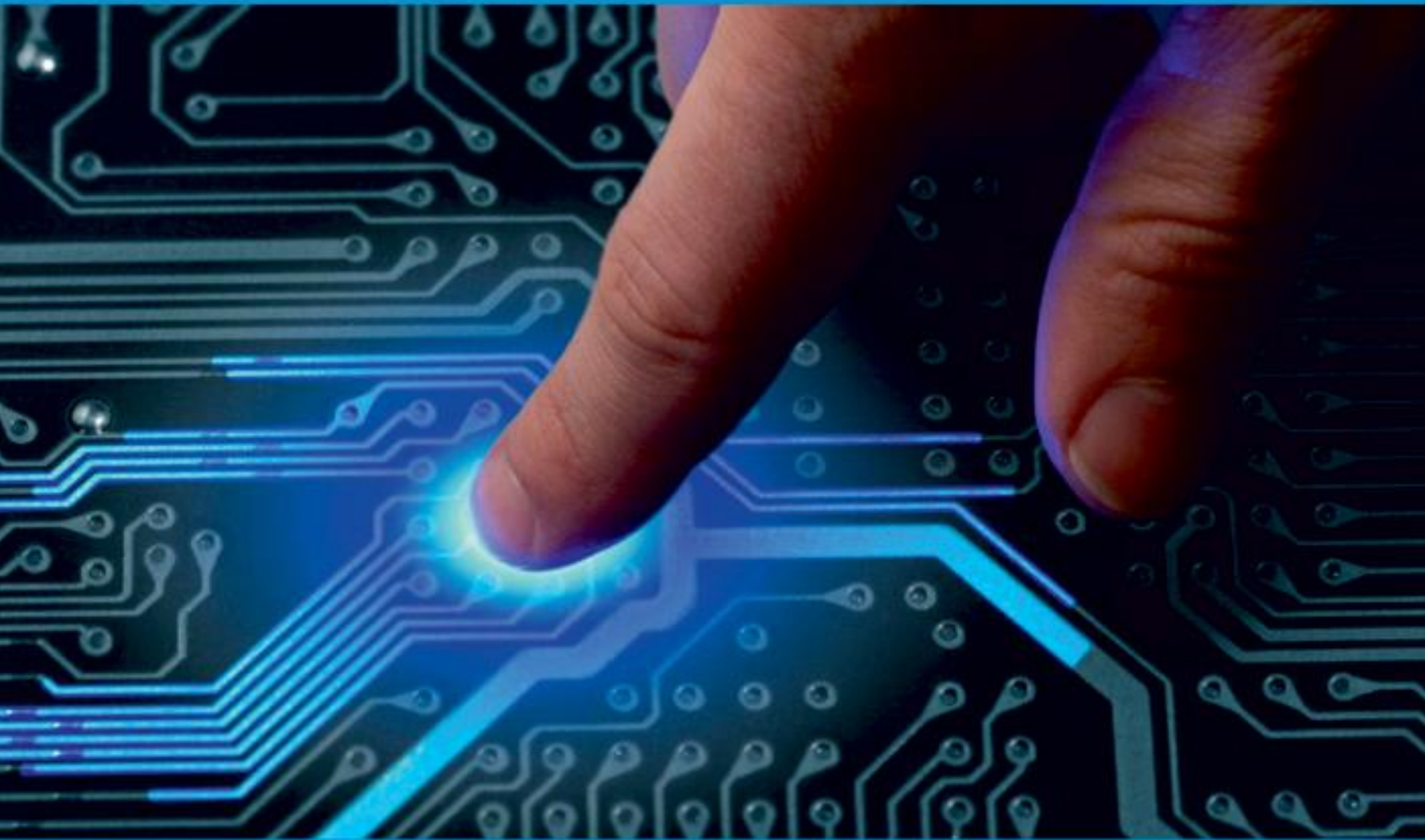
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 5, May 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205337 |

Carpool Crew: Uniting Commuters for a Greener and a Cheaper Ride

Gauri Firke[1], Rinku Bhagwat[2], Yashashri Bhor[3], Sankalp Jagdale[4], Prof. Varsha Gosavi[5]

UG Student, Department of Computer Engineering, SKNCOE, Savitribai Phule Pune University, Pune, India[1 2 3 4]

Professor, Department of Computer Engineering, SKNCOE, Savitribai Phule Pune University, Pune, India[5]

ABSTRACT: Peer-to-peer ridesharing is a project dedicated to establishing a ridesharing platform through decentralizing transactions, thereby ensuring no central authority oversees the transactions or associated data. Additionally, the project aims to achieve equitable and precise pricing by eliminating third-party intermediaries, safeguarding customer privacy, and leveraging smart contracts to automate numerous tasks. Moreover, the implementation of an application-specific cryptocurrency token facilitates seamless exchange between riders and drivers for ridesharing services.

KEYWORDS: Blockchain, Decentralization, Peer-To-Peer Ridesharing, Intermediaries

I. INTRODUCTION

As ride-sharing services experience a surge in demand, numerous companies, such as Uber and OLA, have emerged, each with its unique approach. For instance, OLA in India has incorporated auto-rickshaws alongside traditional cabs, while Rapido offers ridesharing through two-wheelers. However, despite their differences, they all rely on a centralized model for their day-to-day operations, enforcing policies, regulations, and conditions on both customers and drivers.

In an effort to streamline user experience, these ride-sharing providers often engage intermediaries or third-party entities to handle various processes like payments, verifications, and data security. However, this multi-party involvement often leads to transparency issues, inaccuracies in pricing, and compromises in privacy, as data is shared with these intermediaries.

These drawbacks have prompted a thorough examination of blockchain technology, motivating us to propose a solution to the problems inherent in centralized methodologies. Our approach involves transitioning to a decentralized methodology, eliminating third-party involvement, and automating processes using Ethereum smart contracts.

Blockchain, akin to a decentralized and immutable database, offers reliability and distribution across the globe. Ethereum smart contracts, with their predefined rules that cannot be altered once established, provide a means to automate processes without the need for centralized control.

II. METHODOLOGY

This project proposes a decentralized ridesharing model leveraging Blockchain to mitigate the drawbacks of the centralized approach. Additionally, it presents an efficient architecture addressing Blockchain limitations like scalability, network congestion, and key management. The primary objective is to deliver a decentralized application offering a secure ridesharing experience through peer-to-peer communication, facilitating secure transactions between parties. The project adopts several methods to achieve this:

1. Fair and accurate pricing model: By considering factors like weather, peak hours, vehicle type, distance, and passenger count, the project aims to generate precise ride prices. Minimizing third-party involvement ensures only relevant parameters influence the pricing algorithm, benefiting both customers and drivers.

2. Privacy: By reducing third-party involvement, the application securely processes and stores user data within the Blockchain, ensuring user privacy without routing data through external servers.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205337 |

3. Data integrity and security: With transactions processed by smart contracts in a decentralized manner, data integrity is assured as modifications post-transaction are impossible. Additionally, Blockchain's cryptographic security safeguards data stored within it.

4. Token-based transactions: To address Blockchain's key management challenges, the project introduces a dedicated wallet within the application utilizing a token named $PRI for transactions. Users' key pairs are securely managed within the wallet, eliminating the need for users to manage keys manually. Ethereum-compatible ERC20 standards are employed for tokenization, ensuring widespread acceptance.

5. Scalability: To mitigate blockchain scalability issues, the project minimizes data sent through the blockchain, limiting it to textual data. Object data like images, audio, and video are stored and maintained using IPFS storage. This approach significantly improves transaction completion rates, making the application scalable for up to 10,000 users initially.

A. Software Used

The proposed system encompasses two software components: Blockchain development and Android development. Below is the list of software tools, libraries, dependencies, and APIs utilized in the development process:

1. Android Studio 3.8: Employed as the Integrated Development Environment (IDE) for creating native Android applications.
2. Google Maps/Directions API: Utilized to integrate map functionality within the application. The Google Directions API facilitates parsing directions between two specified points on the map.
3. Node v16.13.1: Utilized as both an IDE and a runtime environment for JavaScript and Solidity development.
4. Ganache: A software tool used to host a personal Ethereum-based blockchain, providing a local environment for testing and development.
5. Truffle v5.0.2: A development environment specifically designed for writing smart contracts. It operates on the Ethereum Virtual Machine (EVM), aiding in the creation and deployment of smart contracts.
6. Solidity v0.5.0: The programming language used for implementing smart contracts on the Ethereum blockchain.

B. Proposed Method

The Android application consists of various classes and activities to support its functionalities. Unlike many existing ride-sharing apps like Uber and Ola, which rely on centralized third-party systems, this project opts for a more decentralized approach.

For user authentication, a straightforward UI for login and registration is provided, utilizing Google Firebase authentication at the backend to store and authorize user credentials securely.

The map functionality is integrated into the app through a map fragment, extending classes like Google API Client, Location Listener, and others to enable features such as user location, markers, map type selection, etc.

The application's frontend is defined in XML files, encompassing elements for login/registration, selection activities, map fragments, search bars, and drawers for ride requests and approvals.

To configure the map API, the Google API Client is implemented in the main activity, and the On Map Ready method is utilized for tasks like setting up listeners for map events, setting the user's current location, etc.

In the Manifest file, permissions for internet access, and coarse and fine location are granted. Location listener methods such as on Location Changed, on Connected, on Connection Suspended, on Connection Failed are implemented to handle location updates, with a Location Request object initialized in the on Location Change method.

The search bar's on Query Text Listener feeds the latitude and longitude of the searched address as the destination to the direction parser class. Geocoder is employed to obtain the latitude and longitude of the destination, with the relevant location displayed as a marker on the map. A polyline is drawn between the user's location and the destination, and this data, along with other entities, is transmitted for rider requests via the blockchain network.

The application utilizes a public class Task Request Directions, which operates as an A sync Task to generate a response string serving as a URL for the directions API to parse polylines for directions. Another public class, Task Parser, is responsible for parsing the polyline received from the Directions API using the Direction Parser class to construct polylines on the map from the input parameters.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |
|| Volume 12, Issue 5, May 2024 ||
| DOI: 10.15680/IJIRCCE.2024.1205337 |

C. Modules

For this project, we've developed several modules to showcase various functionalities:

1. Dual Dashboards: Two distinct dashboards are created, one for drivers and the other for customers, providing each user type with relevant information.
2. Location Display: Each dashboard displays the real-time locations of other users, corresponding to their respective roles.
3. User Actions: Users can initiate actions such as calling, renting a car, and making payments directly from the application interface.
4. Blockchain Smart Contract: A blockchain smart contract is implemented specifically for car rentals spanning three days, ensuring secure and transparent transactions.
5. Functionality Demonstration: These modules serve the purpose of showcasing the capabilities of each feature, offering a comprehensive overview of the application's capabilities.
6. Location Details: Upon selecting a specific user near one's location, their details, including latitude, longitude, and name, are displayed in Google Maps, enhancing user interaction and facilitating communication.

III. SYSTEM ARCHITECTURE

Our system architecture revolves around service users, who are the primary participants in the carpooling process, along with car owners who offer their vehicles. The trust authority acts as central governing entity, ensuring the security and trustworthiness of the platform. All of the following components operate on the blockchain, which serves as the foundation for secure and transparent carpooling transactions

1.Service Users: Service users are individuals who intend to use the peer-to-peer carpooling platform for transportation. They can be both riders and drivers, depending on their preferences and requirements. As riders, they seek shared rides, while as drivers, they offer rides to other users in exchange for cost-sharing.

2.Car Owners: Car owners are users who possess personal vehicles that they are willing to share with others on the platform. They register their vehicles and provide access codes for users to book and access their cars. Car owners may also specify the conditions and availability of their vehicles.

3. Trust Authority: The trust authority is a central entity responsible for setting up and governing the decentralized carpooling system. It plays a pivotal role in ensuring the security and trustworthiness of the platform. The trust authority issues credentials to both users and car owners, facilitating secure transactions and interactions on the platform. This entity acts as a trusted intermediary that enhances user confidence and safety.

4. Blockchain: Blockchain is the underlying technology that powers the decentralized carpooling system. It serves as a distributed ledger where all carpooling transactions are securely recorded. The blockchain ensures the transparency, immutability, and security of these transactions, making it resistant to manipulation and providing a high level of trust. Smart contracts, programmed on the blockchain, automate various processes, such as matching drivers with passengers and handling payments, further enhancing the efficiency and security of the system.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |
|| Volume 12, Issue 5, May 2024 ||
| DOI: 10.15680/IJIRCCE.2024.1205337 |

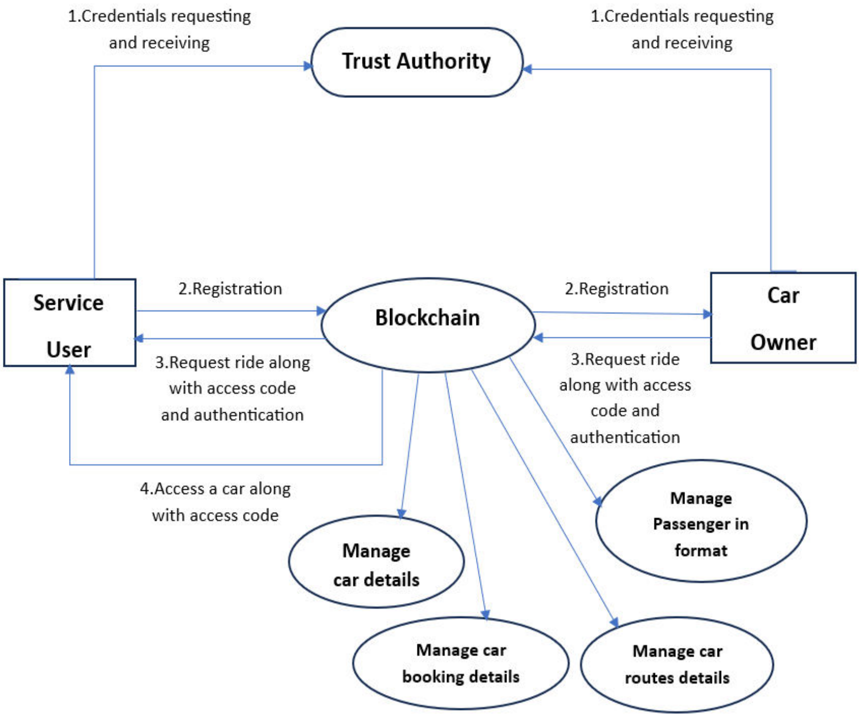


Fig 1: System Architecture

IV. IMPLEMENTATION APPOROACH

The implementation approach adopted for this study is Waterfall Model.

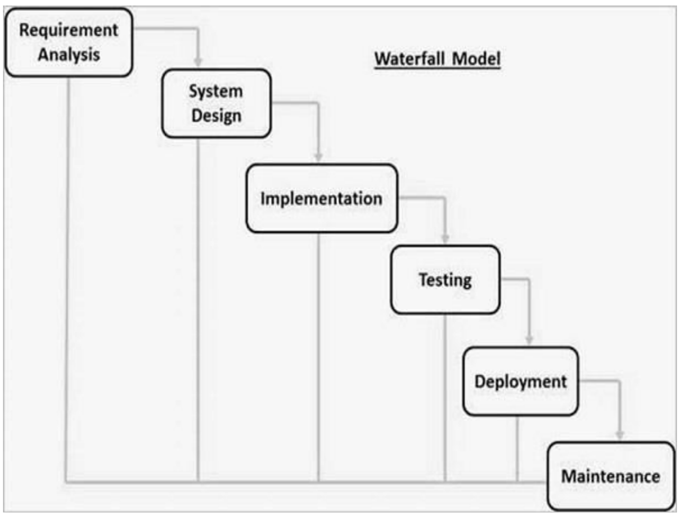


Fig 2: Waterfall Development Model

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205337 |

- **Requirement gathering and analysis:** In this step of waterfall we identify what are various requirements are need for our project such are software and hardware required, database, and interfaces.
- **System Design:** In this system design phase we design the system which is easily understood for end user i.e. user friendly. We design some UML diagrams and data flow diagram to understand the system flow and system module and sequence of execution.
- **Implementation:** In implementation phase of our project we have implemented various module required of successfully getting expected outcome at the different module levels. With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
- **Testing:** The different test cases are performed to test whether the project module are giving expected outcome in assumed time. All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployments of System:** Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.
- **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Maintenance is done to deliver these changes in the customer environment. All these phases are cascaded to each other in which progress is seen as flowing steadily downwards like a waterfall through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.

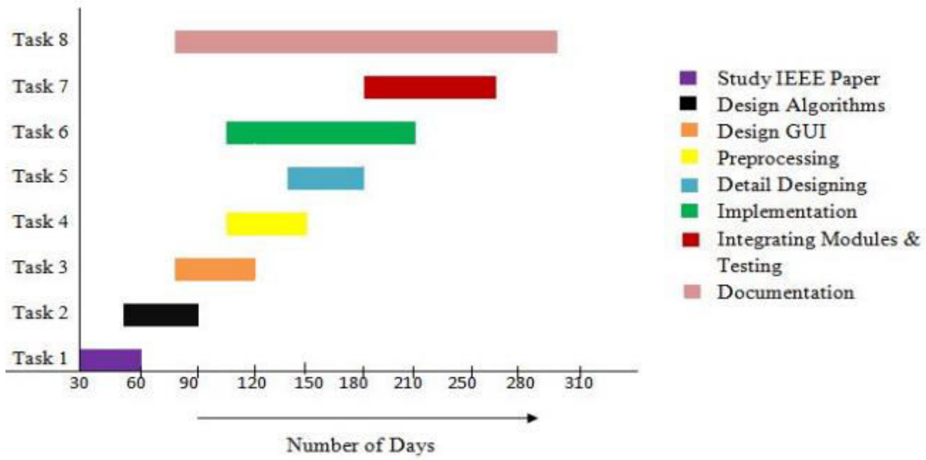


Fig 3: PERT Chart/Gantt Chart

V. SYSTEM REQUIREMENTS

- **Database Requirements**

MySQL : MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site (http://www.mysql.com/) provides the latest information about MySQL software. MySQL is database software which is used to store all database related activities regarding to our project and it is easily stored and retrieve the data.

- **Software Requirements**

Java is an easy to learn, powerful programming language. Java's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for platform Independent and rapid application development in many areas on most platforms. Java is often described as a"batteries included" language due to its comprehensive standard library.

- **HARDWARE REQUIREMENTS**

1.System Type : 64-bit or 32-bit
2.Processor : Intel core i3, 2GHz
3.Storage Capacity : 256 GB• RAM : 4GB (Min)
4.I/O Devices : Mouse and Keyboard

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 5, May 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1205337 |

VI. FUNDAMENTALS OF TEST CASES

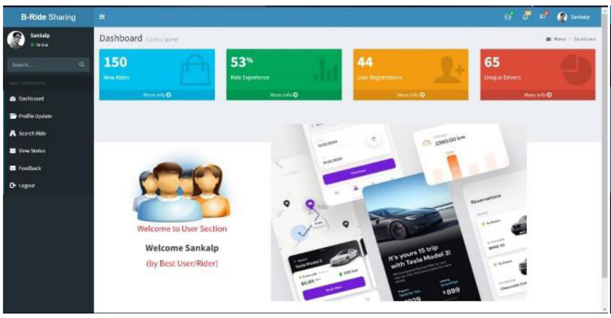
Testing web applications involves several fundamentals principles and practices to ensure the application functions correctly, securely and efficiently. Here are the key fundamentals:

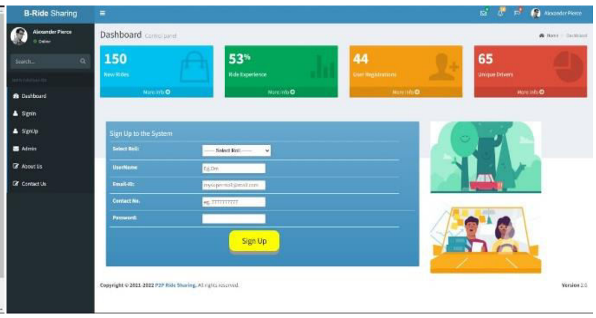
- **Functionality Testing:** Verify that all features and functions of the web application work as intended
- **Usability Testing**: Ensure the application is user-friendly and provides a good user experience.
- **Compatibility Testing:** Ensure the application works across different browsers, devices, and operating systems.
- **Performance Testing**: Assess the application's performance under various conditions.
- **Security Testing:** Identify and mitigate security vulnerabilities.

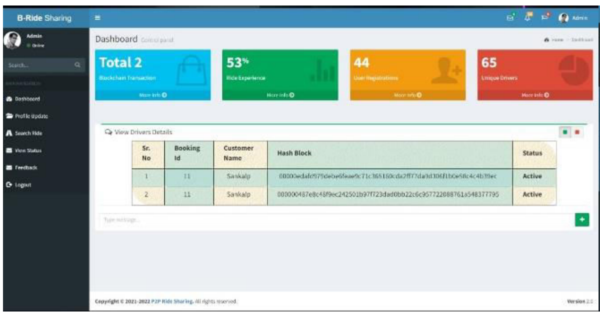
VII. TOOLS AND TECHNOLOGY USED

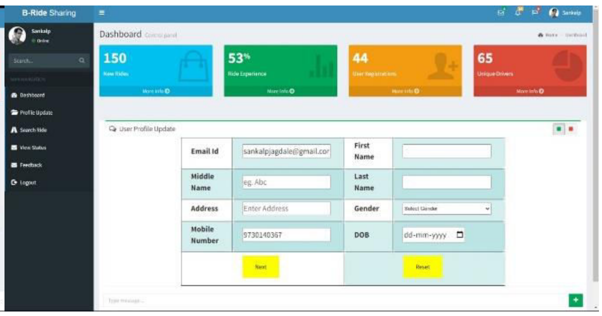
- JAVA
- XAMPP
- JDK
- Apache
- MySQL

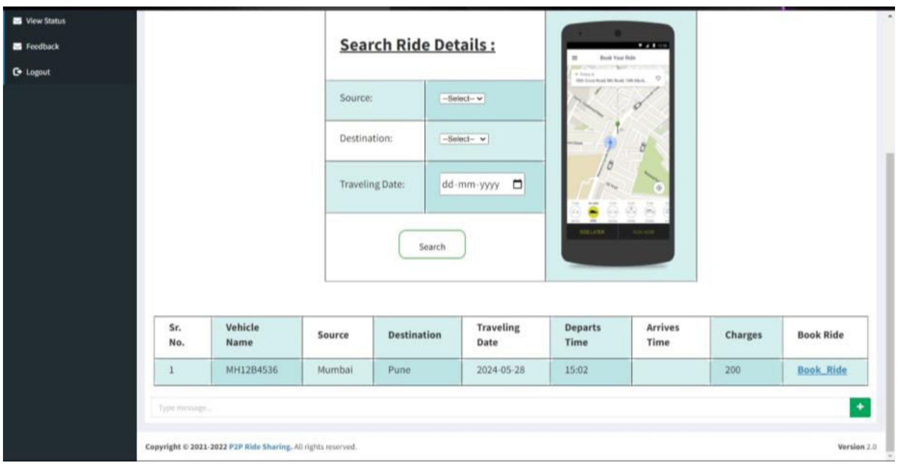
VIII. RESULTS

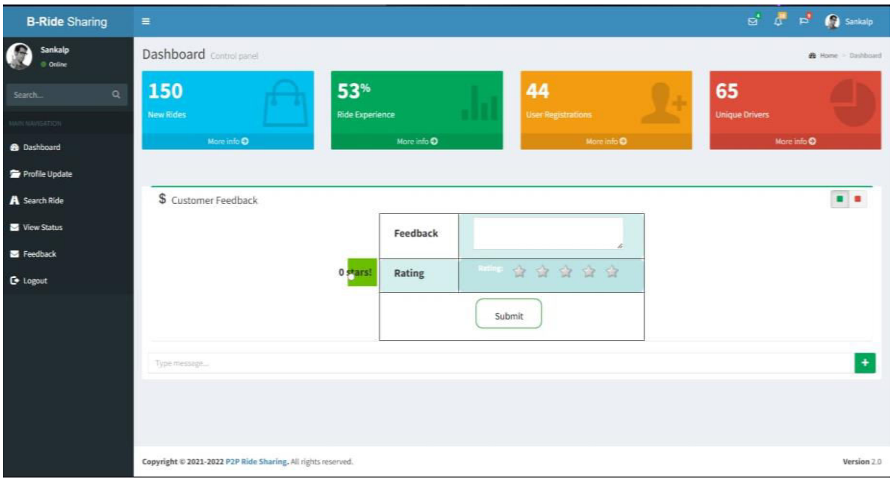












IX. FUTURE SCOPE

Blockchain technology has the potential to enhance transparency, security, and efficiency in the ridesharing sector by enabling decentralized networks that connect drivers and passengers directly. This could lead to a fairer system with lower fees and increased trust between users. However, for blockchain-based ridesharing platforms to succeed, they must overcome challenges related to scalability, data privacy, security, and integration with existing systems.

X. CONCLUSION

While current ride-sharing platforms have gained popularity, they face challenges in pricing models, user safety, transaction transparency, privacy, and data security. Blockchain technology presents an opportunity to address these shortcomings and introduce innovative features. By eliminating intermediaries, blockchain-enabled systems can ensure fair pricing, prioritize user privacy and data security, and enhance transaction transparency. Additionally, smart contracts can automate payment processing, streamlining operations and improving overall user experience.

REFERENCES

[1] Baza, M., Mahmoud, M., Srivastava, G., Alasmary, W. and Younis, M., 2020, May. A light blockchain-powered privacy-preserving organization scheme for ride sharing services. In 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring) (pp. 1-6). IEEE

[2] Yuan, Y. and Wang, F.Y., 2016, November. Towards blockchain-based intelligent transportation systems. In 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC) (pp. 2663- 2668). IEEE.

[3] Wang, D. and Zhang, X., 2020. Secure Ride-Sharing Services Based on a Con sortium Blockchain. IEEE Internet of Things Journal

[4] Pal, P. and Ruj, S., 2019, July. BlockV: A Blockchain Enabled Peer- Peer Ride Sharing Service. In 2019 IEEE International Conference on Blockchain (Blockchain) (pp. 463-468). IEEE

[5] Abuhashim, A. and Tan, C.C., 2020, July. Smart Contract Designs on Blockchain Applications. In 2020 IEEE Symposium on Computers and Communications (ISCC) (pp. 1-4). IEEE

[6] Baza, M., Lasla, N., Mahmoud, M., Srivastava, G. and Abdallah, M., 2019. Bride: Ride sharing with privacy-preservation, trust and fair payment atop public blockchain. IEEE Transactions on Network Science and Engineering.

[7] Chang, S.E. and Chang, C.Y., 2018, July. Application of blockchain technology to smart city service: A case of ridesharing. In 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) (pp. 664-671). IEEE.11

[8] Xu, B., Agbele, T. and Jiang, R., 2020. Biometric Blockchain: A Secure Solution for Intelligent Vehicle Data Sharing. In Deep Biometrics (pp. 245-256). Springer, Cham.

[9] Zhang, X., Liu, J., Li, Y., Cui, Q., Tao, X. and Liu, R.P., 2019, October. Blockchain based secure package delivery via ridesharing. In 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP) (pp. 1-6). IEEE.

[10] Sharma, P.K., Moon, S.Y. and Park, J.H., 2017. Block-VN: A distributed blockchain based vehicular network architecture in smart City. Journal of information processing systems, 13(1).

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details