case Description Run code on virtualmachineSteps1.DebugandCompilethecode.

2. Run the code. Test Case Result Website should be successful installed. Action Result Website successfully installed and started. Status Pass

Test Case ID 2 Test Case Description Website home page should get displayed on start-up with option

of login and registration. Steps 1. Open Website 2. Register 3. Login Test Case Result Website should be successfully started and Register Form should be accessible. Action Result Website accepted registration data and able to login successfully. Status Pass

Test CaseID3 Test Case Description Selection between driver and rider Steps 1. Login 2. Select driver or rider Test Case Result Display the respective page for driver and rider. Action Result both pages are displayed after each selection test. Status Pass

Test Case ID5 Test Case Description Processing of data for rider page Steps 1. Select as rider 2. Enter the required data like time and place. 3. Submit the data Test Case Result Data is compared with the available rider detain database. Action Result Data compared and match was found for the ride. Status Pass

6. CONCLUSION

This Flask application serves as a robust platform for facilitating carpooling services, catering got both drivers and riders. Upon registration and authentication, users are directed to personalized dashboards tailored to their roles. For drivers, the system offers seamless vehicle management capabilities, enabling them to add, update, or remove vehicles from their profiles, thereby provide essential details that riders might consider when choosing a ride. The scheduling feature empowers drivers to create and manage ride schedules, allowing them to specify crucial parameters such as departure times, destinations, and available seat capacities, providing riders with a comprehensive overview of upcoming rides. Riders, in turn, benefit from a user-friendly interface where they can explore available rides, view relevant details, and book seats based on their preferences and travel needs. The booking management system ensures efficient seat allocation, with drivers having the flexibility to accept or reject bookings depending on seat availability, thereby optimizing the utilization of vehicle capacities and enhancing the overall user experience. Additionally, the integration of secure payment processing functionalities facilitates seamless financial transactions, instilling confidence an reliability in the carpooling service.

REFERENCES

1. Liu, K., Zhang, J., & Yang, Q.(2019). Bus Car Pooling: A Large-Scale Bus Ridesharing Service. IEEE Access, 7, 74248–74262. doi:10.1109/access.2019.2920756
2. Hasan, R., Bhatti, A. H., Hayat, M. S., Gebreyohannes, H. M., Ali, S. I., & Syed, A. J. (2016). Smart peer car Car Pooling system. 2016 3rdMECInternationalConferenceonBigDataand Smart City (ICBDSC). doi:10.1109/icbdsc.2016.7460384
3. Dejan Dimitrijevic, Vladimir Dimitrieski, N. Nedic (2013), Real-time Carpooling and ride- sharing: Position paper on design concepts, distribution and cloud computing strategies Computer Science 2013 Federated Conference on Computer Science and Information Systems.
4. Binu, P. K., & Viswaraj, V. S. (2016). Android based Website for efficient Carpooling with user tracking facility. 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC). doi:10.1109/iccic.2016.7919536
5. Craig Standing,SusanStanding&SharonBiermann (2018): The implications of the sharing economy for transport, Transport Reviews, DOI: 10.1080/01441647.2018.1450307
6. Gedam,C.,Sahare,M.,Sachdeo,R.,&Kulkarni,N. (2020). Smart Transportation Based Car Car Pooling System. E3S Web of Conferences, 170, 03004. doi:10.1051/e3sconf/202017003004
7. Jaiswal, J. K., &Samikannu, R. (2017). Website of Random Forest Algorithm on

FeatureSubsetSelection and Classification and Regression. 2017 World Congress onComputing and Communication Technologies (WCCCT). doi:10.1109/wccct.2016.25

8. Wen, Z., He, B., Kotagiri, R., Lu, S., & Shi, J. (2018). Efficient Gradient Boosted Decision Tree Training on GPUs. 2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS). doi:10.1109/ipdps.2018.00033
9. Oza, N. C. (n.d.). Online Bagging and Boosting. 2005 IEEE International Conference on Systems, Man and Cybernetics. doi:10.1109/icsmc.2005.1571498
10. Taunk, K., De, S., Verma, S., & Swetapadma, A. (2019).A Brief Review of Nearest NeighborAlgorithm for Learning and Classification. 2019 International Conference on Intelligent Computing and Control Systems(ICCS).doi:10.1109/iccs45141.2019.9065747
11. Ozanne , L.,D. (1999). "Understanding consumer intentions to carpool: a test of alternative models." In Proceedings of the 1999 annual meeting of the Australian Zealand Marketing Academy. smib.vuw.ac.nz (Vol. 8081).Hide Context Google Scholar However, carpooling formallyappearedintheUSinthemid-1970s, after the1973 oil crisis [1].