OPEN ACCESS JOURNALS
Indiana Journal of Multidisciplinary Research
Abbreviated Tittle- Ind J Multi Res
ISSN *(Online)*- 2583-3820
Journal Homepage Link- https://indianapublications.com/Journals/IJMR
DOI: https://doi.org/10.5281/zenodo.21623415

INDIANA PUBLICATIONS
PRODUCTIVE AND QUALITY RESEARCH

Research Articles

Volume-06|Issue-04|2026

Personalized Destination Tracking and Alerting Using QR Ticketing and GPS for Public Transport

Vighnesha[1], Sachin Ananthakrishna[2], Terence Johnson[3]

[1,2]UG Students, Mangalore Institute of Technology & Engineering, Moodabidre, Mangalore, Karnataka, India.
[3]Professor, Mangalore Institute of Technology & Engineering, Moodabidre, Mangalore, Karnataka, India.

Article History
Received: 05.05.2026
Accepted: 22.06.2026
Published: 25.07.2026

Citation
Vighnesha, Ananthakrishna, S. & Johnson, T. (2026) Personalized Destination Tracking and Alerting Using QR Ticketing and GPS for Public Transport. *Indiana Journal of Multidisciplinary Research, 6*(4), 488-493.

Abstract: This work introduces a useful app made to assist people commuting by public transportation in avoiding missing stops because of inattention or drowsiness. The technology extracts destination coordinates from tickets using QR codes. Following a scan, the app initiates real-time GPS tracking and shows the path along with the traveler's present location. The software delivers repeated alerts every 30 seconds when the bus is within one kilometer of the destination. The system notifies the conductor and stops alarms if the passenger reacts. Otherwise, a missed stop notification is sent to the conductor. Additionally, the app offers minimal battery usage background operation and digital tickets. It safeguards user privacy, decreases human inspection, and increases trip safety.

Keywords: Smart Public Transportation, Conductor Notification, Digital Ticketing, GPS Tracking, Passenger Alert System, Missed Stop Detection.

INTRODUCTION

Commuters using public transportation, especially on long trips, often find it difficult to navigate to their destination as they miss stops due to disorientation by being distracted or due to sleep. There are currently no stop-based, personalized alert systems for personalized destination tracking and alerting. Additionally, because conductors and drivers lack an efficient way to determine which passengers have not yet left, they usually call out to check on passengers. To address this problem, we propose an application that combines automatic alarm systems, real-time GPS tracking, and QR code-based ticketing. The application instantly alerts passengers before their stop and alerts the conductor if a passenger does not respond. In existing systems, users have to manually set their destination to get alerts; however, the proposed system in this work automatically fetches the destination information and auto-sets the notification alarm when they scan the generated ticket. The novelty of this work lies in the system uniquely integrating QR-based ticketing with dynamic extraction of destination coordinates and a real-time GPS engine that not only tracks the journey but actively prevents passengers from missing their stops. Another novelty is that the approach includes conductor notifications and follows up with acknowledgement or missed stop alerts to the conductor. The scanning of tickets, which not only confirms payment but also sets the exact route for the passengers, based on the information encoded in their unique ticket, also adds to the novelty.

LITERATURE REVIEW

Gangurde et al. created and safely scanned tickets using AES encryption in their QR-based ticketing system. It ensures data security and backend dependability through the use of Apache Kafka [1]. Sonar et al. proposed a sophisticated QR authentication method using Firebase for secure, real-time ticket verification and backend management [2]. Pawar et al. suggested BookMyBusRide (BMBR), a QR-based ticketing system that features dynamic fee calculation, automatic pass management, and Aadhaar identification [3]. A QR-based bus ticketing system with wallet-based digital payments and SMS confirmations was created by Reddy et al. It makes fare transactions more efficient and gives transportation companies the ability to keep an eye on collections [4]. For speedy bus services, Keerthana and Annapriya suggested a smart ticketing app based on QR codes. By using driver-side scanning to track seat occupancy and ticket verification, the system enhances boarding [5]. A ticketing system based on QR codes that ensures boarding and exit validation was introduced by Ashok et al. It issues stop alerts and enforces distance-based pricing [6]. For route analysis and stop optimization, Vidanapathirana et al. suggested a GPS-based data gathering system that makes use of a portable device and a mobile app [7]. A GPS-based bus tracking system was created by Singla and Bhatia that uses past speed data to enhance ETA forecasts and sends out updates via SMS, displays, and applications [8]. Babu et al. suggested a GPS-based tracking system that provides real-time bus whereabouts using MQTT and Google

***Corresponding Author:** Vighnesha

Maps API [9]. A PSDM system with GPS and GSM for destination notifications, emergency alarms, and real-time tracking was presented by Jagannadham et al. [10]. A passenger alert app featuring a peer conversation module and a smart alarm to wake users close to their stop was created by Amarnath et al. [11]. According to Santhasheela et al., Android Location Alarm is an app that provides customized notifications, map distance view, and destination-based alerts [12]. SmartCommute, an AI-powered travel assistant with GPS tracking, geofencing, and traffic-aware routing, is presented by Karthikeyan et al. It offers emergency notifications as well as destination alerts for specific users [13]. To address the evening travel problem, Jagadale and Hegde suggest an SMS-based alert system to inform train passengers of impending stops [14]. In order to help illiterate travelers, Murugaiyan and Soundarya suggest an embedded GPS-based alert system that uses speakers and an LCD to warn passengers of impending stops [15]. Most works are devoid of crucial real-time commuter safety elements. Conductor-side missed stop alerts, passenger sleep detection, live GPS tracking, QR-based check-ins, customized alert systems, and end-to-end coordination between conductors and passengers are unavailable features. None offer automated alerts based on user responsiveness or combine destination-based alarms with real-time tracking. Existing methods do not guarantee personalized commuter involvement or active conductor participation, in contrast to our proposed system. Our method is more suited for realistic implementation in contemporary public transportation scenarios because it combines automation, GPS monitoring, and QR integration in a unique way.

PROPOSED SYSTEM

A mobile-based application to help conductors and passengers on public buses is developed with destination tracking and alerting. The two main components of this system are the Passenger Application and the Conductor Application. Together, these two modules use GPS tracking, QR code scanning, and alerting systems to make sure that travelers, even those who are sleepy or preoccupied, don't miss their target stops.

Passenger App:

Users can make a profile on the passenger app by submitting their name, age, and profile photo. The traveler scans a QR code on their ticket, which has the destination coordinates, when they board the **bus**. An “Enter” button appears after scanning. This button, when clicked, does the following:

- Shows the ticket's digital form.
- Opens a real-time map with the passenger's destination indicated, the current location, and the entire bus route (determined by the conductor).
- Starts tracking GPS continuously in the background.

The software repeats the alerts every 30 seconds as the bus gets closer to the passenger's destination within 1 kilometer. There is an "OK" button on every notification. The passenger's alert status is removed and they are shown as awake if they tap the button.

The software invokes repeated alerts if passengers do not hit "OK." The technology looks for passengers who are not responding when the bus nears its location and notifies the conductor about it.

Conductor App:

The conductor application establishes and oversees the bus route using stop locations, scanning passengers' QR codes to register them and record their destination in the system, keeping track of which passengers must get off at each stop. The conductor's app saves and displays the passengers’ destination when they scan their QR code. A passenger is immediately taken off the "to-exit" list if they hit "OK" on the alert from their own app. If not, a missed passenger alert is sent to the conductor, enabling them to take appropriate action, such as waking the passenger, when the bus reaches that stop. The app’s real-time coordination increases safety, lessens the conductor's physical labor, and guarantees passengers with a smooth travel experience.

SYSTEM ARCHITECTURE & FLOW DIAGRAM

The proposed system consists of two main components:

- Passenger Application Module
- Conductor Application Module

These two modules can connect with one another through cloud-based syncing and QR code exchange. The whole architecture guarantees synchronized journey tracking and destination alerts using GPS, QR code information, and real-time alerts.

Block Diagram of the System:

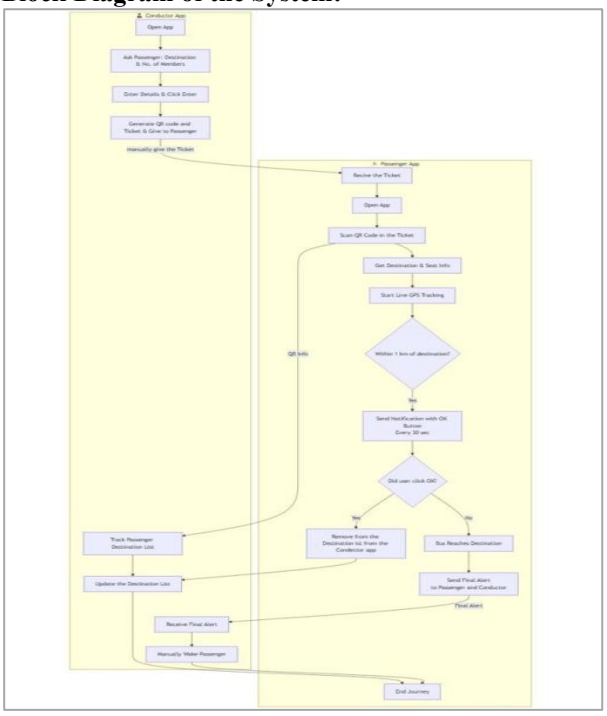

Fig 1: Flowchart of how two apps communicate

***Corresponding Author:** Vighnesha

1. Conductor App for QR code generation
The conductor app requires embedded parameters for generation of QR logic, which is detailed in the steps below.

a) Embedded Parameters:
1. Origin Coordinates: fromStop.latitude, fromStop.longitude
2. Destination Coordinates: toStop.latitude, toStop.longitude
3. Travel Mode: Fixed as "driving"
4. API Version: OSRM and OpenStreetMap for routing

b) Generation Logic:
Algorithm: QR Generation
1. Conductor selects FROM and TO stops.
2. System retrieves GPS coordinates for both stops.
3. Constructs Google Maps URL with origin/destination (QR format only).
4. Generates QR code containing the complete URL.
5. QR displayed on screen + printed on physical ticket.

2. Passenger Alert System with Threshold Configuration & Monitoring
The algorithm for Passenger Alert System and Threshold Configuration is detailed in the steps below. When passenger scans QR code, the following process is initiated to set threshold.

a) Threshold Configuration Logic
Algorithm: setPassengerAlert(qrData)

begin
1. Parse QR to extract destination coordinates
   destination = parseQRCoordinates(qrData)
2. Set fixed thresholds for this passenger
   PRIMARY_THRESHOLD = 1000 // meters (main alert)
   SECONDARY_THRESHOLD = 500 // meters (future use)
3. Start background monitoring service
   ForegroundService.start()
4. Initialize state tracking

```
alertState = {
  notified1km: false,
  startedReminders: false,
  notified500m: false
}
end
```

b) Monitoring Logic:
Algorithm: backgroundMonitoring()

Every 5 seconds do:
currentLocation = GPS.getCurrentPosition()
distance = calculateDistance(currentLocation, passengerDestination)
IF distance <= 1000m AND NOT alertState.notified1km THEN
triggerImmediateAlert()
startRepeatingReminders(10 seconds interval)
alertState.notified1km = true
end IF
end

3. Passenger Notification System
Once the passengers reach their destination threshold points, the following process details the way in which they are notified.

Notification Delivery Logic:
Algorithm: passengerNotificationFlow()

Begin
/Stage 1: Immediate Alert (when crossing 1000m)/
showOneTimeNotification({
title: " Bus Alert",
body: "You're near your stop!",
channel: "bus_reminder_channel",
buttons: ["OK"]
})

/Stage 2: Repeating Reminders (every 10 seconds)/
DELAY(1 second)
Timer.periodic(10 seconds, () => {
showRepeatingNotification({
title: " Reminder!",
body: "You're near your stop!",
buttons: ["OK"]
})
})
End

4. Conductor Notification for Non-Alighting Passengers
This section describes the logic for notifying the conductor when the passengers fail to alight after having received the notification of their destination arrival.

Detection Logic:
The conductor app is not programmed for notification when passengers don't alight, but is initiated into action when they don't initiate the journey end by clicking OK after having received the notification.

Algorithm: conductorNotification()
Every 10 seconds DO:
busLocation = GPS.getCurrentPosition()
activePassengers = Database.getActivePassengers()

FOR each stop do:
distance = calculateDistance(busLocation, stop)
passengersAtStop = activePassengers.filter(p => p.destination == stop)
IF distance < 1000m AND passengersAtStop.length > 0 THEN
showConductorAlert({

***Corresponding Author:** Vighnesha

title: "Passenger Stop Reached",
body: "You have X passengers to get down at 'StopName'"
})
break
end IF
end FOR
End

Backend Data Flow:

- When the passenger scans QR code, their information is added to the database 'passengers' collection.
- Conductor app monitors destinations and shows notifications based on the data in the database.
- Once the passengers end the journey, their information is removed from the database.
- The conductor app resets and removes the passenger information upon initiation of journey completion by passengers.

METHODOLOGY

Conductor App – Home Page:

During the bus ride, the Conductor App's main page serves as the main dashboard for controlling and keeping an eye on all ticketing and passenger-related activities. The current position is first retrieved when the conductor launches the app in order to sync with the real-time bus tracking system.

The following modules are part of the home page:

1. **Generate Ticket** – With this option, the conductor can gather passenger input and then create and print tickets. Information is entered such as the number of members, the source (From), and the destination (To). The traveler scans the created and printed QR code on their own app to track their progress as shown in Fig.
2. **Printer Settings** – The conductor has the ability to control printer connections via Bluetooth or Wi-Fi. The screen makes it simple to reconnect and shows the printer's state (connected or disconnected).
3. **Passenger Status** – A list of passengers who have scanned their QR codes is shown via the Passenger Status feature. It detects if any passengers have missed their stop and displays each passenger's destination.
4. **Add Stop** – Add Stop gives the conductor the ability to establish or alter the bus route and add or edit bus stops. Although it can be changed as needed, this is originally set up after app installation as shown in Fig. 2.
5. **Live Map View** – The bus's current location, the designated route, and the passenger stops are displayed on an interactive map. This keeps the conductor informed about impending stops and people who are currently on board.

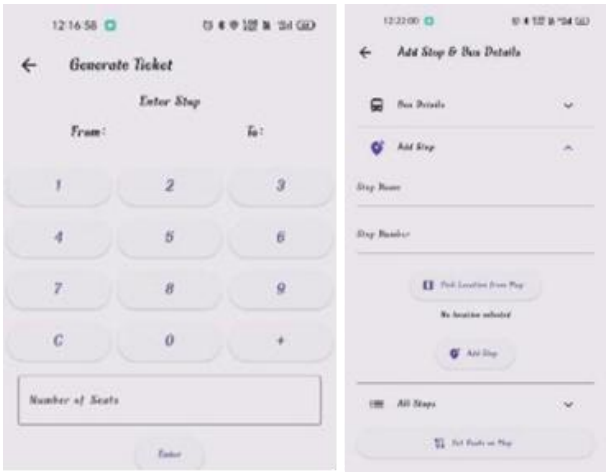

Fig. 2. Ticket Generation Interface in Conductor App & Add Stop and Route Configuration Interface

Passenger App – Home Page:

The passenger app is intended to help travellers plan their trip and maintain alertness prior to arriving at their destination. When the program is first installed, it asks the user to create a profile by entering their name, age, and profile image. These can be changed at a later time.

The program asks for location permission when it first launches in order to activate live GPS tracking capabilities. The following modules are part of the Home Page:

- **Scan Ticket & View Route** - This feature enables the traveller to view the bus route on a map by scanning the QR code on the ticket they received from the conductor.
- **Passenger Info** - Users have the option to view or modify their name, age, and profile picture.
- **Map Display** - Improves navigation awareness by displaying the traveller's real-time location on the map.

IMPLEMENTATION AND RESULTS

We used a real-time GPS and Bluetooth-enabled ticket printer to test our system on Android handsets. While the passenger app was utilized on different devices, the conductor app was loaded on a device connected to the printer. Field testing was conducted on a public transport bus plying from Moodabidri to Kaikamba in Karnataka, India. Specifically, the system was tested from Moodabidri bus stand to a destination 4.7 kilometers away in Thodar Dwara (popularly known as Yenepoya stop) to undertake a realistic simulation. The conductor of the bus was given a mobile device at Moodabidri bus stand that had a Bluetooth printer and the conductor app installed, as shown in Fig. 4. The testing confirmed essential features like the ability to generate tickets, lock destinations using QR codes, and identify missing stops for both conductors and passengers. Conductor opens the Auto Alert application, displaying the app's homepage on the mobile device as shown in Fig. 5. Conductor asks the

***Corresponding Author:** Vighnesha

passenger for journey details (from: Moodabidri, to: Yenepoya) and inputs the corresponding stop numbers and number of seats depicted in Fig. 5. After clicking the enter button, the ticket is generated with details like bus name, bus number, from & to destinations, fare, date, time, and QR code visible in Fig. 6. The ticket is printed using a Bluetooth-connected thermal printer displayed in Fig. 6. Printed ticket is handed over to the passenger captured in Fig. 7. Passenger opens the application and views the initial home interface as seen in Fig. 7. Passenger scans the QR code on the printed ticket. Upon successful scan, destination latitude and longitude are displayed along with an "Enter" button highlighted in Fig. 8.

Fig 4: Bluetooth Portable Printer

Fig 5: Conductor App Home Screen & Entering Ticket Information

After clicking "Enter", the map view updates with bus route, current location (blue icon), and destination (red flag icon) visible in Fig. 8. Conductor sees a list of passengers along with their destination. Destinations are marked green (no drop-off) or red (at least one drop-off), included in Fig. 9.

Fig 6: Ticket Generation & Ticket Printing

Fig 7: Ticket Issued & Passenger App Home Screen

Fig 8: QR Code Scanning & Map with Destination

Passenger receives an alert notification when the destination (Yenepoya) is less than 1 km away, portrayed in Fig. 10. Since the passenger doesn't click "OK" in any alert, the conductor receives a warning to manually wake the passenger, as illustrated in Fig. 10. Passenger clicks "End Journey" after successfully reaching the destination. Passengers on public transportation, particularly those that often miss their stops because of sleep or inattention, will find that this application greatly improves their travel experience. The technology provides a number of benefits over conventional systems and earlier studies by combining real-time GPS tracking, automated alert systems, and QR code-based ticketing.

***Corresponding Author:** Vighnesha

Fig 9: Conductor's Passenger List View

Fig 10: Passenger Alert Notification & Conductor Missed Stop Alert

CONCLUSION

This application enables conductors to swiftly generate digital tickets using established stop numbers, in contrast to outdated techniques that depend on manual inputs or printed timetables. The app continuously monitors the traveler's location when they scan their ticket. To lessen the need for conductor announcements, passengers receive periodic warnings when the bus is within one kilometer of their destination. The conductor receives a final alert containing the passenger's information in the event that the passenger does not react to alerts, allowing for manual intervention. Even under the worst sleep situations, this ensures no missed stops. Our system includes the conductor as a crucial participant, enabling two-way coordination and real-time status updates, in contrast to other solutions that just concentrate on passenger-side warnings.

REFERENCES

1. Gangurde, N., Ghosh, S., Giri, A., & Gharat, S. (2022). Ticketing system using AES encryption based QR code. *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 201–206.
2. Sonar, L., Gite, T., Taralkar, T., Bendale, P., Bansode, S., & Salunke, D. (2024). Secure and efficient public transport management through smart QR code authentication. *2024 5th International Conference on Image Processing and Capsule Networks (ICIPCN)*, 173–177.
3. Waghmare, P. B., Kondekar, A. P., Shinde, P. S., & Chimgave, S. B. (2024). Review paper on BookMyBusRide (BMBR): QR-driven intelligent bus ticketing system. *IOSR Journal of Engineering, 14*(12), 37–51.
4. Reddy, C. U., Reddy, D. L. S. V. P., Srinivasan, N., & Mayan, J. A. (2019). Bus ticket system for public transport using QR code. *IOP Conference Series: Materials Science and Engineering, 590*, Article 012036.
5. R., K., & Annapriya, N. (2022). *Smart bus ticket system using QR code* (Bachelor of Science degree project report). School of Computing, Sathyabama Institute of Science and Technology.
6. Ashok, K., Sanketh, B. M., Shashank, M. A., Tanush, K. N., & Varun, R. (2024). QR scan ride: Smart bus passenger ticketing system. *International Journal of Progressive Research in Engineering Management and Science, 4*(5), 903–909.
7. Vidanapathirana, C. J., Thilakshan, T., Sharic, A. H. S., & Bandara, S. J. (2024). A GPS-based solution for real-time data collection of public transport systems. *International Journal of Research in Engineering Science, 12*(11), 97–100.
8. Singla, L., & Bhatia, P. (2015). GPS based bus tracking system. *2015 International Conference on Computer, Communication and Control (IC4)*, 1–6.
9. Babu, J., Kuruvilla, K., Mani, V. A. G., Sethulakshmi, K. R., & Priyanka, K. S. M. (2021). GPS based bus tracking system. *International Journal of Engineering Research and Technology (IJERT), 9*(13).
10. Jagannadham, D. B. V., Sai Narayana, D. V., Koteswar, D., & Tripathi, S. (2019). Passenger safety monitoring and destination alert system for public transports. In G. Panda (Ed.), *Microelectronics, Electromagnetics and Telecommunications* (Lecture Notes in Electrical Engineering, Vol. 521, pp. 766–777). Springer.
11. Amarnath, J., Raizudeen, S., & Kumar, K. (2023). Passenger alert application. *International Journal for Research in Applied Science and Engineering Technology, 11*(5), 4893–4896.
12. Santhasheela, K., Haritha, C., Sri Devi, K. A., & Pooja, M. (2021). A dynamic alarm for passengers. *International Journal of Advanced Research in Computer and Communication Engineering, 10*(7), 1–6.
13. Karthikeyan, C., Chinnamani, S., Suriya Prakash, H., Arun, R., & Vigneshwaran, R. (2025). Destination alert wakeup–alarm with travel management. *International Journal of Research Trends and Innovation, 10*(4), 45–55.
14. Jagadale, B. N., & Hegde, K. (2013). SMS based alerting system for train passengers. *Journal of Pure and Applied Industrial Physics, 3*(1), 1–67.
15. Murugaiyan, S., & Soundarya, S. (2021). Smart system for preventing passenger destination missing in bus. *Journal of Physics: Conference Series, 1717*, Article 012016.

***Corresponding Author:** Vighnesha