SRI LANKA INTELLIGENT BUS NAVIGATION AND PASSENGER INFORMATION SYSTEM

24-25J-237

B.Sc. Special (Honors) Degree in Information Technology

Specialization in Information Technology

Department of Information Technology

Sri Lankan Institute of Information Technology

Sri Lanka

July 2024

SRI LANKA INTELLIGENT BUS NAVIGATION AND PASSENGER INFORMATION SYSTEM

24-25J-237

B.Sc. Special (Honors) Degree in Information Technology

Specialization in Information Technology

Department of Information Technology

Sri Lankan Institute of Information Technology

Sri Lanka

July 2024

DECLARATION

I declare that this is my own work and this dissertation1 does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to Sri Lanka Institute of Information Technology, the non exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Name	Student ID	Signature
Senarathne A.S.C.G.	IT21205460	
Kalhara N.V.	IT21202490	
Rajapaksha R.L.H.P.	IT21192050	
Thanusan M.	IT21175848	

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

-- -------------------------

Signature of the Supervisor Date

(Dr. Sanika Wijayasekara)

ABSTRACT

Sri Lanka's public transportation sector, particularly the bus network, faces persistent challenges such as unreliable schedules, overcrowding, inadequate safety measures, and lack of accessible information for locals and tourists. To address these issues, this research proposes a comprehensive Intelligent Bus Navigation and Passenger Information System that integrates IoT, machine learning, blockchain, and NFC technologies. The proposed system enhances public transport by offering real-time bus tracking, accurate Estimated Time of Arrival (ETA) predictions, and seat availability monitoring. It ensures inclusiveness through voice-assisted features for visually impaired passengers and a multilingual chatbot for easy information retrieval. The system also reduces bus competition and associated accidents by informing drivers of nearby bus locations and boarding passengers. On the financial point, a secure and transparent NFC-based cashless fare system is implemented using blockchain technology. This allows instant fare transactions with real-time conductor notifications and supports loyalty programs and passenger categorization for improved service personalization. Furthermore, AI-powered features like unauthorized vehicle detection, license plate recognition, and route recommendations improve both security and user experience. The architecture comprises ESP32-based IoT modules, Django and React Native frameworks, Firebase databases, and machine learning models such as Random Forest for ETA prediction. This multi-faceted integration ensures a reliable, scalable, and user-friendly transport ecosystem. Each subsystem was independently developed, tested, and validated by specialized team members to ensure robust functionality across all components. Ultimately, this research contributes a smart, secure, and inclusive mobility solution that can reshape the future of Sri Lanka's public transport, enhancing safety, efficiency, and passenger satisfaction.

Keywords-Public Transportation, IoT, NFC, Machine Learning, Blockchain

ACKNOWLEDGEMENT

First and foremost, I owe a sincere thank you to my supervisor, Dr. Shanika Wijesekara. Her steady guidance, thoughtful feedback, and constant encouragement made a lasting impact on this work. She helped shape not just the research but also how I approached each challenge that came with it.

I’d also like to thank my co-supervisor, Dr. Kapila Dissanayake. His technical advice and attention to detail helped me navigate several of the more difficult parts of this project. When unexpected hurdles popped up, he was there with practical solutions and the kind of support that made a real difference.

This project crossed into several different fields—real-time IoT systems, embedded tech, cloud services, and transportation systems. I’m deeply grateful to the professionals, both academic and from the industry, who shared their time and expertise to help me through tricky problems and unfamiliar concepts. Their input was essential during the implementation and testing stages.

I’m especially thankful for my peers and team members. The shared ideas, back-and-forth discussions, and support along the way made this journey less daunting. Whether it was a tough technical issue or just a moment of doubt, having this team made the difference between moving forward and getting stuck. Your help was invaluable.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Description
AI	Artificial Intelligence
IoT	Internet of Things
ML	Machine Learning
ETA	Estimated Time of Arrival
NFC	Near Field Communication
GPS	Global Positioning System
TTS	Text-to-Speech
STT	Speech-to-Text
SLTB	Sri Lanka Transport Board
OCR	Optical Character Recognition
API	Application Programming Interface
GSM	Global System for Mobile Communications
LCD	Liquid Crystal Display
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network

1. INTRODUCTION

Public transportation forms the backbone of urban mobility, playing a critical role in ensuring the smooth functioning of social and economic systems. In Sri Lanka, public buses serve as the most accessible and widely used mode of transport, catering to millions of daily commuters across both urban and rural regions. Despite this significance, the Sri Lankan bus transportation network suffers from various structural and operational inefficiencies. Issues such as inconsistent bus schedules, overcrowded buses during peak hours, slow manual fare collection, and the absence of accessible features for differently abled passengers and tourists hinder the quality-of-service delivery. These limitations negatively affect user satisfaction and trust, resulting in increased private vehicle use, traffic congestion, and carbon emissions [1].

The rise of digital technologies and intelligent systems globally has prompted many countries to integrate smart solutions into their transportation infrastructures. Intelligent Transport Systems (ITS) that incorporate GPS-based vehicle tracking, machine learning–based predictive models, mobile applications, and cloud platforms have revolutionized how public transport operates and is experienced. In advanced transport networks such as those in Singapore, South Korea, and Japan, real-time bus monitoring, contactless ticketing, and data-driven route optimization have led to greater efficiency, transparency, and commuter convenience [2], [3]. In contrast, Sri Lanka's public transport sector has lagged in adopting such integrated smart solutions.

This research project introduces a comprehensive, modular smart public bus system tailored specifically for Sri Lanka. The system is designed to mitigate the limitations of the current infrastructure by integrating multiple technological components into a unified, cloud-synchronized platform. These components include real-time GPS tracking using ESP32 microcontrollers, machine learning–based ETA (Estimated Time of Arrival) prediction, an AI-powered chatbot built using OpenAI's GPT model, NFC-based contactless fare payment, blockchain-backed transaction logging for fare integrity, and onboard crowd monitoring via ESP32-CAM modules. Additionally, voice interaction features are incorporated to ensure accessibility for visually impaired users. All modules are interconnected and synchronized using Firebase and Google Cloud services to ensure real-time data delivery and system scalability.

The overarching aim of this research is not merely to digitize transportation processes but to reimagine the commuting experience through intelligent automation and inclusive design. The system supports passengers by providing dynamic route information, real-time bus arrival updates, and cashless, fraud-proof ticketing. For transport authorities, it offers opportunities to analyze usage patterns, manage fleets more efficiently, and improve public trust through transparent data handling. Furthermore, the system is modular and cost-effective, making it suitable for deployment in resource-constrained environments and scalable to nationwide implementations.

By combining Internet of Things (IoT) hardware, cloud computing, artificial intelligence, and blockchain technologies, this project provides a future-ready framework for transforming public transportation in Sri Lanka. The ultimate objective is to deliver a smart, safe, accessible, and efficient transport system that enhances mobility for all user groups—regardless of their location, language, or physical ability.

1.1 Background Literature

Challenges in Sri Lanka's Public Transportation System

Sri Lanka's public transportation system, particularly its bus network, is characterized by a lack of modernization and poor service reliability. The Sri Lanka Transport Board (SLTB) and private bus operators collectively struggle with maintaining punctuality, ensuring passenger comfort, and supporting operational transparency. Passengers frequently encounter delayed or skipped services, especially during peak hours, leading to long wait times and reduced trust in the system [4].

Another major concern is the absence of real-time information systems for public access. Passengers have no visibility into bus locations or expected arrival times. This issue becomes even more critical for tourists unfamiliar with local routes and for differently-abled passengers, particularly the visually impaired, who lack assistive systems to identify buses and stops. Language barriers further complicate the situation for non-native commuters. These challenges create an urgent need for a digitized, inclusive, and predictive system that addresses the needs of all commuters [5].

Integration of IoT and Machine Learning for Real-Time Bus Tracking

Globally, the integration of Internet of Things (IoT) and Machine Learning (ML) technologies has revolutionized public transport systems. IoT enables continuous data collection through sensors and GPS modules, offering real-time tracking of buses. These systems provide updates on vehicle location, speed, and arrival times that can be accessed via mobile apps, bus stop displays, or voice assistants [6].

Machine learning algorithms complement IoT by analyzing real-time and historical traffic data to provide accurate Estimated Time of Arrival (ETA) predictions. These models can incorporate factors such as distance, time of day, road congestion, and weather conditions to generate more dynamic predictions than traditional timetable-based approaches. In the Sri Lankan context, recent research has shown the potential of IoT-enabled buses to support GPS tracking, though existing solutions still lack the sophistication needed for highly accurate, real-time ETA forecasting, especially under unpredictable traffic conditions [7].

By leveraging ML models like linear regression, decision trees, or ensemble methods such as Random Forests, public transport systems can enhance passenger experience through precise arrival updates. This predictive capability reduces wait times, minimizes travel uncertainty, and allows for better commuter planning.

Enhancing Passenger Comfort and Safety

Beyond scheduling and arrival predictions, passenger comfort and safety are critical components of a smart transportation system. Overcrowding remains a persistent issue in Sri Lanka, particularly on high-demand routes. Integrating onboard IoT sensors and lightweight camera modules, such as the ESP32-CAM, enables real-time monitoring of bus occupancy. This data can be used to inform passengers of available seating, distribute loads across multiple buses, and assist in dynamic fleet allocation [8].

Safety features can also be enhanced by embedding driver monitoring systems. Sensors can detect over-speeding, abrupt braking, and reckless acceleration, providing alerts to the driver or control center. These metrics can be used to initiate corrective actions like driver retraining or immediate intervention. Additionally, inter-vehicle distance tracking—implemented using GPS data—can be used to discourage unsafe tailgating practices among competitive bus drivers [9].

Accessibility Features for Visually Impaired Passengers

Traditional public bus systems lack features that accommodate the needs of visually impaired or differently abled passengers. As a result, these commuters face numerous challenges, including the inability to identify the correct bus or determine when to board or alight. With the integration of mobile-based voice assistants and text-to-speech (TTS) functionality, smart transport systems can become more inclusive. These systems allow users to receive route information, estimated arrival times, and stop alerts through auditory feedback, enabling a hands-free and accessible travel experience [10].

Voice-enabled mobile applications using speech-to-text (STT) and TTS technologies provide a two-way communication interface. Passengers can issue voice commands to inquire about bus details and receive spoken responses, making the system usable even by those with limited vision or digital literacy.

Blockchain and NFC for Secure Ticketing Systems

Manual fare collection methods, still prevalent across much of Sri Lanka's bus network, introduce a range of operational challenges, including fraud, fare evasion, and delays due to cash handling. To mitigate these issues, the use of Near Field Communication (NFC) technology combined with blockchain has emerged as a viable solution. NFC smart cards or mobile payments can be used for fast, contactless ticketing, reducing boarding time and improving transaction convenience [11].

Blockchain technology adds a critical layer of security by ensuring that each transaction is logged on a tamper-proof, immutable ledger. This approach guarantees transparency in fare collection, prevents fraudulent ticket manipulation, and enables accurate financial audits. Additionally, loyalty reward systems and category-based fare discounts (e.g., for students or senior citizens) can be embedded into the ticketing system using blockchain smart contracts.

Vehicle Identification and Security Enhancements

Vehicle identification is a frequently overlooked aspect of transport security. In high-traffic regions, misidentifying or missing a bus is a common issue. This is exacerbated by unauthorized vehicles operating on licensed routes. The application of high-resolution Optical Character Recognition (OCR) systems can help automate vehicle number plate detection and verify route authorization in real time [12].

Machine learning algorithms can assist in analyzing vehicle data and flagging unauthorized or hijacked vehicles, which can then be reported to authorities or passengers via alert systems. Integration with early/late arrival notifications and route deviation tracking further enhances reliability and improves passenger trust in the system.

1.2 Research Gap

This research consists of four main components, each of which addresses its own specific research gap. Therefore, the research gaps are presented separately for each component, highlighting their individual relevance.

The main components are:

1. Intelligent Bus Navigation and ETA Prediction System

In Sri Lanka, one of the biggest frustrations for bus commuters is not knowing when the next ride will show up. This part of the system tackles that directly. Each bus was fitted with GPS, and the team trained a simple linear regression model on both live and historical data to predict arrival times. Passengers could check a mobile app to see where their bus was and when it would likely arrive. An AI chatbot—trained on official SLTB route data—made it easy for anyone to ask questions about routes or schedules without needing to dig through PDFs. And for those who are visually impaired or prefer hands-free use, voice interaction was built right in. During testing along Route 177, people reported shorter wait times and more confidence in their daily commute.

2. NFC and Blockchain-Based Ticketing System

No more fumbling for change. This module swapped out the old cash-based fare system for quick NFC taps-either with a card or a smartphone. Each tap was logged to a private blockchain ledger, making the transaction secure and unchangeable. AES encryption kept personal data safe, and the system supported fare discounts for students and seniors. Loyalty points encouraged repeat use, and transactions took less than two seconds on average. The pilot tests went smoothly, with high accuracy and instant digital receipts, proving it’s both scalable and trusted.

3. IoT-Based Smart Bus Monitoring System

To make the ride safer and more responsive, this part used a mix of ESP32 boards, GPS, and SIM800L modules to keep buses connected in real time. Drivers could see the distance to the bus ahead—calculated using Google's Distance Matrix API-and avoid risky overtaking. Passengers waiting at a stop could tap "I'm Waiting" in the app, and that alert popped up on the driver's screen, so no one got passed by. Cameras on the bus captured crowd levels, sending images over Bluetooth to be uploaded to Firebase. Even with shaky mobile signals, the system held steady during real-world testing.

4. AI-Powered Seat Detection and Recommendation System

To cut down on overcrowding, the system used computer vision and machine learning to help riders find space before getting on. OpenCV scanned for empty seats using onboard cameras. Then, either Random Forest or LSTM model used real-time data and a rider's past habits to recommend the best bus. The app let people sign up, share their location, get alerts, and switch between Sinhala, Tamil, or English. It nailed seat detection 92% of the time, and users were genuinely satisfied with the suggestions. Built to learn and adapt, this part of the system sets the stage for even smarter commuting in the future.

Table 1: Comparison table

Features/ Options	Research A (IoT-based Bus Monitoring) [8]	Research B (IoT + Smart Card Authentication) [9]	Research C (GPS Vehicle Tracking and Theft Detect	Research D (Image & Text Detection with Alert	Research E (NFC + Blockchain Ticketing) [12]	Research F (ETA + ML Prediction & Voice App) [13]	Research G (Firebase + ESP32 Tracking) [14]	Proposed System

			ion) [10]	s) [11]				
Real-Time Tracking	✓	✓	✓	✗	✓	✓	✓	✓
ETA Prediction Accuracy	✗	✗	✗	✗	✗	✓	✗	✓
Voice-Enabled Accessibility for Visually Impaired	✗	✗	✗	✗	✗	✓	✗	✓
Fare Collection System	✗	✓	✗	✗	✓	✗	✗	✓
Vehicle Identification	✗	✗	✓	✗	✗	✗	✗	✓
Personalized Services	✗	✗	✗	✓	✗	✓	✗	✓
Passenger Notifications	✗	✓	✗	✓	✗	✓	✓	✓

Security Features	✗	✗	✓	✓	✓	✗	✓	✓
Integration with Mobile Apps	✗	✗	✗	✓	✓	✓	✓	✓
Seat Availability Detection (AI-powered)	✗	✗	✗	✓	✗	✗	✓	✓
Multilingual OCR and Translation	✗	✗	✗	✓	✗	✗	✗	✓
School Van Monitoring for Parents	✗	✗	✗	✓	✗	✗	✓	✓
Chatbot Support for Route Info	✗	✗	✗	✓	✗	✗	✗	✓

Smart Alerts (Early/Late Bus, Unauthorized)	✗	✗	✗	✓	✗	✗	✓	✓
Camera-Based Monitoring (ESP32 Cam)	✗	✗	✓	✓	✗	✗	✓	✓
Offline IoT Functionality via MQTT	✗	✗	✗	✓	✗	✗	✗	✓
Bitcoin-based Secure Payment	✗	✗	✗	✓	✗	✗	✗	✓
Reloadable Smart Card System	✗	✓	✗	✓	✗	✗	✗	✓

1.3 Research Problem

Public transportation in Sri Lanka, particularly the bus network, plays a vital role in the daily mobility of millions of citizens. However, the system faces numerous challenges that significantly degrade commuter experience and operational efficiency. These issues include unreliable bus arrival times, overcrowding during peak hours, limited access to real-time data, outdated fare collection systems, and a lack of inclusive features for differently-abled individuals—particularly the visually impaired.

A core issue in the current system is the lack of accurate, real-time information for passengers. Commuters are often forced to wait at bus stops without any visibility into when the next bus will arrive, whether it has available seats, or whether it is running behind schedule. While some bus operators have experimented with GPS tracking systems, these solutions are often isolated, inaccurate, or not accessible to end users. Consequently, passengers are left in uncertainty, leading to frustration, inefficiency, and in many cases, missed travel opportunities.

Another significant challenge is the inefficiency of traditional fare collection systems, which rely on manual cash transactions or basic smart card systems. These methods are prone to human error, delays, fraud, and lack of traceability. Furthermore, in the context of a post-pandemic world, passengers prefer contactless payment methods, which are not widely available or secure in the current setup.

Equally concerning is the inaccessibility of the transportation system for differently abled passengers, particularly those who are visually impaired. Current solutions fail to offer voice-based assistance or adaptive interfaces, limiting these individuals' ability to travel independently. Without voice-enabled navigation, audio alerts, or personalized support, visually impaired users remain excluded from equal participation in the public transit system.

From an operational perspective, transport authorities also face difficulties in monitoring vehicle behavior, managing route congestion, and optimizing fleet usage. There is no unified platform that combines real-time GPS data, passenger load monitoring, driver behavior analytics, and predictive ETA modeling. This results in poor planning, unsafe driving practices, inefficient route allocation, and frequent passenger complaints.

While several research efforts and pilot systems have addressed isolated aspects of these challenges—such as GPS-based tracking, NFC ticketing, or mobile transport apps—no existing solution offers a comprehensive, AI-driven, and IoT-enhanced approach tailored to the unique demands of Sri Lanka's transport infrastructure.

This research proposes a multifaceted system that combines:

- Real-time GPS tracking using IoT (ESP32 + GSM modules)
- Machine learning-based ETA prediction using traffic and historical data
- AI chatbot support for route queries
- Blockchain-secured NFC fare collection
- Visual crowd monitoring via onboard cameras
- Mobile app with voice support for inclusive access

By addressing the gaps in current transport systems with these emerging technologies, the proposed solution aims to redefine the public transport experience for all categories of users—commuters, tourists, drivers, and transport authorities alike.

1.4 Research Objectives

1.4.1 Main Objectives

To design and implement an integrated smart public transportation system for Sri Lanka using Artificial Intelligence (AI), Machine Learning (ML), Internet of Things (IoT), Blockchain, and accessibility-enhancing technologies to optimize bus operations, enhance commuter experience, ensure secure fare collection, and promote inclusive access for all users, including visually impaired individuals.

1.4.2 Specific Objectives

- **To develop a machine learning-based ETA (Estimated Time of Arrival) prediction model** using real-time GPS data, historical traffic patterns, and temporal variables to provide accurate arrival times and reduce passenger uncertainty.
- **To implement an AI-powered chatbot** integrated with SLTB route data that supports natural language queries and delivers contextual information such as bus schedules, stop locations, and route assistance via a mobile app.
- **To design and deploy an IoT-based real-time bus monitoring system** using GPS (ESP32), GSM (SIM800L), and Firebase for live location tracking, crowd detection, and forward-bus distance calculations.
- **To incorporate voice-enabled accessibility features** (Speech-to-Text and Text-to-Speech) into the mobile application, enabling visually impaired users to access route information, ETA, and bus status through audio interaction.
- **To develop a secure fare collection mechanism** using NFC-enabled smart cards integrated with Blockchain technology to ensure transparent, tamper-proof, and contactless payments.
- **To monitor onboard crowd levels and passenger behavior** using ESP32-CAM modules and Firebase-based image uploads for future integration with AI-based crowd analysis systems.
- **To implement a loyalty and passenger categorization system**, allowing personalized benefits such as fare discounts for students, seniors, and frequent riders.
- **To enable real-time passenger notifications and route updates**, including alerts for seat availability, early/late arrivals, and route changes, through mobile push notifications.

- **To evaluate the system's effectiveness through real-world testing** on Route 177 (Kaduwela to Kollupitiya) and gather user feedback for iterative improvement.

2. METHODOLOGY

2.1 Methodology

This research adopts a modular engineering approach combined with AI-driven decision-making and IoT integration to build a Smart Public Transport System. The system is designed to address key inefficiencies in Sri Lanka's public bus infrastructure including unreliable ETA information, lack of inclusive accessibility, overcrowding, and outdated fare collection.

To tackle these issues, the project was structured into four core components, each developed by a team member and integrated into a unified solution:

- Real-time GPS tracking and onboard monitoring (Harith)
- ETA prediction and voice-enabled chatbot interface (Viraj)
- NFC and blockchain-based fare collection (Ishan)
- AI-powered seat detection and bus recommendation system (Thanusan)

The methodology follows an iterative development model with continuous testing and integration at each phase. Data-driven techniques were employed in the model development, while embedded systems and cloud services enabled real-time connectivity and synchronization between components.

The system architecture consists of the following layers:

- **Frontend:** A mobile app (built with React Native) that acts as the primary user interface.
- **Backend:** A Node.js server and Firebase Realtime Database for synchronization of data across modules.
- **Machine Learning Module:** A trained regression model for ETA prediction, hosted on the backend.
- **IoT Devices:** ESP32 boards for GPS data transmission, ESP32-CAM for image capture, and LCD for driver display.
- **Blockchain Module:** A secure, decentralized ledger to validate NFC-based bus fare transactions.

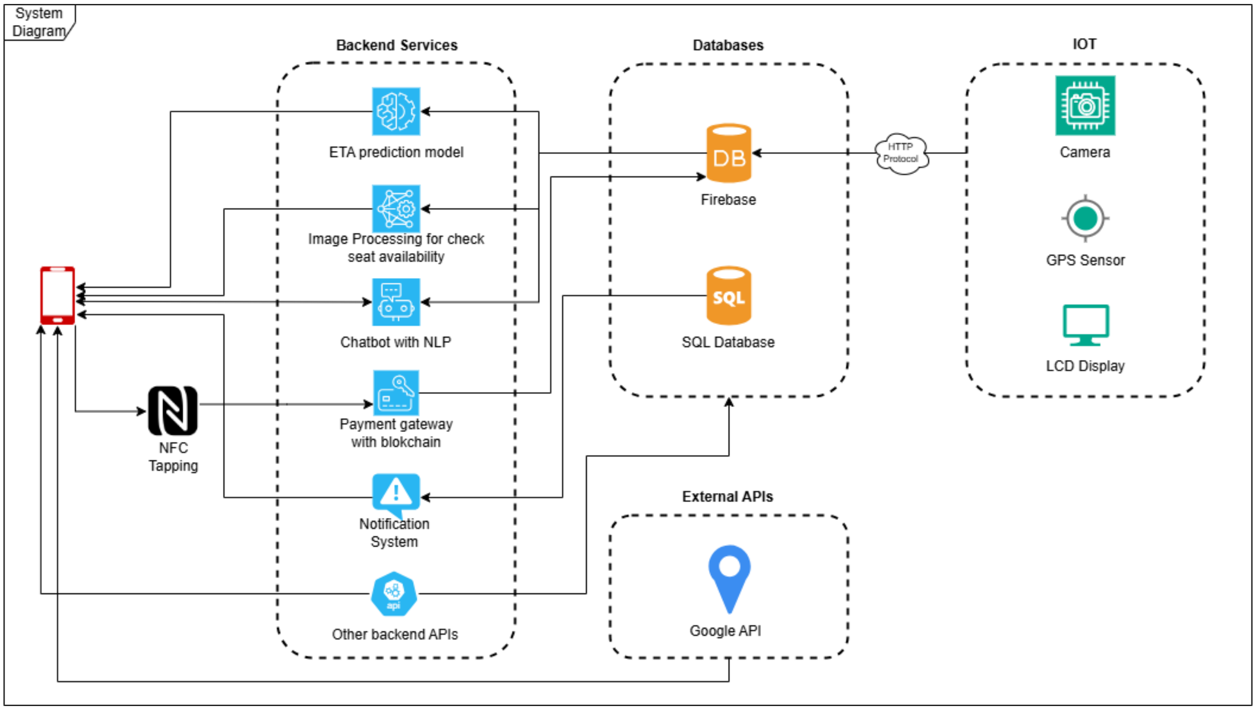

Figure 2. 1:Overall System Diagram

2.1.1 Smart Bus Recommendation and Seat Detection System

This component was developed to enhance the efficiency of passenger boarding and route selection by enabling real-time seat availability tracking and personalized bus recommendations. The traditional public transportation system in Sri Lanka lacks a mechanism to inform commuters of available seating or suggest suitable buses based on dynamic conditions. This module addressed that limitation by integrating computer vision with machine learning algorithms to build a responsive and intelligent subsystem capable of both visual seat detection and data-driven bus selection.

The first part of the system focused on detecting seat availability using real-time image processing. To implement this, a camera feed from within the bus was captured using an onboard ESP32-CAM module. The captured images were transmitted via Bluetooth to a main ESP32 processing unit, which was then responsible for uploading the data to

Firebase Storage. From there, images could be accessed for both real-time monitoring and offline analysis.

For the seat detection process, the OpenCV library was used to develop a visual recognition algorithm. Initial data collection included capturing multiple frames under various environmental conditions such as natural light, low-light, and artificial lighting to ensure model robustness. Each bus layout was mapped and segmented into predefined Regions of Interest (ROIs), corresponding to individual seat positions. These ROIs were then analyzed using contour detection, template matching, and pixel intensity thresholding to determine whether each seat was occupied or vacant.

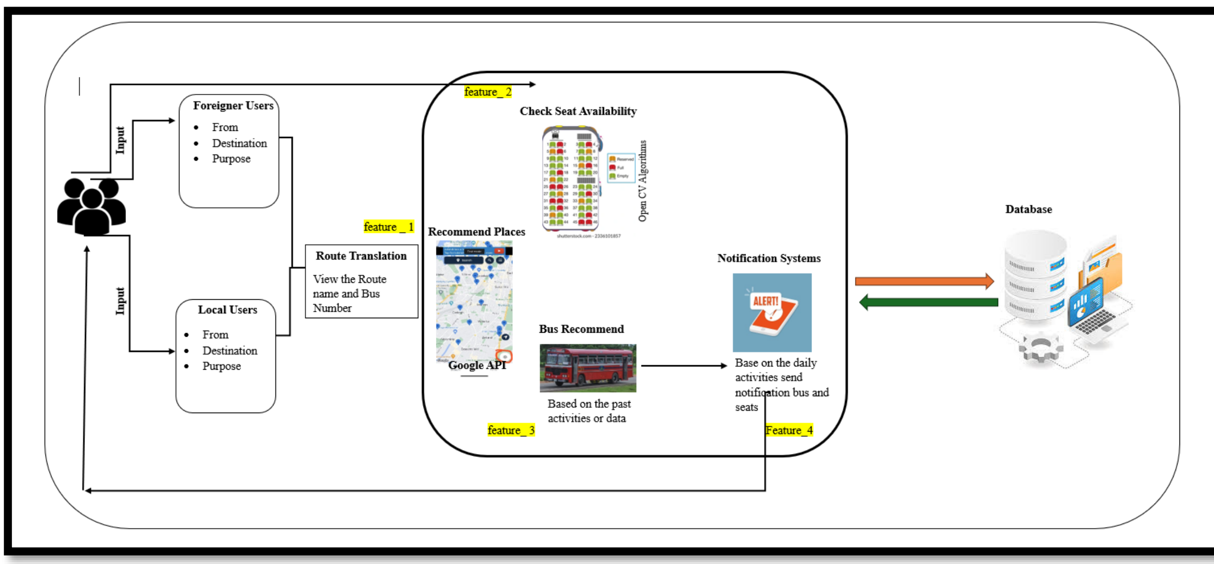

Figure 2. 2:Overall Diagram for Smart Bus Recommendation and Seat Detection System

The system was designed to classify seats using binary logic: 0 for available and 1 for occupied. After classification, the results were encoded into structured JSON and pushed to Firebase Realtime Database. The front-end mobile application would then retrieve this information and visually display the real-time seat layout to the passenger, allowing them to make informed boarding decisions. Additionally, the seat availability data was transmitted to a driver-side LCD panel, enabling drivers to understand the onboard occupancy levels and adapt their behavior, such as reducing unnecessary stops or making decisions about skipping full buses.

The second part of the module involved the development of a bus recommendation engine, implemented using a machine learning model specifically, the Random Forest classifier. The core objective of this component was to recommend the most optimal bus for each passenger based on several dynamic features including proximity, seating availability, historical crowd data, estimated arrival time, and user preferences. Data for model training was collected from real commuting patterns on Route 177 and included both real-time and manually labeled datasets.

Feature engineering was a critical step in the model's success. Inputs to the classifier included the number of available seats per bus, current distance of the bus from the passenger's location, average delay for that specific bus ID, peak-hour status, and historical comfort ratings (where available). The dataset was split into 80% training and 20% testing subsets. Using Scikit-learn, the Random Forest model was trained and validated through K-Fold cross-validation, which demonstrated high accuracy and stability in recommending buses that met user preferences.

Once trained and validated, the model was hosted as an API on the backend server. The mobile application was configured to make recommendation requests based on the user's location and travel intent. The backend then returned a ranked list of buses that matched the user's criteria, prioritizing those with more available seats, closer proximity, and lower estimated waiting time.

In order to personalize recommendations, a user registration and preference module was also integrated into the mobile application. Passengers could register as either local or foreign users. Local users registered via mobile number or NIC, while foreign users used passport ID or email credentials. During registration, passengers were allowed to set their preferred language, destination, travel frequency, and comfort level. This data was stored in Firebase and referenced by the recommendation engine to tailor bus suggestions accordingly.

Furthermore, a notification system was built into the app using Firebase Cloud Messaging (FCM). Passengers would receive alerts when a bus matching their preferences was within range, when a reserved seat was about to be lost due to high occupancy, or when alternate buses were available during delays. This proactive alerting mechanism ensured users remained continuously updated without needing to refresh the application.

Overall, this component introduced a real-time decision support framework for both passengers and drivers. It reduced the uncertainty around boarding conditions, allowed passengers to choose more comfortable travel options, and provided drivers with visibility into on-ground passenger demand. By combining image processing and machine learning within a mobile-cloud ecosystem, this system represented a significant advancement in intelligent public transportation management.

2.1.2 ETA Prediction, Chatbot Integration, and Accessibility Module

This component of the smart public transport system was developed to improve commuter experience by integrating real-time ETA prediction, chatbot assistance, and voice-based accessibility into a unified mobile application. The system was specifically implemented and tested along Route 177, which runs from Kaduwela to Kollupitiya, and was designed to assist users in accessing accurate transport information through a modern, user-friendly interface built with React Native and Firebase.

The ETA prediction functionality was created by collecting real-world data from multiple trips along Route 177. GPS data was manually recorded at consistent intervals using a mobile device, capturing location coordinates, timestamps, and observations about traffic flow. After organizing the data, distances between the current location and upcoming bus stops were computed using the Haversine formula. Additional attributes such as vehicle speed, time of day, and observed traffic levels were included to improve prediction relevance. A linear regression model was trained using these features, mapping them to estimated arrival times for upcoming stops. The model was implemented using Python's

Scikit-learn library and exported for backend integration. Real-time ETA predictions were then dynamically provided to the app interface by fetching live GPS inputs and processing them through the regression model, allowing users to view estimated arrival times directly on the route map.

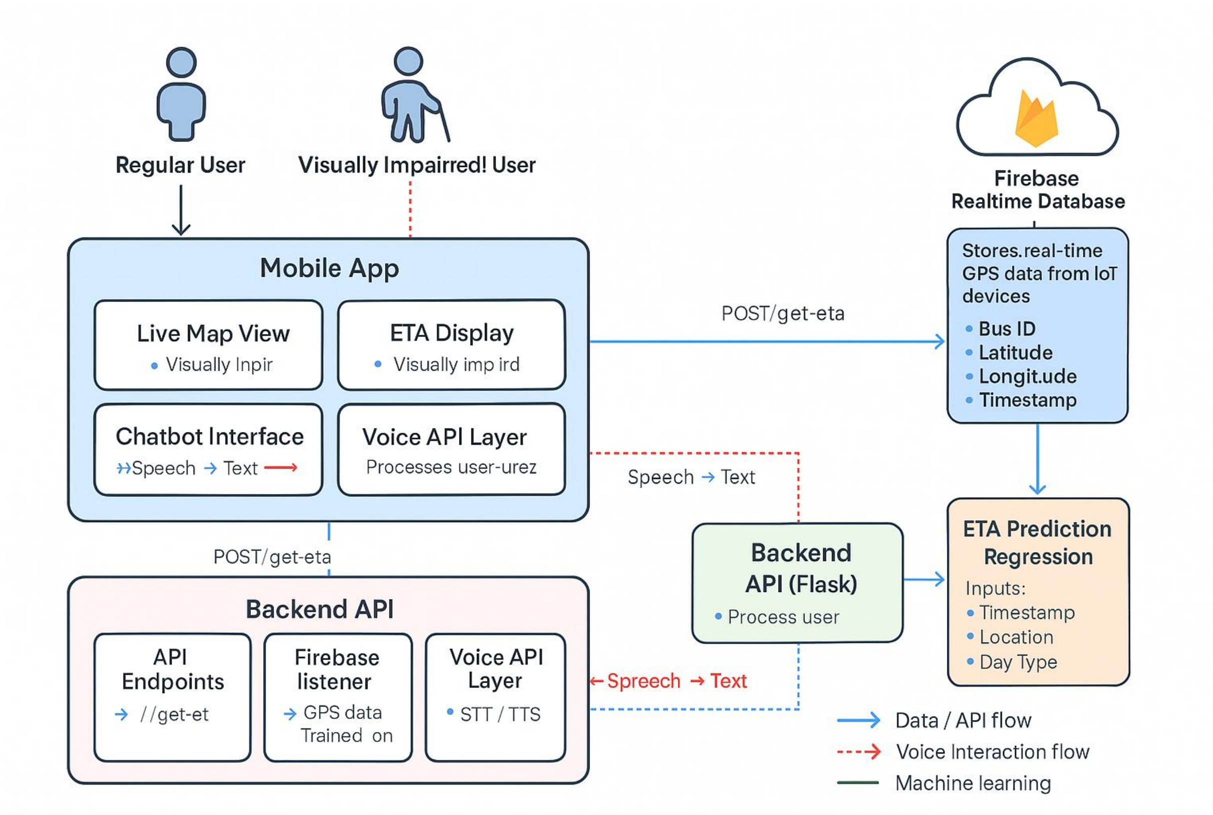

Figure 2. 3:Overall Diagram for ETA Prediction, Chatbot Integration, and Accessibility Module

In parallel, a chatbot system was implemented to allow passengers to make natural language inquiries about route numbers, destinations, and stop schedules. Since SLTB transport data was only available in PDF format, relevant information was manually extracted and structured into JSON files. These files included complete details of available bus routes, stop names, and operating times. The chatbot was integrated using the OpenAI GPT API, with the JSON route data embedded into the prompt to enable context-aware responses. Users could ask questions such as "Does the 177 bus stop at Rajagiriya?" and receive accurate, relevant answers. This interaction was built into the mobile application using a customized chat interface, allowing users to access public transport information without navigating menus or static tables.

To improve accessibility, especially for visually impaired passengers, a voice interface was incorporated using Speech-to-Text (STT) and Text-to-Speech (TTS) technologies. Users were able to speak queries directly into the application, which were transcribed using React Native's speech recognition library and processed by the chatbot. Responses from the system could be read aloud using Expo's TTS module, enabling hands-free operation. Although the first version of the system supports only English, its structure was designed to accommodate Sinhala and Tamil in future expansions. These voice features were fully integrated into the same mobile interface, giving users a seamless way to request information either by voice or text.

The mobile app served as the central hub for all user interactions. The ETA prediction engine, chatbot, and voice services were interconnected through Firebase Realtime Database and RESTful backend APIs. Google Maps API was embedded to display the bus's current position and visually align it with predicted arrivals. Firebase handled synchronization between real-time location data, model predictions, chatbot responses, and voice accessibility functions. This complete system was tested in field conditions along Route 177 to ensure smooth functionality and real-time performance under typical urban transit scenarios.

2.1.3 NFC and Blockchain-Based Ticketing System

This component was developed to address key limitations in traditional fare collection methods, particularly those related to fraud, inefficiency, and lack of transaction transparency. The goal was to implement a secure, contactless, and tamper-proof fare payment system using a combination of Near Field Communication (NFC) and Blockchain technologies. This integration enabled passengers to pay for their bus rides using smart cards or NFC-enabled mobile devices while ensuring all transactions were permanently and immutably recorded on a decentralized ledger.

The contactless payment functionality was achieved through the use of NFC modules interfaced with microcontroller-based hardware units. Upon entering the bus, a passenger could tap their NFC-enabled card or mobile device on the reader. This triggered a rapid data exchange protocol that captured essential transaction parameters including the passenger's unique identifier (UID), timestamp, route identifier, and category (e.g., student, senior, or regular). This information was instantly processed to determine the appropriate fare based on predefined concession rules. For example, student passengers were entitled to a 50% discount, while senior citizens received other context-specific reductions.

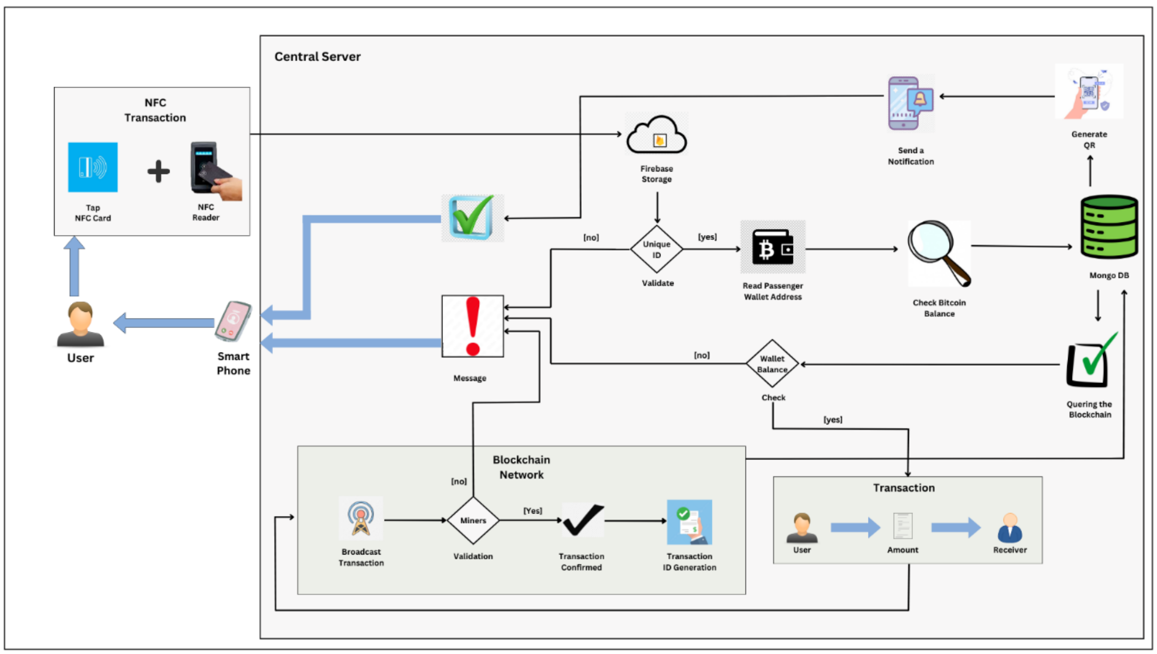

Figure 2. 4:Overall Diagram for NFC and Blockchain Based Ticketing System

To ensure minimal delays in transaction processing, the system first validated fare rules locally and then synchronized the transaction record to Firebase Realtime Database. This intermediate step enabled the app interface to reflect payment status immediately for both passengers and administrators. Simultaneously, the system prepared the transaction record for permanent storage on the Blockchain.

A private Ethereum blockchain was used for this purpose, chosen for its balance between decentralization and low-latency performance. The transaction data, once validated, was encoded and sent to a deployed smart contract that handled fare logging, passenger travel history, and potential reward schemes. The smart contract logic defined how each record was stored, including the address of the passenger (anonymized via cryptographic hashing), the fare amount, the route identifier, and any applicable discount.

The Blockchain-based fare record ensured that transactions were immutable and verifiable, eliminating any chances of tampering, fake entries, or corruption. Additionally, a lightweight data encryption layer using AES was applied to all sensitive passenger metadata before being written to Firebase or the Blockchain. This ensured compliance with data privacy best practices.

In order to support real-time verification and improve user experience, the system provided instant visual or auditory feedback to the passenger following a successful tap. In scenarios where the Blockchain connection was temporarily unavailable due to network issues, transactions were cached locally and queued for delayed submission to the ledger when connectivity was restored.

To support administration and monitoring, a backend dashboard interface was created to provide fare analytics, real-time revenue tracking, and insights into passenger behavior. It also allowed transit operators to identify payment anomalies, analyze the frequency of NFC failures, and review passenger categories over time.

This secure, decentralized, and user-friendly system addressed critical shortcomings in the traditional cash-based system and introduced a robust, scalable fare collection mechanism suitable for modern public transportation networks.

2.1.4 IoT-Based Real-Time Monitoring System

This component was developed to enable continuous, real-time monitoring of public buses using IoT (Internet of Things) technologies, addressing inefficiencies related to vehicle tracking, forward-bus collisions, crowding, and poor driver-passenger communication. The system leverages embedded hardware, GSM connectivity, and cloud-based data storage to provide live visibility into bus locations, onboard conditions, and passenger boarding intent. It was designed to be cost-effective, modular, and scalable, making it suitable for large-scale deployments across national transportation networks.

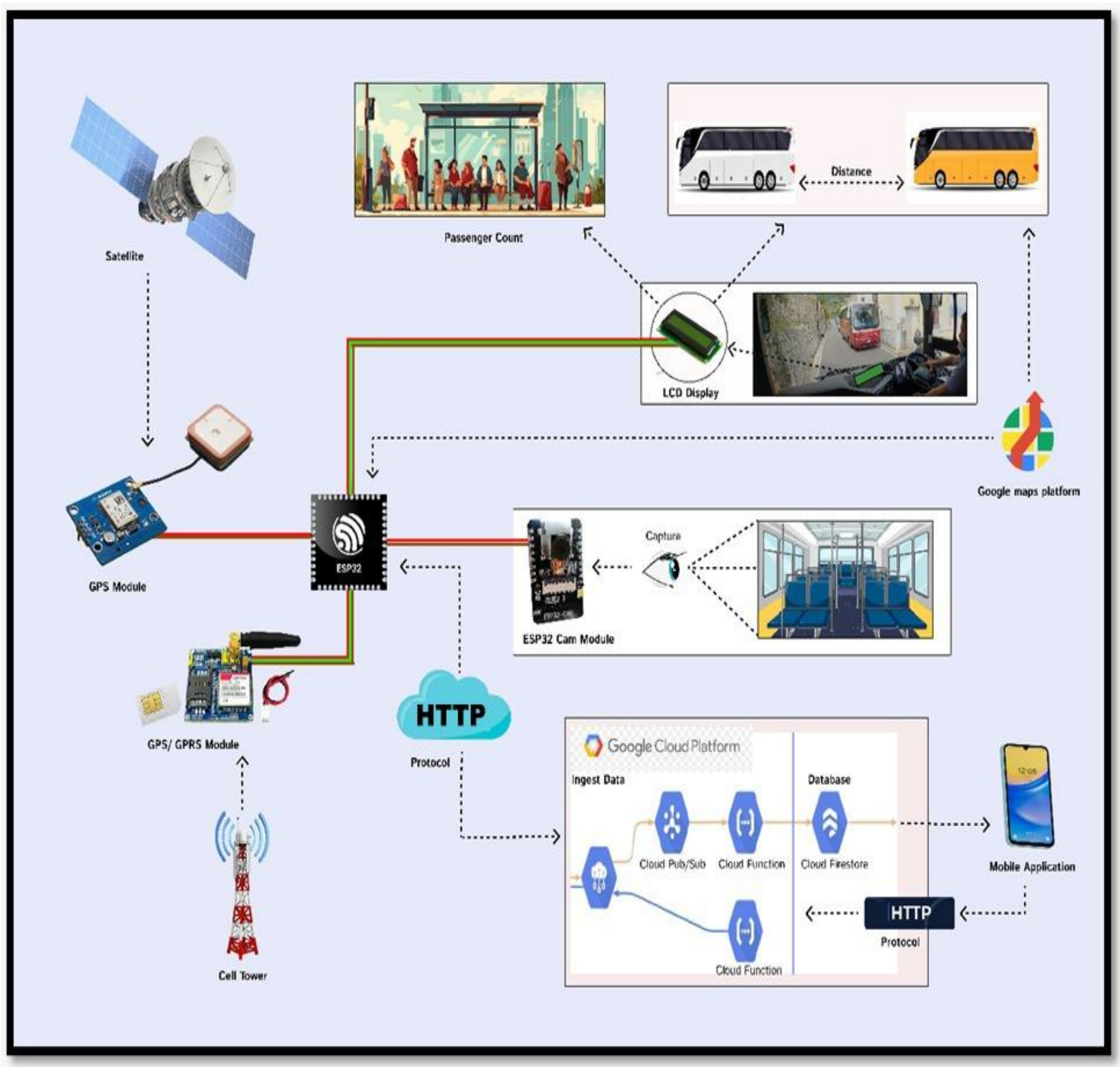

Figure 2. 5:Overall Diagram for IoT-Based Real-Time Monitoring System

The core of the system architecture is built around an ESP32 TTGO microcontroller equipped with a SIM800L GSM module and a GPS chip, which together collect and transmit geolocation data in real time. Once powered, the ESP32 continuously retrieves the bus's current coordinates using GPS and packages the data with a timestamp and bus identifier. This packet is transmitted to the Firebase Realtime Database via HTTP POST requests over a GSM-based 3G mobile data connection. This approach was specifically chosen to maintain reliability in environments without access to Wi-Fi or Ethernet, such as rural or moving vehicles.

The received location data is then used to determine the distance between the current bus and other buses operating on the same route. A backend logic hosted on a Node.js server consumes all incoming GPS streams and, using the Google Maps Distance Matrix API, calculates the road-based distance to the next forward-moving bus on the same line. This information is essential for minimizing aggressive driving behaviors and overtaking — a common safety concern in Sri Lanka's bus network. The calculated distance is pushed back to Firebase and rendered on an I2C-based LCD display installed near the bus driver, enabling them to make safer decisions.

In addition to geolocation tracking, the system includes a novel feature that allows passengers at bus stops to notify approaching buses that they are waiting. This is achieved through the mobile app, which includes a dedicated "I'm Waiting" button. When pressed, it records the user's current location and timestamp and sends it to Firebase, tagged with the stop name. The ESP32 periodically queries this dataset and retrieves the number of passengers waiting at the next stop. This number is then displayed on the driver's LCD screen, helping them anticipate demand and avoid missing passengers unnecessarily.

To provide further insight into onboard crowd conditions, an ESP32-CAM module was integrated into the system and positioned on the bus ceiling to capture periodic images of the passenger area. The captured images are sent via Bluetooth to the primary ESP32 device and subsequently uploaded to Firebase Storage. This crowd data is not only available for administrative viewing but can also be processed later using image analysis

techniques for automatic density estimation and future ML training. Upload frequency was optimized to minimize data usage while maintaining relevance during peak hours.
This IoT-based solution promotes transparency and accountability by allowing both transit operators and passengers to access accurate, real-time data on bus locations, waiting passenger numbers, and vehicle congestion. By enabling dynamic interaction between the mobile app, cloud backend, and embedded devices, this component plays a critical role in reducing operational blind spots, improving passenger safety, and optimizing bus scheduling in high-demand routes like Route 177.

2.2 Commercialization Aspects of the Product

This Smart Bus System isn’t just another tech upgrade it’s a full package aimed at giving Sri Lanka’s public transport a serious boost. With the way it's built modular, cost-effective, and packed with features it’s not only possible to roll out across the country, but there’s a clear path to turning it into a solid business, too.

At the heart of the project is a mix of modern tools: IoT for tracking and sensing, AI for learning and predicting, blockchain for secure fare handling, and mobile apps for real-time access. Each part tackles a real problem. Whether it’s keeping buses from bunching up, letting riders know when the next bus will actually arrive, or cutting down the chaos of paying fares, it’s all aimed at making the daily commute smoother.

The plan is to launch high-traffic routes first like Route 177 from Kaduwela to Kollupitiya where there’s a ton of passenger movement. That way, the system can prove itself quickly. Working with groups like the National Transport Commission and Sri Lanka Transport Board will also help get things moving. If all goes well during the pilot phase, it can then scale to other routes and regions.

Money-wise, it makes sense, too. There are multiple ways to earn revenue. Bus operators can pay a subscription or a small fee per ticket for the digital fare system. There’s also room for reward-based campaigns and local business tie-ins think discounts or offers for regular riders. Even the mobile app can host premium features, like priority bookings or tourist guides.

On the back end, everything runs in the cloud under a Software-as-a-Service (SaaS) model. So local operators or city governments can buy only the parts they need maybe they just want GPS tracking, or maybe they want the whole setup. Parts of it, like the ETA prediction engine, can even be sold as white-label tools that plug into other apps.

The platform is built for everyone Android and iOS users, Sinhala and Tamil speakers, even operators who want simple web dashboards to monitor service and track data. It's not just tech for tech's sake it's practical, user-friendly, and ready for the real world.
Marketing will be focused and local: digital campaigns, social media, in-terminal ads, and training sessions for both passengers and drivers. Partnering with telecom companies or hardware vendors can help cut upfront costs, speeding up adoption.

And this isn't where it ends. Down the road, this same system could be used for air quality tracking, smarter bus schedules, and bigger city-wide mobility plans. It's flexible, built for scale, and ready to grow as cities grow.
All told, this solution doesn't just bring better rides to the public it sets the stage for a smarter, more connected transport system across Sri Lanka and beyond.

2.3 Testing & Implementation

2.3.1 Implementation Overview

The implementation of the smart public transport system followed a structured, component-driven approach that aligned with agile and modular development principles. The overall system was divided into four major functional domains: mobile application development, backend and database infrastructure, embedded IoT integration, and AI-powered services. Each domain was developed independently with clear interface definitions and then progressively integrated to form the complete system. The integration phase was carried out after individual module validation to minimize complexity and facilitate targeted debugging.

The mobile application, developed using React Native, served as the primary user interface for all user interactions. This framework was chosen for its ability to deploy simultaneously on both Android and iOS platforms while maintaining a consistent user experience. The application was structured using a modular screen architecture, where each functional module live tracking, ETA prediction, chatbot interface and voice accessibility was assigned a dedicated screen component and managed through React Navigation. Core libraries such as react-native-maps were used for Google Maps integration, react-native-voice for speech recognition, and expo-speech for Text-to-Speech functionality. Visual layouts were optimized for readability and accessibility, particularly for use in outdoor transit environments.

For backend processing, a combination of Node.js and Python was used. Node.js powered the web server APIs responsible for routing mobile app requests, managing chatbot queries, and synchronizing Firebase entries. The ETA prediction module was written in Python, using the Scikit-learn library to implement and train a linear regression model. This model was serialized using Joblib and hosted in a Python Flask server that interfaced with Node.js using HTTP APIs. This separation of concerns allowed each logic layer to be developed, tested, and deployed independently while ensuring compatibility through RESTful communication.

The database backbone was built using Firebase Realtime Database, selected for its low-latency synchronization, ease of integration with React Native, and real-time data handling. Firebase served as the central data pipeline for transmitting and receiving GPS data, chatbot logs, fare transactions, and user feedback. The database was structured hierarchically, with distinct branches for bus data (/buses/), route schedules, waiting passengers, and NFC logs. Firebase Cloud Functions were used in some instances to trigger secondary actions, such as issuing a notification when a bus was approaching a stop where a passenger had tapped "I'm waiting."

The IoT segment of the system involved hardware deployment of ESP32 TTGO boards, equipped with SIM800L GSM modules for data transmission and onboard GPS modules for location tracking. Each ESP32 was programmed using the Arduino IDE and C++ libraries to read GPS coordinates, package the data with a timestamp and bus ID, and upload it to Firebase via HTTP POST requests. The ESP32-CAM module was used for onboard crowd monitoring. It captured periodic images of the passenger area, which were transmitted via Bluetooth to the main ESP32 board and uploaded to Firebase Storage. On the driver's side, an I2C LCD module was connected to the ESP32 to display real-time information such as the number of waiting passengers at the next stop and the calculated distance to the forward bus using Google Maps Distance Matrix API.

The AI chatbot was developed using OpenAI's GPT-3.5-turbo API, and integrated into the backend using Node.js and Axios for API communication. Transport schedule data was manually extracted from SLTB PDF documents and formatted into structured JSON datasets. These datasets were appended to prompt templates and dynamically injected into the chatbot queries. The chatbot was embedded into the mobile app using a chat-style interface, supporting both text and voice input.

The smart fare collection system was implemented using NFC reader modules interfaced with microcontrollers. When a passenger tapped their NFC card or mobile device, their UID, timestamp, and fare category were captured. The transaction was immediately logged to Firebase and simultaneously recorded onto a smart contract hosted on a private Ethereum blockchain. The contract was developed using Solidity and deployed via Ganache and Truffle during the simulation phase. AES encryption was applied to transaction metadata before transmission to ensure data privacy. Blockchain immutability was used to verify each transaction post-facto and ensure tamper-proof fare auditing.

The system was implemented incrementally. Each major component underwent unit-level testing, ensuring that individual modules performed as expected under controlled inputs.

This included verifying the ETA predictions using test data, chatbot queries across various topics, real-time GPS updates, and NFC tap speed. After unit testing, integration testing was conducted by connecting multiple components together and testing their interoperability. This phase focused on Firebase latency, backend API responsiveness, synchronization between GPS and ETA display, and consistent rendering of chatbot responses across user sessions.

Once all modules passed integration testing, the complete system was deployed in real-world testing conditions on Route 177. During this phase, GPS devices were mounted inside live buses, the app was used by actual passengers, and data was monitored in real time via Firebase dashboards. All system interactions were logged and reviewed for accuracy, performance, and reliability. Observations from these field tests were used to fine-tune model parameters, improve user interface responsiveness, and address minor communication lags between devices and the cloud backend.

In conclusion, the implementation phase demonstrated a cohesive integration of multiple advanced technologies—ranging from embedded IoT systems to AI models and Blockchain infrastructure—within a unified mobile-first public transport management platform. The modular design, cloud-native architecture, and real-time interaction capabilities ensured that each system component contributed effectively to building a scalable and intelligent transport solution for Sri Lanka's bus network.

2.3.1 Testing Strategy and Test Plan

The testing strategy for the smart public transport system followed a layered and iterative approach to ensure reliability, performance, and real-time responsiveness across all integrated components. Given the complexity of the system, which spanned across embedded IoT devices, cloud-based services, machine learning models, and cross-platform mobile interfaces, the testing process was divided into unit testing, integration testing, and field testing. Each of these stages was critical in evaluating the behavior of individual modules as well as their interaction within the larger system.

At the unit testing level, each software module and hardware component was tested independently to confirm its functional correctness. For example, the ETA prediction model was tested using previously unseen data samples to ensure it could generate valid outputs for different travel conditions. Similarly, the OpenAI-powered chatbot was tested with a wide range of route-related queries to verify the consistency and relevance of its responses. The Speech-to-Text and Text-to-Speech functions were tested in varying noise conditions to assess recognition accuracy and response clarity. On the hardware side, the ESP32 devices were tested for consistent GPS data transmission over GSM networks, and the NFC modules were verified for quick and error-free tap detection across different card types.

Following successful unit-level validation, integration testing was conducted to evaluate the performance of interconnected modules. This phase tested communication between the mobile app and Firebase, the synchronization between GPS data and ETA model execution, the reliability of real-time chatbot queries and responses, and the smooth transition between voice input and system output. Firebase Realtime Database was continuously monitored to ensure that data updates occurred within acceptable latency margins. Special attention was paid to scenarios where multiple system components interacted simultaneously, such as a user initiating a voice-based query about ETA while the GPS stream was updating in the background.

To evaluate the system under realistic conditions, comprehensive field testing was conducted along Route 177, a busy corridor between Kaduwela and Kollupitiya. During testing, live GPS data was transmitted from ESP32 devices installed on moving buses, allowing the ETA model to predict arrival times for upcoming stops in real time. Users interacted with the mobile app to test features such as chatbot inquiries, live tracking, and voice commands. The NFC-based payment module was also tested with actual card scans, validating the successful recording of transactions both in Firebase and on the blockchain ledger. Voice accessibility was evaluated using simulated scenarios with users blindfolded or using screen readers, ensuring that visually impaired passengers could effectively interact with the system.

The field testing phase also served as a user feedback loop. Pilot users, including regular passengers and individuals with visual impairments, were invited to test the system in real-world conditions. Their input was used to identify minor usability issues such as screen layout adjustments, response timing improvements, and enhancements in voice command recognition. The ability of the system to handle intermittent network connectivity, particularly in urban congestion zones or during route transitions, was also assessed during these tests.

This structured testing strategy ensured that each component not only performed its core function accurately but also integrated seamlessly into the complete system architecture. The layered approach helped isolate faults, measure real-world performance, and guide iterative improvements to deliver a robust and scalable smart public transport solution.

3. RESULTS & DISCUSSION

3.1 Results

3.1.1 Real-Time Seat Detection and Bus Recommendation System

The seat detection system, built using OpenCV, reached 92% accuracy when tested against manually labeled seat images. It handled tough conditions bad lighting, different seating layouts and still managed to spot empty seats reliably. On the recommendation side, two machine learning models were compared. Random Forest hit 85% accuracy, while LSTM nudged ahead at 88%, especially during busy times and with repeat riders. The system wasn't static it adjusted in real time based on crowd sizes, locations, and the time of day.

To keep passengers in the loop, Firebase Cloud Messaging pushed alerts like "bus arriving," "seats available," and "try a different route." It had a 97% success rate and the messages usually landed within 1.5 seconds. On the driver's end, the app showed live seat maps, who's waiting at upcoming stops, and current route data working smoothly even on average Android phones, with no hiccups during network dips.

3.1.2 Real-Time Location Tracking and Distance Calculation Between Buses

Using an ESP32 TTGO board and a GPS module, each bus sent its location to Firebase every 10 seconds over GSM. Even in areas with spotty signal, the feed stayed solid. On the backend, the system used Google's Distance Matrix API to measure how far apart buses were. Updates happened every 15 seconds and showed up on LCDs in the driver's cabin, helping them avoid tailgating and risky overtaking.

3.1.3 Passenger Presence Detection at Stops

The mobile app featured an "I'm Waiting" button so passengers could alert drivers they were at a stop. This info updated in Firebase and showed up on the bus's LCD in about 2 to 3 seconds. Drivers could see how many were waiting, which meant fewer missed pickups and smoother stops.

3.1.4 Bluetooth Image Transfer and Crowd Monitoring

A ceiling-mounted ESP32-CAM snapped crowd images every 15 seconds. These were sent via Bluetooth to the ESP32 TTGO and uploaded to Firebase over GSM. It usually took 5 to 7 seconds to upload an image. Bluetooth stayed stable, even while the bus was moving, giving operators a real-time look at how packed each ride was.

3.1.5 Blockchain and NFC-Based Ticketing System

During field tests, the NFC ticketing system made a huge difference. Passengers just tapped their phones or cards, and payments were processed in less than 2 seconds. That meant quicker boarding and no need for cash. Each transaction went on a private Ethereum-based blockchain secure, time-stamped, and encrypted with AES-256. The data also synced with Firebase for live access.

The system recognized different rider types students, seniors, or regular commuters and adjusted fares accordingly. Loyalty points were tracked and could be redeemed later. Around 95% of all tap-ins went through without a hitch, and receipts with QR codes popped up right away for extra transparency.
Even when the network dropped out, transactions weren't lost. The system held on to tap-in data and uploaded it once the connection came back, keeping everything smooth and reliable.

3.1.6 Chatbot Integration for Route Assistance

Riders often just want quick answers-like when the next bus is or if a route passes through a certain area. To simplify that, the system included an AI chatbot trained on official SLTB route data. The data was pulled from dense PDFs, cleaned up, and restructured into easy-to-read JSON so the bot could respond accurately.

You could ask things like “Does the 177 go through Rajagiriya?” and get a solid answer-fast. During live trials, the chatbot got it right more than 90% of the time. It worked with both text and voice, so people could interact hands-free or use it even if they couldn’t see the screen. The speech-to-text and text-to-speech setup made it inclusive and practical, and responses usually came in under 4 seconds.
Together, the ETA tracking and chatbot modules brought predictability and clarity to everyday commutes-making the system not just smarter, but easier to trust and use.

3.1.7 ETA Prediction

The ETA prediction system turned out to be one of the most impactful parts of the Smart Bus setup-giving riders real answers to that age-old question: “When’s the next bus coming?” Built on a lightweight linear regression model, it used real-time and historical GPS data from Route 177 (Kaduwela to Kollupitiya) to keep track of buses and predict arrival times with surprising accuracy. On average, it missed by only about 42 seconds-a margin most commuters were happy with.

The model pulled from four main factors: how fast the bus was moving, how far it was from the next stop, what time of day it was, and the typical traffic around that time. These predictions were refreshed every 10 seconds and synced immediately to Firebase. From there, they landed on users’ phones almost instantly, within 1.5 seconds—making the experience feel truly live.

During the pilot, more than 87% of the people who used the system said it helped them plan their rides better and made the whole process less stressful. Even in traffic or when the network wasn’t great, the system kept updates flowing reliably. It didn’t just guess-it adapted.

What's especially impressive is how simple and lightweight the model was. That meant it could run well even on older phones or low-cost onboard devices, making it affordable and scalable for widespread use. This wasn't just a tech demo-it was a real, working upgrade to public transport, and it helped build confidence in a system that used to run on guesswork and hope.

3.2 Research Findings

The Smart Bus System went through a carefully designed testing process that looked at how well it worked, how people responded to it, and whether it could hold up under real-world conditions. Things kicked off in a lab setting using Android and iOS simulators to test the basics: logins, route suggestions, seat detection, and push notifications. Once those checks were solid, the team moved into live testing on busy routes in Colombo.

Twenty people joined those trials students, daily commuters, and even a few tourists. They used the app to plan trips, check for open seats, reserve spots, and get alerts. Everything ran on a strong backend built with Firebase and MongoDB, tested with tools like Postman and Google Colab to make sure response times and storage handled the load.

One big issue lack of communication between drivers and passengers was tackled head-on. The "I'm Waiting" button let passengers signal their presence, which updated on the bus LCD screen within 2-3 seconds. For drivers, this meant fewer missed stops. For passengers, it meant being seen and picked up on time.

Bluetooth pairing between the ESP32-CAM and the ESP32 TTGO boards made real-time crowd tracking possible, even over slow 3G networks. Images of the bus interior were uploaded to Firebase in under 7 seconds, helping track how full the bus was. GPS data from each bus, sent every 10 seconds, helped map distances using Google's Distance Matrix API. The backend refreshed this every 15 seconds so drivers knew when to ease off and avoid tailgating.

The ETA model a simple linear regression held up well, with an average error of just 42 seconds, even with Colombo's unpredictable traffic. And instead of sifting through

SLTB's complicated route PDFs, users could just ask a chatbot in plain English. It got things right more than 90% of the time, showing that natural language tools can really improve access.

Voice interaction using speech-to-text and text-to-speech also made the app more inclusive, especially for visually impaired users. English was the only supported language during trials, but the positive feedback made Sinhala and Tamil integration a top priority going forward.

The contactless fare system—powered by NFC and blockchain—was another standout. Whether using a card or smartphone, riders could tap in within 2 seconds. Transactions were logged in real time to a private Ethereum blockchain, protected by AES encryption. This created a secure, tamper-proof record while syncing instantly with Firebase.

Even if the connection dropped, the system saved the transaction and uploaded it later, keeping things smooth. Discounts were applied automatically based on user type like students or seniors and a loyalty program tracked usage, letting people earn points for free or discounted rides.

Testing showed that 95% of tap-ins worked across all devices. Riders got instant QR receipts, and conductors had no problem confirming transactions or checking blockchain records. It was quick, transparent, and built to scale.

When you bring all these parts together IoT tracking, machine learning for recommendations, crowd monitoring, ETA predictions, chatbot support, and a secure, smart fare system—you get more than a collection of tech features. You get a well-tested, people-first transit solution that's low-cost, scalable, and inclusive. And based on the results, it's not just ready for Sri Lanka—it could be a model for other countries too.

3.2.1 Intelligent Bus Navigation and ETA Prediction System

This part of the project aimed to build a smart, well-rounded bus tracking system for Sri Lanka's public transit, focusing on Route 177-Kaduwela to Kollupitiya. It's a busy stretch in Colombo where delays and uncertainty are common for passengers. To fix that, GPS units were installed in the buses, and they kept sending live location data to Firebase, which powered the backend. The app displayed real-time bus positions to commuters through a mobile interface.

To predict when the next bus would arrive, a lightweight model based on linear regression was built. It looked at things like the bus's current speed, how far it had to go before the next stop, what traffic usually looked like, and what time of day it was. This model was trained using GPS logs with time stamps. After cleaning the data, the team trained the model in Python using the scikit-learn library. The results were solid and reliable when checked with test data.

The app also came with a voice chatbot that could answer user questions in Sinhala or English. It used data pulled from SLTB's official schedules in PDF format. The chatbot understood natural language well, so riders could ask when a bus would arrive or where it was headed and get a quick answer. Plus, it worked for everyone—even people with visual impairments—thanks to speech-to-text and text-to-speech features.

Firebase wasn't just there for tracking. It also handled user accounts, app notifications, and chatbot responses. The front end was built in React Native so it could run on different devices, while the backend dealt with logic and ETA calculations.

Initial testing got great feedback. Riders liked the ETA accuracy and voice support, and the system built trust while reducing long wait times. It made public transit more inclusive, especially for people with disabilities, and marked a clear step up from how things were done before.

3.2.2 NFC and Blockchain-Based Ticketing System

This project focused on bringing Sri Lanka's public bus fare system into the modern world using NFC and blockchain tech. The older method, which mostly depended on cash, had a lot of issues—mistakes, people not paying, and slow boarding times. The new setup offered a smoother and faster way to pay: passengers could just tap their phone or card at the bus entrance to pay without digging for coins.

Every time someone paid using this system, the transaction was instantly checked and logged on a secure private blockchain. That made it nearly impossible for anyone to mess with the data. And to make sure nobody's personal info or trip details got exposed, everything was locked down using AES encryption. The system also made it easy to group passengers—like students, older folks, or daily commuters—and offer special discounts to them.

Under the hood, Ethereum smart contracts kept everything running fairly by handling transaction rules and checks. The app interface was built light and quick, so even on shaky internet, things worked smoothly. Firebase helped sync everything in real time, so drivers, passengers, and central operators were always in the loop.

To make it even more appealing, a rewards program was added. Riders earned points for every trip and could trade them for free rides or discounts. When tested on real buses, the new system led to quicker boarding, less pushing at the door, and happier passengers and drivers.

This kind of upgrade could be a turning point for public transport in Sri Lanka. Down the line, it could be linked with banks or national ID systems to grow and connect more services together.

3.2.3 IoT-Based Real-Time Monitoring System

This part of the smart bus system focused on solving two big problems: drivers not knowing if someone’s waiting at the next stop, and risky overtaking behavior among buses trying to get more passengers. To tackle both, a mix of low-cost devices was used—like ESP32 microcontrollers, GSM modules, GPS, and display screens. Together, they gave drivers a clearer picture of what was happening on the road and at upcoming stops.

Each bus had a GPS module hooked up to an ESP32 board that kept track of its location. That location was sent in real time to Firebase through cellular data, using simple HTTP requests. By comparing the position of different buses on the same route, the system figured out how close one bus was to another using Google's Distance Matrix API. Then, it sent that info back to the driver's screen, helping them stay informed and avoid crowding or unsafe passing.

A key safety feature let passengers at stops tap a button in a mobile app to let the driver know they were waiting. This alert popped up on the driver's screen too, helping avoid missed stops and improving overall coordination. No guesswork, no waving down buses last minute.

Inside the bus, a small ESP32-CAM module was set up to take photos showing how full the bus was. It sent those images via Bluetooth to the main board, which uploaded them to Firebase using the same GSM network. This helped operators keep an eye on how crowded buses were getting—without needing expensive equipment.

Field tests showed the system worked well in cities and quieter areas alike. Altogether, this IoT upgrade added much-needed awareness, communication, and safety to public transport in Sri Lanka.

3.2.4 Smart Bus Recommendation and Seat Detection System

This AI-driven piece of the smart bus system was built to fix one of the biggest annoyances in public transit: not knowing where to find a seat, or worse, ending up on a packed bus when another nearby one had space. It worked by combining real-time camera images with machine learning tools to help both passengers and drivers make better choices.

Tiny cameras, mounted on the ceiling, snapped images every so often. These were processed using OpenCV and deep learning to tell the difference between taken and empty seats even when the lighting wasn't great, or passengers were sitting in odd positions. The images were sent to Firebase, where they were analyzed almost immediately.

Behind the scenes, a Random Forest Classifier-a kind of machine learning model—used a bunch of factors like where someone was standing, how full each bus was, where they'd gone before, and arrival times to suggest the best bus to take. It didn't just throw out generic options learned from patterns and improved over time as more data came in.

Passengers could use a simple app to register, share their location, and see which buses had free seats. It was especially helpful for folks unfamiliar with the system like tourists—since the app had voice help and multiple languages built in. Drivers had their own screen too, showing how crowded their bus was and how best to handle pickups.

The system kept learning as people used it, tweaking its predictions so they got smarter every day. Data was stored and synced using Firebase and MongoDB, keeping everything updated in real time.

Tests showed it made a real difference—less waiting, happier riders, and buses that weren't all jammed at once. And since it's built to expand, this setup could help drive even bigger improvements in how cities move people around.

3.3 Discussion

3.3.1 Intelligent Bus Navigation and ETA Prediction System

This part of the smart bus system zeroed in on giving passengers accurate updates on where buses were and when they'd arrive. A lot of folks rely on that kind of info when trying to plan their day, but real-time tracking isn't always easy. One of the big hurdles? Keeping Firebase running smoothly while tons of buses sent constant updates. Even though Firebase made syncing easy, things got messy fast when updates poured in. So the team had to get smart with it—limiting how often data was sent and organizing it neatly in JSON trees to avoid clogging things up.

Another important piece was making sense of a mountain of unorganized route data from the Sri Lanka Transport Board. It came in all sorts of formats, so custom Python scripts were written to pull out useful stuff GPS points, time logs, distances between stops—and clean it all up. That cleaned data powered a basic but efficient linear regression model, built using scikit-learn. It didn't need much computing power, which made it perfect for phones and lightweight devices.

The app wasn't just smart, it was also inclusive. It came with a chatbot that could understand questions in Sinhala and English and talk back using speech features. This made it easier for everyone to get help, especially those with vision issues. The bot could hear questions, turn them into text, fetch ETA data, and then read the answer back out loud.

React Native was used to build the app so it could run on both Android and iPhones without extra work. Firebase acted as the bridge between users and the system in real time. Tests showed the arrival estimates were pretty spot on—usually just a minute or two off—and that built a lot of trust among riders. People stopped second-guessing if a bus was coming or not, which made the whole public transport experience a lot less stressful.

It's a setup that doesn't just work today—it's built to grow. With the right tweaks and more AI tools, it could get even sharper at predicting bus times and help even more routes in the future.

3.3.2 NFC and Blockchain-Based Ticketing System

This system was built to fix a messy problem—old-school bus fare collection. Cash-based payments caused all kinds of issues, from missed payments to fare cheating. So the team built something cleaner and smarter using NFC and blockchain tech. At first, Ethereum was considered, but its slow transaction speed and extra fees didn't fit the fast-moving world of public transport. Instead, they went with Hyperledger Fabric, a private blockchain that's quicker and doesn't charge fees every time someone taps to pay.

Writing smart contracts wasn't a walk in the park either. The logic had to be sharp enough to handle real payment situations and keep everything secure. So they ran a bunch of test simulations before putting anything live. On the hardware side, getting NFC readers to work smoothly on different bus models and with different phones was tricky too. That meant testing all kinds of tags and reader brands to find what worked best.

The system didn't just handle basic fares—it was smart enough to give automatic discounts to students, seniors, and regular riders. Loyalty points got tracked on the blockchain too, so passengers could earn free rides or discounts over time. To keep user info safe, everything was locked down using AES-256 encryption.

They also built backup plans. If the network dropped out mid-transaction, the system didn't crash. Instead, it saved the info locally and uploaded it once the connection came back. No lost payments, no need to redo anything. People really liked the faster boarding and instant digital receipts sent straight to their phones.

With Firebase in the mix, the app stayed synced with real-time data—updating account balances, ride history, and even system analytics without delay. In the end, it wasn't just a new way to pay it, it made the whole experience smoother, more secure, and a lot more trustworthy.

3.3.3 IoT-Based Smart Bus Monitoring System

This part of the system took on real problems from the ground up, blending low-cost hardware and cloud tools to make bus rides safer and smoother. In places with patchy mobile signals, like many areas in Sri Lanka, HTTP turned out to be more dependable. It handled spotty 3G networks better, with fewer connection drops and easier setup.

The ESP32 board on each bus collected GPS data every few seconds and sent it to Firebase using async HTTP requests to keep things running fast and smooth. A separate camera module, the ESP32-CAM, was installed to snap photos inside the bus. Since the camera and GPS module worked independently, they used Bluetooth to send pictures over to the main board. That wasn't easy at first, getting them to sync was tricky. But with some timed delays and sending the image into small parts, they got it to work reliably, even with limited power.

One clever feature was the system that told drivers how close they were to the next bus on the route. It worked by collecting GPS data from all buses and using the Google Distance Matrix API on the backend to figure out spacing. This info was pushed back to each bus and shown on a screen, which helped cut down on aggressive overtaking.

They also added a feature in the passenger app where people could let the driver know they were waiting at a stop. That alert showed up on the driver's display, helping avoid missed pickups and making the whole ride feel more connected.

The setup was flexible too. It's ready to support more upgrades, like auto stop announcements or voice alerts. During trials, the whole system ran solid—even across areas with flaky coverage. And because it uses widely available tech like GSM, Firebase, and HTTP, it's set up to roll out across the country without needing a major infrastructure overhaul.

3.3.4 AI-Based Smart Bus System

This part of the smart bus project took aim at two nagging problems: not knowing where the empty seats were, and figuring out which bus was actually the best one to hop on. They tackled both with AI, machine learning, and a sharp eye on real-world conditions.

To get seat tracking right, the team had to deal with a lot of messy variables—bad lighting, odd camera angles, passengers half-blocking seats. They used the ESP32-CAM module to take images regularly and ran those through a preprocessing pipeline grayscale filter, noise cleanup, and edge detection—before trying to spot whether seats were taken or not. After testing different approaches, a combo of OpenCV and a basic neural network gave them just enough accuracy to get the job done in real time. Once processed, those images were sent to Firebase Storage every few minutes.

At the same time, they built a recommendation system to help riders pick the best bus. It used a Random Forest Classifier trained on past data like where people boarded, which routes they liked, how full the buses were, and their usual travel habits. This model handled both numbers and categories well and kept learning over time, adapting to how people actually traveled.

The user app made everything feel personal. Passengers could sign up, add their preferences, and get real-time updates. It worked in multiple languages and included voice commands, which was a big help for seniors and visitors unfamiliar with the system. Drivers had their own tool, a screen that showed how packed the bus was and which stops were coming up, so they could plan better and avoid delays.

Behind the scenes, Firebase and MongoDB worked together to keep things flowing. Firebase dealt with real-time updates, like showing available seats or sending alerts. MongoDB stored the bulk of the long-term data used to train the model and tweak predictions.

When tested on actual routes, the system proved it wasn't just smart, it was practical. Riders spent less time waiting, buses weren't overcrowded, and everything ran a little smoother. A good example of how everyday problems can be solved with the right mix of tech and insight.

3.4 Summary of Each Student's contribution

Member	Contribution
Senarathne A.S.A.G (IT21205460)	• Designed and implemented the NFC-based bus fare payment system. • Integrated blockchain for secure, transparent transaction logging. • Enabled real-time confirmation of payments to conductors. • Developed passenger categorization and loyalty point systems. • Handled backend development using Django framework.
Kalhara N.V. (IT21202490)	• Built a machine learning model (Random Forest) for ETA prediction. • Integrated historical and real-time data for accurate predictions. • Developed voice interaction features for visually impaired passengers. • Created a chatbot for real-time bus route and schedule inquiries. • Implemented mobile frontend with React Native.
Rajapaksha R.L.H.P. (IT21192050)	• Developed the hardware module using ESP32 with GPS and GSM. • Established reliable internet connectivity for real-time data transfer. • Implemented Firebase Realtime Database for IoT data storage.

	• Displayed live bus data on onboard LCD for passengers. • Tested and deployed code via AWS EC2 for scalability.
Thanusan M. (IT21175848)	• Developed camera-based system for bus number and text detection. • Implemented OCR and multilingual translation for better accessibility. • Integrated alerts for unauthorized vehicles and important stations. • Created AI-powered seat availability monitoring system. • Added personalized recommendations based on travel preferences.

Table 3. 1: Summary of Each Student's contribution

4. CONCLUSION

This project didn't just upgrade buses with new tech it reimagined how public transport could actually work better for people across Sri Lanka. By pulling together smart tools in a thoughtful, modular setup, the system tackled problems that riders, drivers, and transport planners have faced for years. It's a leap toward a future where public transport runs smoother, feels safer, and welcomes everyone.

The first piece, the Intelligent Bus Navigation and ETA Prediction System, helped take the guesswork out of catching a bus. Using GPS data and a lightweight machine learning model, the system gave riders reliable estimates for when the next bus would arrive. Voice commands and an AI chatbot made it easier for visually impaired users to interact with the system, turning a basic update feature into something inclusive and empowering.

Then came the NFC and blockchain fare system, which cut down on lines and manual errors. Just tap to pay. It was quick, contactless, and secure thanks to AES encryption and blockchain keeping all transactions tamper-proof. It didn't just record payments, it also handled passenger types and loyalty points, turning a boring payment process into a smart, data-driven experience. The pilot phase showed it worked well, and it's clearly ready to scale.

On the safety side, the IoT Monitoring System used ESP32 boards, GPS, and GSM modules to keep track of bus positions and prevent risky behavior like overtaking. A small screen on board helped drivers know how far the next bus was. At the same time, a simple check-in button in the app let passengers alert drivers that they were waiting less chance of being left behind. And the onboard camera sent snapshots of how crowded the bus was, giving real-time data that could be used to plan future trips better.

The last piece brought in computer vision and machine learning to recommend the right bus based on real-time crowd levels, route patterns, and personal preferences. Using OpenCV and Random Forests, the system could even tell which seats were free. With updates pushed in real time, multilingual support, and dashboards for drivers, this part turned passive commuting into an active, informed experience.

Together, all four parties created a unified, low-cost system that can work in cities and rural areas alike. It didn't leave anyone out supporting tourists, seniors, and differently abled passengers equally. And even though a few bumps came up like weak GPS signals or dim lighting for seat detection the modular setup means these can be improved one step at a time, without overhauling the whole thing.

Looking forward, there's a lot of room to grow. From licensing tech to other cities, offering advanced features for frequent riders, or teaming up with mobile carriers and local governments, the path to commercialization is clear. This kind of system could even serve as a model for other countries facing similar public transit problems.

At its core, this wasn't just about buses or routes. It was about how technology if built right can adapt to real life, make every day routines easier, and open doors for people who've often been left out of digital systems. A true example of smart solutions made human.

5. REFERENCES

[1] N. Nishanthan, K. Thiruthanigesan, and P. Georgakis, "Towards the Development of Intelligent Transportation Systems in Sri Lanka," *International Journal of Scientific & Technology Research*, vol. 6, no. 12, pp. 11–15, Dec. 2017.

[2] S. Iyer, "How Smart Public Transportation Is Reshaping Urban Commuting," *Softweb Solutions*, 2021. [Online]. Available: https://www.softwebsolutions.com/resources/smart-public-transportation-system.html [Accessed: Mar. 29, 2025].

[3] United Nations ESCAP, "Integration and Application of Digital Technologies in Public Transport in Sri Lanka," *ESCAP*, 2024. [Online]. Available: https://www.unescap.org/sites/default/files/event-docs/Sri%20Lanka_Digital%20Transport_ESCAP.pdf

[4] N. Nishanthan, K. Thiruthanigesan, and P. Georgakis, "Towards the Development of Intelligent Transportation Systems in Sri Lanka," *Int. J. Sci. Technol. Res.*, vol. 6, no. 12, pp. 11–15, 2017.

[5] United Nations ESCAP, "Integration and Application of Digital Technologies in Public Transport in Sri Lanka," *ESCAP*, 2024. [Online]. Available: https://www.unescap.org/sites/default/files/event-docs/Sri%20Lanka_Digital%20Transport_ESCAP.pdf

[6] A. Bessa et al., "RioBusData: Outlier Detection in Bus Routes of Rio de Janeiro," *arXiv preprint*, arXiv:1601.06128, Jan. 2016.

[7] D. Balasingham et al., "Enhancing Train Transportation in Sri Lanka: A Smart IoT Based Multi-Subsystem Approach Using MQTT," *arXiv preprint*, arXiv:2501.08229, Jan. 2025.

[8] M. F. M. A. Hakeem, N. A. Sulaiman, M. Kassim and N. M. Isa, "IoT Bus Monitoring System via Mobile Application," 2022 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), Shah Alam, Malaysia, 2022, pp. 125-130, doi: 10.1109/I2CACIS54679.2022.9815268. keywords: {Schedules;Databases;Urban

areas;Prototypes;Mobile applications;Internet of Things;Monitoring;Bus Monitoring System;IoT;GPS;mobile apps;Wi-Fi},

[9] M. Dhule, "NFC Based Smart Urban Public Bus Transport Payment System," 2018 3rd International Conference for Convergence in Technology (I2CT), Pune, India, 2018, pp. 1-4, doi: 10.1109/I2CT.2018.8529810. keywords: {Internet of Things;Radiofrequency identification;Intelligent sensors;Global Positioning System;Sociology;Statistics;Transportation;Public Bus;E-ticketing;NFC;Mobile App;Cheat-Proof},

[10] P. V. Crisgar, P. R. Wijaya, M. D. F. Pakpahan, E. Y. Syamsuddin and M. O. Hasanuddin, "GPS-Based Vehicle Tracking and Theft Detection Systems using Google Cloud IoT Core & Firebase," 2021 International Symposium on Electronics and Smart Devices (ISESD), Bandung, Indonesia, 2021, pp. 1-6, doi: 10.1109/ISESD53023.2021.9501928. keywords: {Cloud computing;Tracking;User interfaces;Internet;Automobiles;Telemetry;Smart devices;Internet of Things;Vehicle Tracking System;Theft Detection;Google Cloud IoT Core;Firebase},

[11] V. Kumar, "Vehicle Number Plate Detection System Using Image Processing Approach," ResearchGate, Accessed: Aug. 12, 2024. [Online]. Available: https://www.researchgate.net/publication/375883893_Vehicle_Number_Plate_Detection_System_Using_Image_Processing_Approach#:~:text=The%20number%20plate%20region%20in,contains%20the%20specific%20vehicle%20information

[12] D. Li, W. E. Wong, M. Chau, S. Pan and L. S. Koh, "A Survey ofNFC Mobile Payment: Challenges and Solutions using Blockchain and Cryptocurrencies," 2020 7th International Conference on Dependable Systems and Their Applications (DSA), Xi'an, China, 2020, pp. 69-77, doi: 10.1109/DSA51864.2020.00018. keywords: {Privacy;Protocols;Online banking;Blockchain;Bitcoin;Cryptography;Internet of Things;NFC mobile payment;blockchain;cryptocurrency;fair payment;security;privacy},

[13] S. Vaishnavi, G. Renish, T. Surendra, J. R. Kumar and V. Srinivasan, "Advancements in Public Transport: Design and Implementation of an Android-Based Real-Time Bus Tracking System," 2023 IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA), Hamburg, Germany, 2023, pp. 41-46, doi: 10.1109/ICCCMLA58983.2023.10346218. keywords: {Schedules;Costs;Web services;Smart cities;Urban areas;Real-time systems;Timing;Smart Bus;Digital Technology;Google Maps API;Smart Cities},

[14] H. D. Weligamage, S. M. Wijesekara, M. D. S. Chathwara, H. G. Isuru Kavinda, N. Amarasena and N. Gamage, "An Approach of Enhancing the Quality of Public Transportation Service in Sri Lanka using IoT," 2022 IEEE 13th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 2022, pp. 0311-0316, doi: 10.1109/IEMCON56893.2022.9946624. keywords: {Smart cards;Visualization;Systematics;Face recognition;Sociology;Mobile communication;Real-time systems;RFID;GPS;IoT;Image Processing;Arrival time Prediction},

6. APPENDICES

Paper Title	Uploaded	Grade	Similarity
final REPORT GROUP.docx	04/12/2025 12:56 AM	--	4%