Journal of Innovative Technology, Vol. 8, No. 2, pp. 37-44, September 2026
http://doi.org/10.29424/JIT.202609_8(2).0004

On-Time Bus: A Smart Real-Time Public Transportation Tracking and Fare Management System using Mobile and IoT Technologies

Muhammad Aown Ali [1], Usman Karamat [2], Majid Bashir [3], Tariq Mahmood [4*]

ABSTRACT

Developing countries still have many difficulties when it comes to the operations of their transport systems. Such difficulties include inefficiency in scheduling, lack of real-time information provision, and outmoded fare collection techniques. All these problems impact the users and transport operators negatively. The passengers have difficulties predicting the arrival of the buses, and also end up having longer waiting times. Transporters also find it hard to manage their vehicles' schedules and the fare collection process. Therefore, this paper presents an innovative solution known as On-Time Bus, which is a real-time management system of public transportation using the mobile application and IoT infrastructure. The new system helps users track their buses via mobile applications, receive updated information concerning the bus routes and schedules, and use the automatic touch-on touch-off fare payment technique. Operators will be able to monitor and control their vehicles easily and efficiently via the provided online management platform. The system architecture has been built employing React Native software for creating cross-platform mobile applications, Firebase for NoSQL database services on the cloud that can synchronize the data in real time, and an IoT hardware module built using Arduino for implementing real-time tracking of buses using GPS technology and automated processing of payments. Such an integration makes the communication process between hardware, cloud services, and user-end applications efficient. The proposed framework has been analyzed in terms of performance and feasibility through experimenting with real-life simulations of use cases. From the tests conducted, it can be seen that the proposed architecture offers better efficiency and reduces passengers' waiting time considerably. Moreover, it has proven to be highly flexible and cost-efficient, and hence applicable in developing countries for upgrading their transport systems.

Keywords: Smart transportation; real-time tracking; IoT; public transportation automation; mobile and computer applications.

1. INTRODUCTION

Across the world, automation [1, 2] has already been implemented in most manual systems. Especially in the world of automobiles and transportation, automation and technology have become a significant part of the industry [3, 4], without which the industry cannot imagine a future, especially in the world of AI, where work is going on a large scale in system identifications and neural network fields [6-10]. However, in Pakistan, we can observe a lack of innovation and technological advancements, which can prove to be highly beneficial for the public as well as businesses. To solve such a problem, we have come up with a brilliant "Soft-Solution" which would ease the lives of millions daily. The main purpose of this project is to bring ease into the lives of travelers who use the outdated transport system daily. We aim to bring innovation into this field, which is already underway in countries other than ours. Thus, there is a need to launch this project in this region to get this country up to speed in this field.

This project will be able to revolutionize the way public transport operates. The clients of this app will be able to add value to their businesses. The application of this system will be in the transportation industry. As there is a need to bring forth automation and innovation in the field of people moving. Hence, there is a need to launch this project in this region to get this country up to speed in this field. This system does exactly that. On-Time Bus is a smart bus scheduling system [11, 12] that allows users to view, track, and plan their journey on a given bus service. This system also acts as a reminder system for the drivers to allow them to be punctual and track their progress on a given route by using GPS. This system would also enable the business owners/managers to manage their running buses/ routes/ drivers. They could also track the buses running on a particular route. Certain transportation apps are currently in business in this region, some of which include: SWVL, Careem, Uber, etc. Even though these systems are similar in terms of the area of an industry being implemented, these systems are missing some of the aspects that are being provided in the On-Time Bus service. Our service is being provided to the public to check the bus Schedule and track the progress of the bus on its respective route. This platform provides the customers and the bus operators with a completely stand-alone and all-in-one system, including payment, journey management, tracking, etc., but can also be implemented as an extra feature and a convenience of

Manuscript received January 31, 2026; revised June 8, 2026; accepted June 25, 2026.

[1] Masters Student, Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Taiwan, R.O.C.

[2] Developer, Enableps, 664 Collins Street, Docklands, Victoria, Australia

[3] Lecturer, Department of Computer Science, International Islamic University, Pakistan

[4*] Lecturer (corresponding author), Department of Mathematics and Statistics, International Islamic University, Pakistan. (email: tariq.mahmood@iiu.edu.pk)

the bus operator's business, which would increase the worth of the company as a business in the eyes of the customers. The On-Time-Bus system acts as an intermediary between the end-user (passenger) and the service provider (Business owner/Manager) and fills the gaps in communication using the latest technology, which is not necessarily human-to-human communication.

With the development of this system, we aim to take out the "guesswork" and ambiguity involved in the automobile industry, for example, guessing how long it would take for a bus to arrive at a particular stop. The main objective of this system is to eliminate the feeling of isolation when waiting for the bus. No more asking other people, no more fidgeting, no more anxiousness. All of that goes away with the tap of a button. The users will be able to view their current location and find the nearest bus stop. The users can look for the schedule of the next arriving bus. Additionally, the system will also keep the users informed about the whereabouts of the bus in real-time. The bus operators go through a lot of hassle while chasing their fleet. They must constantly keep calling the drivers to let them know their location. With GPS-enabled buses. All they need to do is log on to their systems, and the buses will appear right in front of their eyes. This reduces botheration on both sides, the bus drivers as well as the service operators. The travelers will be able to travel without the worry of prior bookings, taking time out to inquire about the schedules, and being ready before they need to travel. This system will allow them to "Touch-On and Touch-Off" as they are ready to travel, and they are present at the designated bus stops. All they will need to do is scan a QR code present at the bus stop, and the system will register that as the start of their journey. While disembarking from the bus, they need to scan the QR code again, and the system will register that as the end of that journey. With the pre-coded algorithm, the system will calculate the fare of the journey and automatically deduct that amount from the users' virtual wallets. This will allow fairness for both service providers and the customers, as they will only be charged for the distance that they have traveled. The On-Time Bus will allow the bus operators to manage their fleet, assign fleet to routes, manage routes, and manage the regions that their service extends. Certain transportation apps are currently in business in this region, some of which include: SWVL [13], Careem [14], Uber [15, 16], etc. Even though these systems are similar in terms of the area of an industry being implemented, these systems are missing some of the aspects that are being provided in the On-Time Bus service. Our service is being provided to the general public to check the bus Schedule and track the progress of the bus on its respective route. This platform provides the customers and the bus operators with a completely stand-alone and all-in-one system, including payment, journey management, tracking, etc., but can also be implemented as an extra feature and a convenience of the bus operator's business, which would increase the worth of the company as a business in the eyes of the customers.

1.1 Key Contributions

The following are the research contributions of our project/product.

- Real-time IoT bus tracking.
- QR Touch-On/Touch-Off fare automation.
- Distance-based automatic fare calculation.
- Unified passenger and operator platform.
- Low-cost implementation using commodity hardware.

While the current generation of cloud-reliant IoT-based tracking systems collapses when the quality of the cellular signal is degraded, the current paper proposes a robust and economical method of monitoring transit systems. The innovations and contributions of the current study are listed below:

- Robust Data Architecture: The development of a new state management system that allows synchronization of data without the need to burden the cloud database while ensuring data persistence across volatile connections.
- Resource-Friendly IoT Edge Architecture: Design and implementation of a cheap IoT hardware tracking device with a software watchdog that ensures system uptime despite difficult operating conditions.

The remaining subsections of the manuscript contain: Section 2 contains System Analysis; Section 3 includes System Designs, Section 4 contains Implementation and outcomes; Section 5 has information related to supplementary materials, and the last section 6 includes conclusion and future work.

2. SYSTEM ANALYSIS

Users of this system are described in more detail below:

2.1 Passengers

Bus-Scheduling: The system will allow users to view the bus schedule for its route and bus stop. The user will also be displayed at the nearest bus stops that are accessible from the user's geographical location.

Bus Tracking: The user can look up a particular bus stop and view how long it will take for the bus to arrive.

Bus-Stop Search: The user can also search for a particular bus stop using keywords and predictive suggestions.

Touch-On/Off system: The system will also contain a touch-on and off system, which will allow "spontaneous customers" to pay for the bus service without any prior booking. The system will have a QR code scanner and a QR code associated with each bus stop.

2.2 Bus-Operators

Fleet Management: This system would allow the business owners/managers to manage their running buses. They could also track the buses running on a particular route.

Route Management: This system would also enable the business owners/managers to manage the routes to which their services extend.

User Management: This system would let the business owners/managers manage their registered clients. They can review the client's activities in case any anomalies in the system need to be corrected. And also modify their user profiles.

Bus stop Management: This system would also enable the business owners/managers to manage their bus stops.

2.3 Requirement Analysis

Requirement analysis analyzes, defines, and specifies the requirements needed to build a good application, which may include the application's functionality, constraints, and goals understandable and acceptable to the client, developer, and user. It also defines the input, output, and dependencies of the application. The On-Time-Bus system acts as an intermediary between the end-user (passenger) and the service provider (Business owner/Manager) and fills the gaps in communication using the latest technology, which is not necessarily human-to-human communication. With the development of this system, we aim to take out the "guess-

work" and ambiguity involved in the automobile industry, for example, guessing how long it would take for a bus to arrive at a particular stop.

2.4 Software and functional requirements

The proposed system is built using React Native CLI to facilitate cross-platform mobile application development using a common JavaScript codebase for both Android and iOS platforms, which increases application efficiency while providing better performance, improved user interface design, and faster loading times. The backend architecture is built using Node.js, Django, and Firebase to facilitate secure data processing, application functionality, and cloud-based storage solutions. For the implementation of the proposed system, the hardware requirements would be the presence of a mobile device that is capable of accessing real-time location information and gallery facilities, while the minimum hardware requirements for the deployment of the proposed system would be 4 GB RAM and an Intel Core i5 processor of the 5th generation or later. The proposed system would be equipped with an extensive set of functional requirements, including user registration and login functionality, real-time bus tracking, graphical representation of the bus schedules, and touch-on and touch-off functionality to facilitate spontaneous travel without any prior booking through the scanning of the bus stop's QR code. The proposed system would be equipped with an electronic wallet facility referred to as TraWallet, which would facilitate automated fare calculation, bill generation, and subsequent deduction of the payment amount based on the distance covered during the travel. The proposed system would also be equipped with management facilities to enable the concerned service providers to monitor user accounts, vehicle management, route management, and bus stop management. The proposed system would be developed to meet an extensive set of non-functional requirements, including high performance during the execution of the system under practical constraints, constant availability, ease of maintenance, portability between different platforms, secure access control mechanisms, reliability, usability, and understandability. The domain model and use case diagram of the proposed system are depicted in Figures 1 and 2. The justification for selection of the used technologies are given in Table 1.

Table 1 Justification for use of selected technologies

Technology	Reason
React Native	Android iOS single codebase
Firebase	Real-time non-relational database
Arduino	Low cost IoT hardware
Google Maps API	Accurate location services

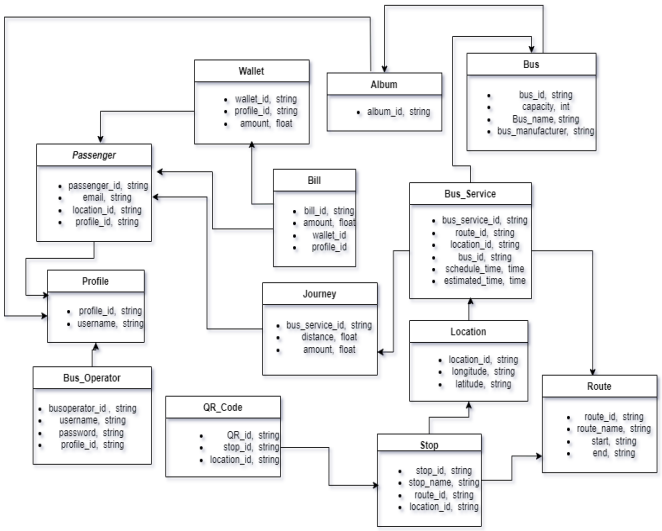


Fig. 1 Domain model of the proposed system

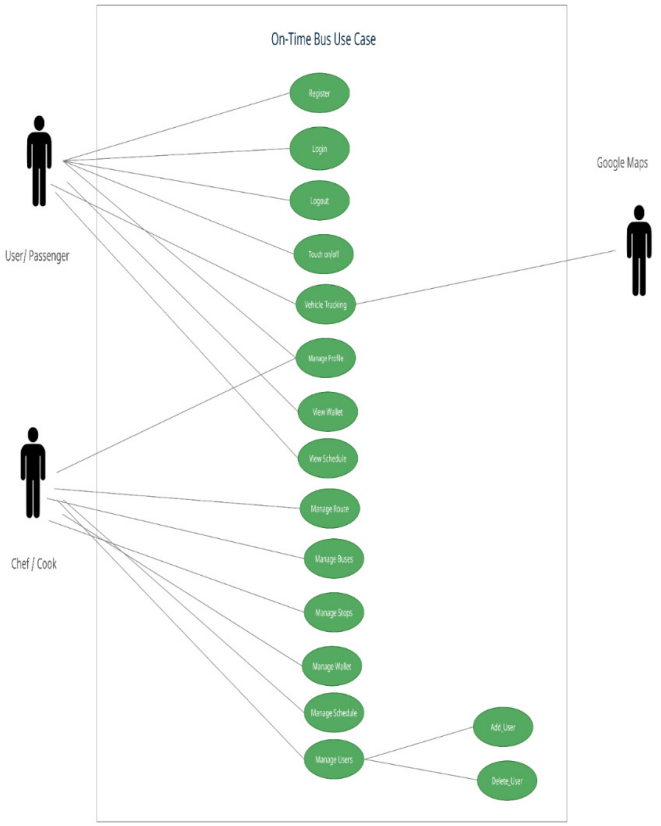


Fig. 2 Use case diagram of the proposed system

3. SYSTEM DESIGN

The design of the solution has been made in such a way as to ensure modularity, scalability, and proper interaction among the components of the system. The design of the entire system has been modeled using the sequence diagrams in order to show the interaction process among the users, components of the system, and the backend services in relation to the major operations of

the system, such as registration, login, real-time tracking, and payment of fares. The class diagram and entity relationship diagram are used to design the static system structure in terms of classes, attributes, and methods. This method helps in efficient database management concerning users, buses, routes, schedules, and transactions. Activity diagrams are used to detail the flow of major processes in the system. This helps in understanding the decision and concurrent parts of the system processes. These design models help in ensuring the system design is functional and easy to manage. The sequence diagram of the user and admin is shown in Figures 3 and 4, and the class diagram and activity diagram of the proposed system are represented in Figures 5 and 6, respectively.

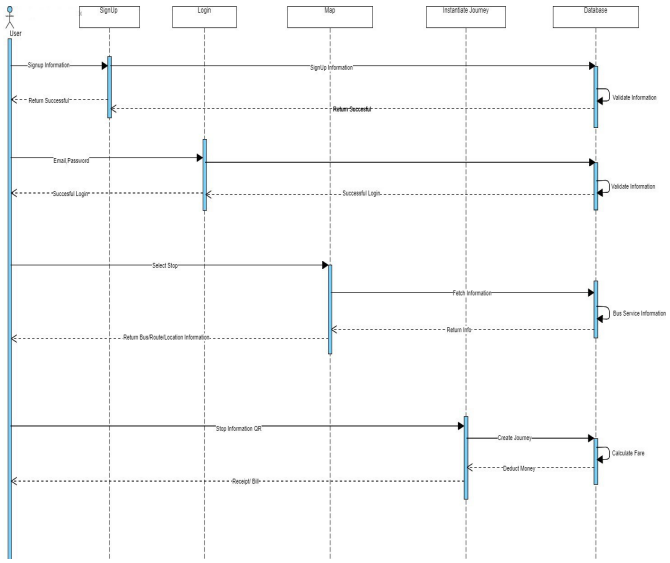


Fig. 3 System sequence diagram of the user

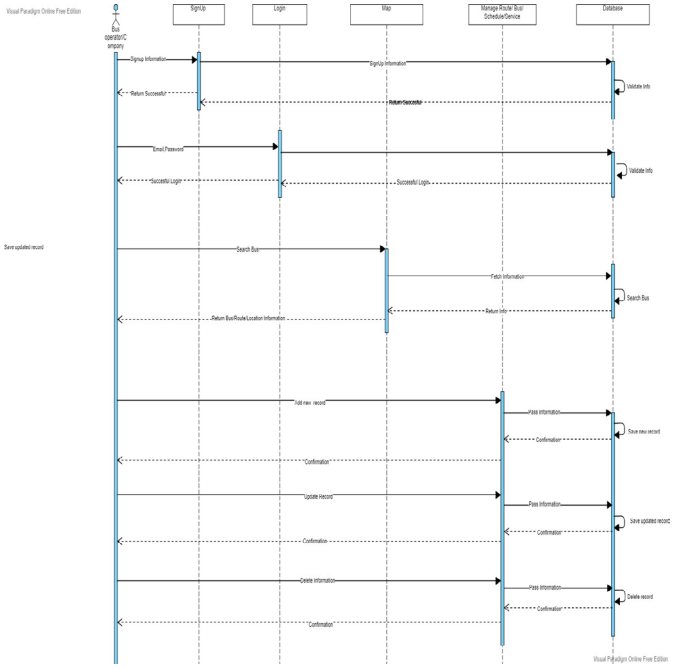


Fig. 4 System sequence diagram of the admin

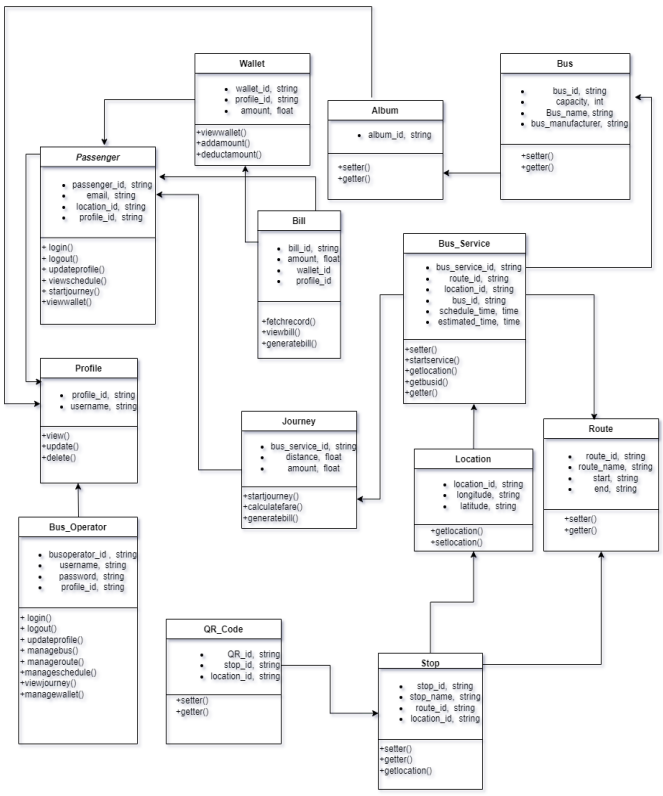


Fig. 5 Class diagram of the proposed system

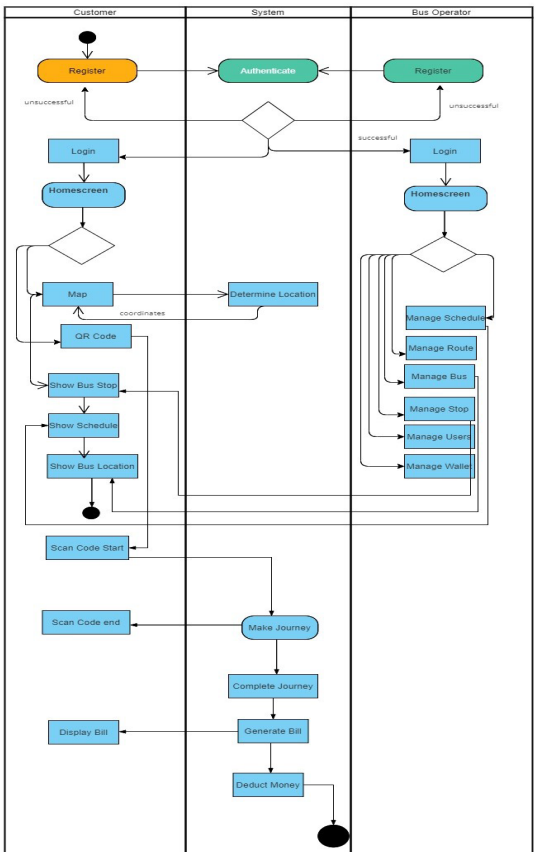


Fig. 6 Activity diagram of the proposed system

3.1 Touch-On/Touch-Off Security Mechanism

Figure 7 depicts the security mechanism of touch-on/touch-off implemented in the suggested smart fare management system design. It starts with the authentication of passengers using Firebase Authentication, and then a journey session is started via scanning a QR code. Once this is done, an individual journey ID is assigned, along with the corresponding timestamp and GPS coordinates. When the journey ends, the distance is calculated, the fare is deducted, and then the details of the fare transaction are logged into the system. The security of this approach is improved by implementing JWT authentication, HTTPS, RBAC, and QR code validation.

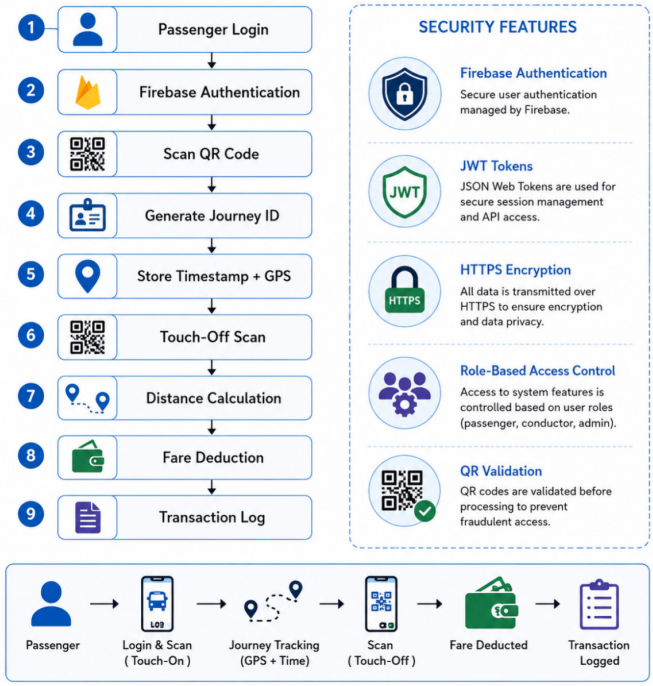


Fig. 7 Touch-On/Touch-Off Security Mechanism

3.2 System Architecture and Data Flow

The architecture of the proposed On-Time Bus system is a multi-layered one, composed of a mobile application layer, an IoT tracking layer, a backend services layer, and a cloud database layer. The mobile application for passengers and the operator dashboard exchange data with the backend services via secure REST APIs. Firebase Realtime Database servers for saving information about the users, routes, timetables, journeys made by them, and the current vehicle position.

Arduino-powered IoT device installed in each bus receives location data and transmits this information to Firebase via backend services, keeping everything synchronized and up-to-date within a few seconds. These data are presented on the screens of the passenger mobile application and operator dashboard. The API from Google Maps will be used for displaying the buses' locations, routes, and neighboring bus stations.

For achieving scalability, the cloud platform of Firebase enables automatic synchronization, load balancing, and high efficiency for processing a large number of concurrent users. The design of the software with its multi-layer structure enables its scaling separately for mobile clients, backend services, and database independently of each other.

4. IMPLEMENTATION AND OUTCOMES

The proposed On-Time Bus system demonstrates its effectiveness as it increases the efficiency of the public means of transportation. By conducting functional testing and scenario validation, the proposed system allows for real-time location of the buses, an accessible graphic representation of the schedules, as well as automated fare computation. This eradicates the waiting time and the doubts of the commuters. The touch-on/off functionality simplifies the processing of the fares without the need for any pre-booking, which favors the spontaneous commuters. On the part of the operator, the implementation of the system allows for greater visibility and greater control through central monitoring. Delays and routes can be identified more efficiently, keeping the whole fleet running according to schedule. This new system also cuts back on the need for verbal communications from drivers to operators, which would result in faster and more accurate decisions. It appears that the results offer an optimal method for managing transportation, particularly in regions that have not yet developed automation. In sum, the outcome of these findings demonstrates benefits to service quality, efficiency, and satisfaction. While the purpose of these assessments certainly pertains to functionality and practical applications, the tool has succeeded at what it was intended to do. This effectively indicates there is a potential to be made to upgrade public transit systems and begin working on further developments, such as the prediction of arrivals and intelligent traffic patterns. The system has been developed using current trends in web and mobile technology for performance, reliability, and platform independence. The admin dashboard has been created using React, which has a component-based solution for efficient and reusable user components for UI development. The declarative programming paradigm in React helps with predictable and debug-friendly state management, which helps with complex admin tasks such as managing routes, users, and fleets. The mobile app is developed using the React Native platform, which enables the app to run on Android and iOS platforms using the same code and supports near-native performance. Firebase is employed as a cloud-based NoSQL database for real-time data synchronization, secure storage, and easy integration with the backend. Arduino-based hardware elements are integrated to provide real-time vehicle tracking data, which is then processed and displayed on the mobile app. Microsoft Visual Studio Code is employed as the main development tool because of its flexibility, language neutrality, and adaptability to emerging development tools. A thorough testing phase is required to assess the performance, functionality, and reliability of the system in a manner that is very close to practical usage. Component-testing is also done to assess the accuracy of various functions and modules of a system, which can lead to the identification of solutions to potential discrepancies in the initial stages. Integration testing is required to ensure that various subsystems work correctly when integrated. User and acceptance testing are done in a process where the system is run under the best- and worst-case scenarios, and the system functionality is verified for conformity to the system specifications. The test confirms the system runs well without any functional defects and performs well all the time, producing the correct results. Completion of all the test scenarios confirms the system is stable, reliable, and ready for implementation in the real-world environment. Table 1 shows the test case id, test case scenarios, data, step, results, and whether the system passes or fails the following test case.

Table 2 Test cases of the proposed project

Test Case ID	Test Scenario	Test Steps	Test Data	Results	Fail/Pass
TU01	Compatibility	Run the Application on different browsers	N/A	The frontend and backend servers are working	Pass
TU02	Sign up (User)	1) Click on the Sign-up button in the top right corner 2) Fill in the Sign-up form 3) Click on the Sign-up button	First name= Usman Last name= Karamat Email= muhammed.usman77@gmail.com Password=Soccer#95	The user is signed up successfully. Please verify your email	Pass
TU03	Sign in	1) Open the mobile app 2) Fill in the sign-in data 3) Click on sign in button	Email= muhammed.usman77@gmail.com Password=Soccer#95	Complete your Profile template is displayed	Pass
TU04	Show Map	Toggle Map Screen from Bottom tab navigator	Current location	Bus stops near the current location	Pass
TU05	Scan the card on the provided RFID machine	Mobile App Navigates automatically to the Journey Screen.	Mobile App Navigates automatically to the Journey Screen.	Journey details are shown.	Pass
TU06	Scan the card on the provided RFID machine.	Journey Ends automatically, and the app navigates to the journey summary screen.	The journey summary screen is shown.	Journey details are shown.	Pass
TU07	Show Bus Stops	Search for the stop name using the search	Stopname: Iqra College	Related results shown	Pass
TU08	Show Stop Schedule	Touch a particular bus stop	Touch a particular bus stop	Stop schedule shown	Pass
TU09	Show bus location.	Scroll on the map to find the Bus	Touch on the bus	Bus journey details shown	Pass
TU10	Sign in (Admin)	Open the Webapp and log in using credentials.	Username= admin Password = Admin@123	Signed in Successfully	Pass
TU11	Manage Schedule	Navigate to the schedule screen using the side navigation panel	CRUD appropriate details	Success	Pass
TU12	Manage Route	Navigate to the Route screen using the side navigation panel	CRUD appropriate details	Success	Pass
TU13	Manage Stop	Navigate to the Route screen using the side navigation panel	CRUD appropriate details	Success	Pass
TU14	Manage Users	Navigate to the Route screen using the side navigation panel	CRUD appropriate details	Success	Pass
TU15	Manage Wallet	Navigate to the Wallet screen using the side navigation panel	CRUD appropriate details	Success	Pass

In order to measure the efficiency of the proposed On-Time Bus System, it was compared against popular transportation apps such as Uber and SWVL, considering the criteria for developing nations. The said comparison is given in Table 3.

Table 3 Comparison with state-of-the-art systems

Evaluation Metric	Uber / Ride-Hailing	SWVL / Premium Shuttles	Proposed On-Time Bus System
Primary Target	Private/On-demand transit	Premium fixed-route commuters	Low-income mass public transit
Connectivity Dependency	High (Requires continuous 3G/4G/5G)	High (Requires active data for booking/tracking)	Low (Supports offline caching & SMS backup)
Hardware Overhead	High-end smartphone required	High-end smartphone required	Low-cost IoT (Arduino-based) + Basic smartphone
Affordability	High cost per trip	Medium to high cost per trip	Extremely Low (Public transit rates)
Data Resilience	Fails during network drops	Fails during network drops	Sustained via local storage & auto-sync

4.1 Practical development challenges and limitations

Despite the On-Time Bus system being developed in such a way that it is economical and scalable, implementing the system in practical scenarios, particularly in developing areas, faces some hurdles. The issues, along with their mitigation methods, are highlighted below and visually represented in Figure 8.

- Poor Internet Connectivity: Poor or intermittent internet connectivity can impact the syncing of data in real time, location tracking, and verification through QR codes. For the purpose of overcoming this hurdle, the application makes use of offline caching in the mobile application, and journey data is stored locally on the phone.
- Drift in GPS (5-20 m): GPS might suffer from signal interference due to any obstructions, environmental conditions, or even urban canyons, causing a drift in position (about 5-20 meters). This problem is resolved by implementing the techniques of coordinate smoothing, regular testing, and map-matching.
- Hardware Failure: Internet of Things devices, such as the GPS module used in Arduino, may be subjected to hardware failures, improper connections, or even physical damage. The system has a built-in watchdog feature that can detect any device inoperability and alert operators immediately.
- Battery Drainage: Constant data transmission and tracking via GPS drains the battery of the device. For minimizing this effect, power-efficient circuitry is implemented, along with sleep mode functionality. Besides, low-battery alarms ensure prompt action by the operators.
- Limitations Related to Rural Areas' Network Coverage: In most cases, many rural locations experience limited coverage of 3G/4G. Hence, there will be delays in collecting and uploading the data due to limited network coverage. This system was designed to collect and keep data stored temporarily until the data can be transmitted through the network.
- Limitation in Real-World Application: The present system design has been successfully tested via functionality, integration, and acceptance tests. Nonetheless, the system has yet to be deployed on a significant scale in the context of actual public transport operations. In light of this limitation, it has not been possible to evaluate the performance parameters, satisfaction levels, and feedback of the passengers and operators in this particular study. Future research shall include the deployment of the system at the pilot level using public transport operators and passengers.

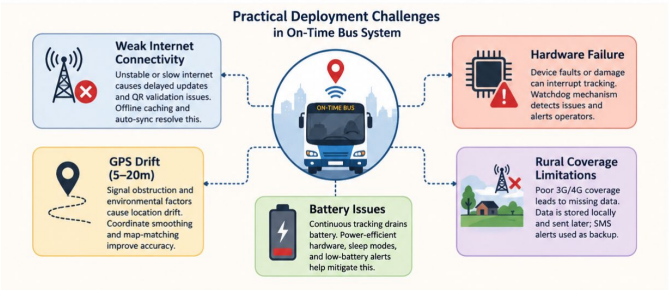


Fig. 8 Challenges in development and limitations of the proposed project

4.2 Performance Evaluation

In addition to functional testing, the proposed On-Time Bus solution was quantitatively evaluated through various parameters assessing the responsiveness and effectiveness of its operation. Specifically, the analysis included location update latency, processing time for QR codes, GPS accuracy, and database synchronization delays. Performance results attained during system testing were presented in Table 4 below.

Performance Metric	Observed Value
Average GPS Tracking Accuracy	±5-10 m
Location Update Delay	2.1 s
Firebase Synchronization Delay	1.4 s
QR-Code Processing Time	0.8 s
Average Mobile Application Startup Time	2.3 s

The acquired data proves that the proposed solution offers real-time performance capabilities alongside efficient fare-processing functionality. The small synchronization delays and rapid QR code processing contribute to the smooth experience, and the GPS accuracy is sufficient for public transportation monitoring.

5. SUPPLEMENTARY MATERIALS

The screenshots of the proposed system's functionality, workflow, and working are provided in the supplementary material file.

6. CONCLUSION AND FUTURE WORK

On-Time Bus is a smart bus scheduling system that allows users to view, track, and plan their journey on a given bus service. This system would also enable the business owners/managers to manage their running buses/ routes/ drivers. They could also track the buses running on a particular route. The system utilizes the modern multi-Platform app development tool, React-Native. It also utilizes APIs from Google Maps. The On-Time-Bus system acts as an intermediary between the end-user (passenger) and the service provider (Business owner/Manager) and fills the gaps in communication using the latest technology, which is not necessarily human-to-human communication. This system also takes 3 out of the "guesswork" and ambiguity involved in the automobile industry, for example, guessing how long it would take for a bus to arrive at a particular stop and pay for its ride through RFID, and for the admin panel, an admin always sees users' data, adds routes, add buses using the web. The application is designed for a user of local buses, and the user always requires an internet connection to watch bus schedules, routes, and real-time locations. For the user to ride on a bus and to pay for a ride, he/she should register in the app. Admin should be logged in to use/access/see the data for the user, schedule, buses, etc. Whereas the current implementation confirms the possibility of using the proposed inexpensive and robust tracking solution, some plans are being made to improve upon the future versions of the solution in question with the help of different algorithms [17]:

ETA Prediction Using Machine Learning: It is hoped to use the location data from our IoT units to develop machine learning solutions (e.g., LSTMs) capable of predicting the Estimated Times of Arrival based on such variables as time of day, weather conditions, and potential historical traffic jams.

AI-Powered Route Optimization: Further versions of our tracking solution are expected to include the use of genetic algorithms and reinforcement learning techniques in order to optimize public transport bus routes based on real-world events like sudden road obstructions or peak travel times.

REFERENCES

[1] Wang, S., Qiao, S., Li, Z., Zhao, W., Yang, H. F., Zhang, G., & Xu, Y. (2026). "Do people believe that automated vehicle is the solution to impaired driving? insights from a statewide survey study." *Transportation Research Part F: Traffic Psychology and Behaviour,* **116**, 103411.

[2] Pikner, H., Sell, R., Majak, J., & Karjust, K. (2022). "Safety system assessment case study of an automated vehicle shuttle." *Electronics,* **11**(7), 1162.

[3] Papulová, Z., Gažová, A., & Šufliarský, Ľ. (2022). "Implementation of automation technologies of industry 4.0 in automotive manufacturing companies." *Procedia Computer Science*, **200**, 1488-1497.

[4] Papulová, Z., Gažová, A., & Šufliarský, Ľ. (2022). "Implementation of automation technologies of industry 4.0 in automotive manufacturing companies." *Procedia Computer Science,* **200**, 1488-1497.

[5] Devaraj, E., Kakaravada, I., & Kumar, N. (2025). "Robotics, Automation, and Artificial Intelligence's Future in the Automotive Sector." *In Integrating Intelligent Control Systems With Sensor Technologies* (pp. 281-310). IGI Global Scientific Publishing.

[6] Ali, M. A., Chaudhary, N. I., Khan, T. A., Mao, W. L., Lin, C. C., & Raja, M. A. Z. (2024). "Design of key term separated identification model for fractional input nonlinear output error systems: Auxiliary model based Runge Kutta optimization algorithm." *Chaos, Solitons & Fractals,* **189**, 115696.

[7] Ali, M. A., Chaudhary, N. I., Khan, T. A., Mao, W. L., Lin, C. C., Khan, Z. A., & Zahoor Raja, M. A. (2025). "Auxiliary Model-Based Chameleon Swarm Optimization for Robust Parameter Estimation of Fractional Order Nonlinear Hammerstein Systems." *Journal of Computational and Nonlinear Dynamics,* **20**(9), 091005.

[8] Arain, M. B., Mahmood, T., Al-Yarimi, F. A., Ali, M. A., Shaheen, S., & Hu, J. (2026). "On the study of acoustic-driven bubble dynamics in Williamson Fluids with NARX neural networks." *Engineering Applications of Artificial Intelligence,* **165**, 113276.

[9] Ali, M. A., Chaudhary, N. I., Malik, M., & Abbas, M. Q. (2025). "Artificial gorilla troops optimization algorithm for key term separated input non-linear output error system identification." *Journal of Innovative Technology,* **7**(2), 37-43.

[10] Ali, M. A., Chaudhary, N. I., Lin, C. C., & Mao, W. L. (2025). "Mountain Gazelle optimization algorithm for identification of nonlinear Hammer-stein output error systems." *Journal of Innovative Technology,* **7**(1), 117-124.

[11] Shah, M. J., Prasad, R. P., & Singh, A. S. (2020, April). "IOT based smart bus system." *In 2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA)* (pp. 130-134). *IEEE.*

[12] Ghareeb, M., Ghamlous, A., Hamdan, H., Bazzi, A., & Abdul-Nabi, S. (2017, September). "Smart bus: A tracking system for school buses." *In 2017 Sensors Networks Smart and Emerging Technologies (SENSET)* (pp. 1-3). *IEEE.*

[13] Hamdy, H., El Deen, B. H., El-Toukhy, I. M., Elsagheer, A., & Mohamed, A. (2021). "Online bus services: A view of Swvl Company." *In Handbook of research on decision sciences and applications in the transportation sector* (pp. 178-188). *IGI Global.*

[14] Javaid, A., Javed, A., & Kohda, Y. (2019). "Exploring the role of boundary spanning towards service ecosystem expansion: A case of Careem in Pakistan." *Sustainability,* **11**(15), 3996.

[15] Mouratidis, K. (2022). "Bike-sharing, car-sharing, e-scooters, and Uber: Who are the shared mobility users and where do they live?." *Sustainable Cities and Society,* **86**, 104161.

[16] Bąk, A., Nawrocka, E., & Jaremen, D. E. (2022). "Sustainability" as a Motive for Choosing Shared-Mobility Services: The Case of Polish Consumers of Uber Services." *Sustainability,* **14**(10), 6352.

[17] Ali, M. A., Chaudhary, N. I., Khan, Z. A., Mao, W. L., Lin, C. C., & Zahoor Raja, M. A. (2026). "Design of flow direction optimization paradigm for fractional nonlinear hammerstein output error system identification with shifted asymmetric laplace distribution noise." *Journal of Computational and Nonlinear Dynamics,* **21**(5), 051008.