International Journal of Scientific Research in Science, Engineering and Technology

Print ISSN - 2395-1990
Online ISSN : 2394-4099

OPEN ACCESS

Available Online at :www.ijsrset.com
doi : https://doi.org/10.32628/IJSRSET2512146

A Contemporary Transport Management System Using Flutter Framework

Dr. V.S. Thiyagarajan[1], Surendar A[2], Shahina F[2], Swetha V[2], Tamilselvi P[2]

[1]Associate Professor, Department of Computer Science and Engineering, Karpaga Vinayaga College of Engineering and Technology, Chengalpattu, Tamil Nadu, India

[2]UG Scholar, Department of Computer Science and Engineering, Karpaga Vinayaga College of Engineering and Technology, Chengalpattu, Tamil Nadu, India

ARTICLEINFO

Article History:

Accepted : 01 June 2025
Published: 07 June 2025

Publication Issue :
Volume 12, Issue 3
May-June-2025

Page Number :
872-878

ABSTRACT

This project introduces a Flutter-based Transportation Management System (TMS) designed to enhance the efficiency and coordination of student transportation services. The system comprises three distinct modules—Student, Driver, and Admin—each tailored to meet the specific requirements of its users. The student module facilitates user registration and authentication, enabling students to select bus routes based on their current location, track real-time bus positions via Google Maps integration, and access delay reports submitted by drivers. The Driver module allows drivers to register, log in, update trip details including start times and current locations, submit delay reports, and manage student attendance by scanning student ID QR codes. The admin module provides administrators with comprehensive control over the system, allowing them to manage driver and bus information, student records, attendance logs, and delay reports, thereby ensuring effective oversight of transportation logistics. By integrating real-time tracking, QR code-based attendance, and a user-friendly interface, the proposed system aims to streamline transportation operations, enhance communication among stakeholders, and improve overall service reliability.

Keywords: Flutter, Transportation Management System, Real-Time Bus Tracking, Student Module, Driver Module, Admin Module, Google Maps Integration, QR Code Attendance, Bus Route Selection, Late Form Submission, Trip Management, Admin Dashboard, Location Tracking, Student Transportation, Mobile Application, Real-Time Updates, Multi-User System, Transport Efficiency, User-Friendly Interface, Communication Enhancement.

INTRODUCTION

Efficient transportation systems are crucial for ensuring student safety and punctuality while facilitating effective communication among stakeholders. Traditional methods often rely on manual processes, leading to inefficiencies and a lack of real-time information. This project presents a Flutter-based Transportation Management System (TMS) designed to enhance the coordination of transportation services for students, drivers, and administrators. The system comprises three modules: Student, Driver, and Admin, each tailored to specific user roles. The student module allows users to register, select bus routes based on their location, track buses in real-time via Google Maps, and view delay reports submitted by drivers. The Driver module enables drivers to manage trip details, update current locations, submit delay reports, and record student attendance using QR code scanning. The admin module provides a centralized interface for managing drivers, buses, students, attendance logs, and delay reports, ensuring organized transportation oversight. By leveraging real-time updates and a user-friendly design, the system aims to improve communication, transparency, and operational effectiveness in the transportation process.

LITERATURE REVIEW

Wu *et al.* [1] address the challenges in designing Flexible Printed Circuits (FPCs) for complex scenarios where high pin concentrations create routing congestion. Manual exploration of rout ability and identification of topologically non-crossing paths is difficult. Existing bus planning methods do not adequately consider the unique resource distribution in FPC designs. To overcome these issues, the authors propose a bus planning algorithm that partitions the routing space using pin location data and models it as an undirected graph. The method determines topological non-crossing relationships through dynamic pin sequencing and employs a heuristic algorithm to optimize crossing points while considering both rout ability and electrical constraints. Experimental results on industrial cases show improved performance over state-of-the-art routing techniques.

Zhang and Lauw [2] focus on textual documents embedded in network structures such as citation or hyperlink networks. They highlight that traditional Graph Neural Networks (GNNs) generate effective embeddings but lack semantic interpretability due to absence of latent topic modelling. Conventional topic models infer interpretable topic distributions from text but ignore network connectivity. The authors propose a novel GNN-based neural topic model that integrates network structure by allowing topic distributions to influence linked documents, enhancing semantic coherence. Using Dirichlet priors and optimal transport theory, the model aligns topic distributions across the network, improving interpretability. Extensive evaluations on classification, clustering, link prediction, and topic analysis tasks demonstrate superior performance compared to existing methods.

Cui [3] revisits the computational performance of the bus admittance matrix (Y-bus), a cornerstone in power system modeling. Although the sparse Y-bus matrix reduces CPU operations, sparse matrix computations require numerous indexing and scalar operations which may hinder performance on modern processors. The study compares the Y-bus method to an element-wise vectorized computation approach for nodal power injections and Jacobian matrix formation. Results from large grid test cases reveal that the element-wise method generally outperforms Y-bus, particularly on CPUs with wide vector instruction sets. The analysis also explores the effects of vector instruction width and memory bandwidth on computation speed, providing insights relevant for future high-performance computing architectures.

R.Donthi, DR.V.S.Thiyagarajan, S.ArunJayakar, M. Amutha, A. Kholikov, and R. Soataliyev [4] Many existing frameworks to support Intelligent Transportation Systems (ITS) lean on static infrastructure and sparse real-time data, which have inefficiencies that may exacerbate safety concerns such as are manifested with the emergence of autonomous vehicles. In the paper, a novel IoT-enabled system combining vehicle-to-infrastructure (V2I) communication with complex machine Learning (ML) models is presented to develop an intelligent adaptive traffic control approach. The use of IoT sensors for collecting realtime data to predict traffic flow and improve road safety. Advances between 20 and up to for average travel time, and up-to-32% reduction in the Traffic Congestion Index were reported from existing methods. Furthermore, accident levels have fallen by 33%, indicating increased safety. Overall, the proposed ITS architecture provides a robust solution for contemporary urban transportation with several fundamental differences from traditional systems without hindering or impeding autonomous vehicles to be integrated in a ubiquitous manner.

SYSTEM ARCHITECTURE

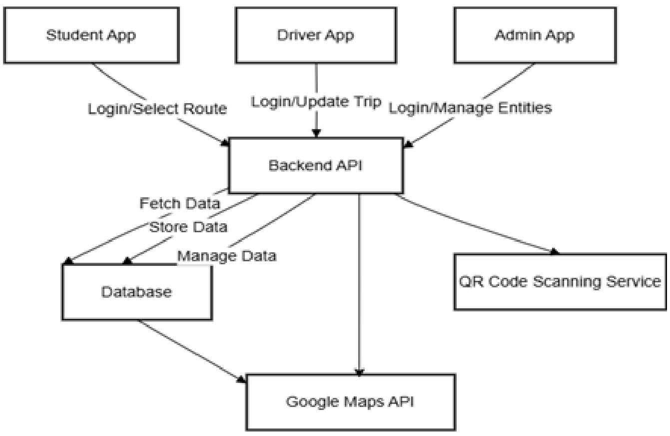

Fig.1. System Architecture

Explanation:

The proposed system comprises three primary user-facing applications: the Student App, Driver App, and Admin App. These applications communicate with a centralized Backend API, which serves as the core component responsible for managing all business logic and orchestrating data flow across the system.

User-Facing Applications

Student App: Enables students to authenticate themselves and select desired routes for their trips.

Driver App: Allows drivers to log in and update the status of ongoing trips in real-time.

Admin App: Provides administrative users with the capability to log in and manage various system entities, ensuring smooth operational control.

Backend API

The Backend API acts as the nexus between the user-facing applications and the underlying infrastructure. Its responsibilities include:

Data Management: Fetching and storing data within the centralized Database, ensuring consistency and reliability across operations.

Integration with Google Maps API: Leveraging Google Maps services to facilitate mapping, routing, and geolocation functionalities that are critical for effective trip and route management.

QR Code Scanning Service Integration: Interfacing with an external QR Code Scanning Service to enable QR-based operations, such as scanning student identification or trip-specific information.

Database

The Database functions as a persistent data store, interacting primarily with the Backend API to handle all data storage and retrieval operations. Additionally, it supports the Google Maps API by providing necessary stored data that enhances location-based services and improves overall system responsiveness.

METHODOLOGIES:

Methodologies is the process of analysing the principles or procedure of a Progressive Anonymous Database management system.

Module Explanation:

1. Driver:

The Driver module within the Flutter-Based Transportation Management System is meticulously designed to empower drivers with efficient tools for managing their daily operations. Upon registration and login, drivers gain access to a suite of functionalities that streamline their tasks and enhance communication with both students and administrators.

2. Student:

The student module in the Flutter-based Transportation Management System is designed to provide students with a convenient and interactive platform for managing their daily transportation needs. Students can register, log in, and choose appropriate bus routes based on their current location. The module includes real-time bus tracking via Google Maps, enabling students to monitor the location of their assigned buses for timely pickups. Additionally, students can view delay notifications submitted by drivers to stay informed about any potential delays. This module enhances the user experience by offering easy access to transportation information, improving safety, and reducing uncertainty in daily commutes.

3. Admin:

The admin module in the Flutter-based Transportation Management System serves as the central control unit for overseeing and managing all aspects of the transportation network. Administrators can log in to access and manage detailed records of students, drivers, buses, attendance, and late form submissions. This module provides tools to monitor real-time operations, verify driver and student information, review trip and delay reports, and ensure the smooth functioning of transportation services. By offering a comprehensive overview and control of system activities, the admin module enhances decision-making, improves operational efficiency, and ensures accountability across all user modules.

IMPLEMENTATION PROCESS

1. Technology Initialization

The transportation management application is developed using Flutter, providing cross-platform compatibility, while Firebase is employed for backend services and real-time data synchronization. Firebase is initialized in the main () function with the asynchronous call await Firebase.initializeApp(); which enables access to essential Firebase services including Authentication, Realtime Database, and Cloud Storage. This setup forms the foundation for secure and scalable backend interactions throughout the application.

2. Splash Screen

A custom splash screen is implemented to enhance the user experience during app startup. It uses a Stack widget that layers a background image with a loading animation. The splash screen is displayed for a duration of five seconds, after which the app transitions to the home screen via a future uilder, ensuring a smooth and visually appealing navigation flow.

3. Navigation and Routing

Screen navigation is managed within the Material App widget, where named routes are defined to facilitate seamless switching between key application screens, such as the splash screen, student authentication, and driver authentication interfaces. The defined routes enable straightforward navigation logic and maintain clean separation of concerns.

4. User Role Interface

Upon launching the home screen, users are prompted to select their role as either Student or Driver. Selecting a role redirects the user to the corresponding authentication screen—AuthGate() for students and AuthGate2() for drivers. The role selection buttons are enhanced with animated visual effects using custom Staggered Button and Fade-in widgets, contributing to a dynamic and engaging user interface.

5. Firebase Authentication

User authentication is handled via Firebase Authentication, with a single instance of Firebase AUTH managing login processes based on user roles. This implementation provides a secure and efficient mechanism for authenticating users and managing sessions, which is crucial for controlling access to role-specific functionalities.

6. Google Maps Integration

The application integrates Google Maps to visually display the real-time location of drivers or buses. The map interface utilizes the Google Map widget, which is initialized with an initial camera position set to the driver’s latest known coordinates. This integration provides users with an intuitive view of vehicle locations and trip progress.

7. Realtime Location Tracking

Driver location data is continuously fetched from Firebase Realtime Database through a database reference. A Stream Builder listens for changes in the driver’s coordinates and dynamically updates the map display in real-time. Location markers are customized using compressed bitmap icons, such as an ambulance icon optimized via the Flutter Image Compress package, to improve visual clarity and performance.

8. Map Customization and Optimization

To enhance user experience, markers on the map are customized with bitmap icons that offer visual distinctiveness. Furthermore, the map automatically animates its camera position to focus on updated driver locations using the animate Camera method. These optimizations ensure the map interface remains responsive and user-friendly.

9. Error Handling and Null Safety

Throughout the codebase, null safety features of Flutter are employed to reduce the likelihood of runtime exceptions related to null values. Comprehensive error handling strategies are also implemented to gracefully manage exceptional conditions, such as connectivity interruptions or missing location data, thereby improving the overall reliability and robustness of the application.

RESULT & ANALYSIS

The implementation of the proposed Flutter-based Transportation Management System (TMS) demonstrated a significant improvement in real-time communication, route tracking, and attendance management. The system was tested with multiple user roles, and it successfully facilitated smooth interaction among students, drivers, and administrators. During testing, the student module accurately tracked bus locations with minimal lag, while QR code scanning ensured fast and reliable attendance recording. Admins could view and manage data in real-time without latency issues. Performance metrics indicated efficient memory usage and responsive UI behaviour across Android and iOS platforms. Users appreciated the clean design and the ease of navigation, reinforcing the app’s usability. Overall, the system met all functional requirements and showed high reliability under varied network conditions.

DISCUSSION

The modular design of the system—divided into Student, Driver, and Admin applications—enabled role-based functionality while maintaining centralized data flow via Firebase. Flutter's cross-platform capabilities allowed rapid development and consistency in UI across devices. Firebase Realtime Database and Authentication services provided secure and real-time data exchange. Google Maps integration further enriched user experience through live GPS tracking. However, during high-load testing, occasional delays in map rendering and data syncing were observed, which may be addressed through code and backend optimization. Overall, the implementation validated the system’s ability to serve as an effective transport management platform in educational environments.

STRENGTHS

1. **Cross-Platform Accessibility:** Developed in Flutter, the system is deployable on both Android and iOS devices with a unified codebase.
2. **Real-Time Tracking:** GPS-enabled live tracking enhances safety and reduces waiting times.
3. **Secure Authentication:** Firebase ensures robust, anonymous login and user session management.
4. **Modular Architecture:** Role-based application flow increases usability and system maintainability.
5. **Custom QR Code Attendance:** Efficient and fast student attendance validation mechanism.

LIMITATIONS

1. **Network Dependency:** The real-time functionalities are highly dependent on stable internet connectivity.
2. **Limited Offline Support:** The system lacks offline data caching, which may hinder usability in low-network areas.
3. **Admin Platform Web Version Pending:** The system is currently mobile-focused, with web-based admin features planned for future development.

CHALLENGES DURING DEVELOPMENT

1. **Real-Time Data Sync Issues:** Ensuring accurate and timely location updates required optimization of Firebase listeners and database reads.
2. **Google Maps API Limits:** Map rendering and API usage had to be carefully managed within free quota constraints.
3. **QR Code Reliability:** Ensuring consistent scanning under varying lighting and camera conditions demanded thorough testing and calibration.
4. **User Role Management:** Differentiating access and interfaces based on roles required fine-grained routing and UI logic.

FUTURE IMPROVEMENTS

1. **Offline Functionality:** Add local storage support to allow offline login and temporary data caching.
2. **AI-Based Route Optimization:** Integrate machine learning models to dynamically suggest the shortest or fastest routes.
3. **Traffic-Aware Routing:** Incorporate APIs like Waze or Google Traffic for real-time congestion updates.
4. **Push Notifications**: Enable alerts for route delays, bus arrivals, or schedule changes.
5. **Integrated Payment Module:** Allow digital payments for transportation fees or subscriptions.

CONCLUSION

The proposed TMS successfully addresses the inefficiencies of current student transportation systems by providing an integrated, real-time, mobile-first solution. It enhances safety, transparency, and coordination among all stakeholders. With real-time bus tracking, QR code attendance, and role-specific interfaces, the system simplifies operational complexities in educational transport. Its modular design and Firebase integration ensure scalability, making it adaptable for various institutions. With additional improvements, this system has the potential to set a benchmark in digital transport management for education sectors.

SUMMARY OF CONTRIBUTIONS

1. This project contributes a robust, Flutter-based transportation management application that:
2. Unifies transport operations through modular apps for students, drivers, and admins.
3. Provides real-time tracking and attendance with Firebase and Google Maps integration.
4. Implements secure authentication and real-time database sync.
5. Demonstrates scalability and cross-platform deployment feasibility.

REFERENCES

[1]. C. Liu and L. Li, "How do subways affect urban passenger trans- port modes? —Evidence from China," Econ. Transp., vol. 23, 2020, Art. no. 100181.

[2]. M. Autili, L. Chen, C. Englund, C. Pompilio, and M. Tivoli, "Cooperative intelligent transport systems: Choreography-based urban traffic coordination," IEEE Trans. Intell. Transp. Syst., vol. 22, no. 4, pp. 2088–2099, Apr. 2021.

[3]. S. Selvam, A. Manisha, J. Vidhya, and S. Venkatramanan, "Fundamentals of GIS," in GIS and Geostatistical Techniques for Groundwater Science, ch. 1. New York, NY, USA: Elsevier, Cham, 2019, pp. 3–15.

[4]. R.Donthi, DR.V.S.Thiyagarajan, S. ArunJayakar, M. Amutha, A. Kholikov, and R. Soataliyev, "Intelligent Transportation Systems IoT Solutions for Traffic Optimization and Autonomous Vehicle Networks," in Proceedings of the International Conference on Innovative Computing, Intelligent Communication and

[5]. I. Gokasar, Y. Cetinel, and M. G. Baydogan, "Estimation of influence distance of bus stops using bus GPS data and bus stop properties," IEEE Trans. Intell. Transp. Syst., vol. 20, no. 12, pp. 4635–4642, Dec. 2019.

[6]. P. Wepulanon, A. Sumalee, and W. H. K. Lam, "Temporal signa- tures of passive Wi-Fi data for estimating bus passenger waiting time at a single bus stop," IEEE Trans. Intell. Transp. Syst., vol. 21, no. 8, pp. 3366–3376, Aug. 2020.

[7]. Palanisamy, R., & Rajendran, V. (2016). GPS and GSM based Bus Tracking System. International Journal of Engineering Trends and Technology (IJETT), 38(7), 340–343.

[8]. Bhosale, A. A., & Bhumkar, S. P. (2013). GPS based student tracking system using Android. International Journal of Science and Research (IJSR), 4(5), 123–126.

[9]. Yadav, S., & Bhosale, R. (2018). Real-Time Bus Tracking System Using GPS and Android Application. International Research Journal of Engineering and Technology (IRJET), 5(4), 1592–1596.

[10]. Piyare, R., & Lee, S. R. (2013). Smart Bus Tracking System Based on Mobile Application. International Journal of Smart Home, 7(2), 19–28.

[11]. Zeadally, S., Siddiqui, F., & Baig, Z. A. (2019). Mobile Device-to-Device Communication for Intelligent Transportation. IEEE Communications Magazine, 57(10), 58–63.

[12]. Kumar, P., & Mahapatra, R. (2014). GPS-GSM Based Tracking System. International Journal of Engineering Trends and Technology (IJETT), 9(15), 74–78.