Sri Lanka Intelligent Bus Navigation and Passenger Information System

Thanusan M.
IT21175848

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

Department of Information Technology

Sri Lankan Institute of Information Technology
Sri Lanka

August 2024

Sri Lanka Intelligent Bus Navigation and Passenger

Information System

Thanusan M.

IT21175848

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

Specialization in Information Technology

Department of Information Technology

Sri Lankan Institute of Information Technology

Sri Lanka

August 2024

DECLARATION

I hereby declare that this is my own work and that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university or other institution of higher learning. To the best of my knowledge and belief, it does not contain any material previously published or written by another person, except where acknowledgements are made in the text.

I also grant the Sri Lanka Institute of Information Technology (SLIIT) the non exclusive right to reproduce and distribute my dissertation, in whole or in part, in print, electronic, or any other medium. I retain the right to use this content in whole or part for future works, such as publications, journal articles, or books.

Name	**Student ID**	**Signature**
Thanusan M.	IT21175848	

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

Signature of the Supervisor　　　　　　　　　　Date

(Dr Shanika Wijesekara)

Abstract

Public transportation is a vital component of urban mobility, yet inefficiencies such as unpredictable bus arrivals, overcrowding, and lack of real-time seat availability impact commuter experience. This research proposes a smart bus system leveraging artificial intelligence (AI) and machine learning (ML) to enhance efficiency and convenience. The system integrates real-time seat availability detection using OpenCV, a bus recommendation engine utilizing a Random Forest Classifier for predictive modelling, and a user-friendly registration module for locals and foreigners. Additionally, it provides real-time notifications, a bus driver interface for monitoring occupancy, and passenger location sharing for route optimization. With incremental learning techniques, the system continuously improves recommendations based on evolving travel patterns. The implementation of this smart bus system aims to reduce waiting times, optimize bus utilization, and improve the overall public transportation experience.

Keywords: *Smart Bus System*, *Artificial Intelligence*, *Machine Learning*, *OpenCV*, *Real-time Notifications*

Acknowledgement

Most importantly, I would like to express my deepest gratitude to my supervisor, Dr. Shanika Wijesekara, for her constant support, invaluable guidance, and constructive criticism during the research and development process of this project. Her guidance and encouragement were instrumental in shaping my work into what it is today. I would also like to truly thank my co-supervisor, Dr. Kapila Dissanayake, for his insightful advice and mentorship, which contributed immensely in depth and direction to the project. My greatest appreciation to the Faculty and Staff of the Faculty of Computing of the Sri Lanka Institute of Information Technology (SLIIT) for providing a solid academic grounding and learning culture for innovation and critical thinking. I am especially grateful to my project group members Senarathne A.S.A.G, Kalhara N.V, and Rajapaksha R.L.H.P for their cooperation, dedication, and unwavering support throughout group work and collaborative activities. Their contribution immensely helped the success of this study. I would also like to thank all individuals who provided technical assistance, particularly during data collection, testing, and implementation. Your practical knowledge and willingness to help were worth a lot. Last but not least, I am very thankful to my family and friends for their constant encouragement, patience, and support throughout. Their moral and emotional support were essential for the successful completion of this project.

Table of Contents

List of Figures

List of tables

1. Introduction

1.1 Background

Urban transportation remains one of the key components of infrastructure in contemporary cities. In most developing nations, such as Sri Lanka, buses form the primary mass transit mode of public transport as a result of its affordability and accessibility. While extensively employed, public transit systems are plagued by a chain of efficiency imperfections such as over-crowding, irregularity in the schedule, inaccessibility to seat details, language limitations to passengers, and minimal provision for real-time data. Such inefficiencies have severely minimized commuter quality experience and are a discouragement to wide adoption of public transit systems.

In the past few years, advances in Artificial Intelligence (AI), Machine Learning (ML), and the Internet of Things (IoT) have transformed urban mobility solution planning and implementation. Cities worldwide have started adopting smart transportation systems with real-time data and predictive analytics to improve public transport systems. Singapore, South Korea, and Germany are exemplary instances of how AI-powered transit systems can improve scheduling precision, commuters' satisfaction, and efficiency. Such transit systems utilize a blend of GPS tracking, predictive analytics, real-time alerts, and computer vision to manage fleets and alert passengers with great accuracy.

Country	Features Implemented	Benefits Observed
Singapore	Real-time bus ETA, mobile seat booking	Reduced waiting time by 30%
South Korea	Multilingual support, AI-based traffic control	45% higher tourist satisfaction
Germany	Predictive scheduling, NFC ticketing	Increased public transport usage
Sri Lanka	Basic GPS (some buses only)	Limited reach and integration

Table 1.0: Smart Public Transportation Features and Benefits by Country

On the other hand, Sri Lanka's public transport system is still more or less analogy in character. Although efforts have been made to introduce GPS-based tracking in some bus fleets, the inability to integrate technologies and absence of centralized data platforms hinder the effectiveness of such initiatives. An integrated, AI-led solution is a pressing necessity that not only modernizes the public transport network but also enhances the experience of both local commuters and international visitors.

1.1.1 Importance of Digital Mobility

Digital mobility refers to the integration of digital technologies in transport systems for enhancing efficiency, accessibility, and the overall experience of the journey. With the advent of the smart city, digital mobility has become an imperative instrument for modernizing public transport services and urban sustainability. By means of digital technologies such as mobile applications, cloud computing, GPS tracking, and data-driven decision-making, cities around the world are transforming the experience of commuting. One of digital mobility's biggest strengths is that it can close the gap between ancient transport infrastructure and today's tech-savvy customers' expectations. It delivers dynamic routing suggestions, real-time seat tracking, intelligent scheduling, and mobile tickets — all tailored to the individual commuter need. Especially in congested urban areas with frequent delays and overcrowding, these technologies allow both operators and travelers to make faster and more rational travel decisions.

For Sri Lanka, adopting digital mobility is not just an upgrade; it is a necessity. With the growing urban population and a flourishing tourism sector, there is an ever-growing need for reliable, efficient, and user-friendly transport options. AI-powered features like real-time seat detection, predictive route recommendations, and location-based alerts can enhance both local and tourist travel experiences. If implemented, these technologies not only improve the quality of services but also reduce traffic congestion, enhance safety, and make the transport environment more inclusive and connected.

1.1.2 Transportation Optimization Role of AI and ML

Artificial Intelligence (AI) and Machine Learning (ML) are now at the forefront of turning traditional transport networks into smart, adaptive networks. AI allows systems to analyze data, make predictions, and make decisions independently — abilities extremely useful in managing complex public transportation systems. AI is revolutionizing how people move around cities from route optimization and travel time prediction to automated alerts and user behavior monitoring.

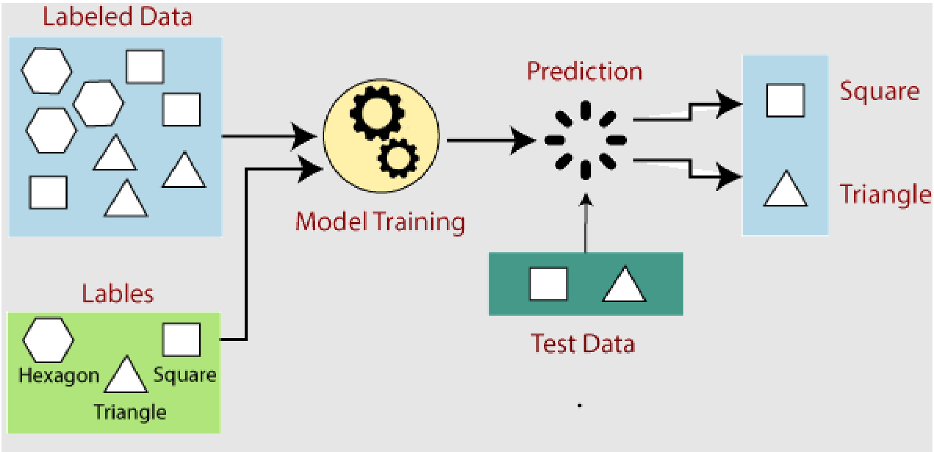

Figure 1: Supervised Machine Learning Workflow for Shape Classification

Machine Learning, which is a subset of AI, is excellent at hunting patterns within huge datasets. In public transportation, ML algorithms predict bus arrival times, monitor crowd sizes, give individualized route recommendations, and identify peak travel hours. They learn from past experience and become more accurate as they are presented with more real-time scenarios. For example, ML can re-route suggestions dynamically if a specific bus is too full or late.

The use of AI and ML for transport optimization is not simply a matter of efficiency. These technologies enable inclusive design, where individual user profiles can adapt the transport experience to meet user needs — whether it's providing travel notices in multiple languages or providing alternative routes for people with disabilities. In Sri Lanka, incorporating AI and ML into public transport can lead to smart scheduling, enhanced customer service, reduced congestion, and cleaner city mobility

1.2 Literature Review

Numerous studies have highlighted the impact of AI and ML in transforming public transportation systems. In a study conducted in Singapore, real-time bus tracking integrated with predictive analytics significantly reduced commuter wait times by 30%. Another research in India implemented a computer vision-based seat monitoring system using OpenCV and TensorFlow, which achieved over 90% accuracy in real-time detection.

Additionally, research has shown that integrating NFC with blockchain can eliminate common issues found in cash-based and conventional e-ticketing systems. The blockchain ensures transaction immutability and traceability, which builds trust among users and operators.

A notable implementation in Seoul, South Korea, uses multilingual mobile apps combined with GPS-based bus arrival prediction, which has improved user satisfaction among international tourists by 45% over two years. These success stories underline the potential of deploying a similar, context-adapted system in Sri Lanka.

1.3 Justification

This research is justified on multiple fronts. Technologically, it leverages readily available tools like OpenCV, TensorFlow, and cloud-based APIs to build a cost-effective, scalable solution. Socially, it aims to bridge the digital divide in public services by making transport systems accessible to everyone, including non-native speakers and the differently-abled. Economically, a well-functioning public transport system can reduce traffic congestion, lower environmental impact, and increase overall productivity by reducing commute times.

The system also aligns with several Sustainable Development Goals (SDGs), particularly SDG 11: Sustainable Cities and Communities. By promoting the use of intelligent public transport, the research contributes to environmental conservation, economic efficiency, and social inclusiveness.

1.4 Significance of the Study

The increasing urban population and vehicle density in Sri Lanka necessitate the need for efficient public transportation solutions. The traffic congestion in cities such as Colombo, Kandy, and Galle is exacerbated by the lack of coordination in public bus services. A well-implemented Smart Bus System can alleviate these issues by optimizing bus routes, improving real-time communication with commuters, and encouraging more people to choose public transportation over private vehicles.

Moreover, tourists visiting Sri Lanka often encounter significant difficulties navigating the local bus system. Language barriers, unfamiliarity with routes, and lack of seat reservation capabilities make public buses an unviable option for many. Introducing features such as multilingual interfaces, tourist destination alerts, and seat reservation through a mobile app can dramatically improve the experience for foreign passengers, thereby enhancing the country's image as a tourist-friendly destination.

Issue	**Impact on Passengers**
No seat availability info	Overcrowding, discomfort
Inaccurate bus arrival times	Missed appointments, waiting at stops
Language barriers	Confusion for tourists
Poor passenger-driver comms	No visibility into passenger loads or needs
Manual ticketing	Delays, cash disputes, unsafe handling of money

Table 2: Common Issues in Public Bus Systems and Their Impact on Passengers

1.5 Research Gap and Novelty

Current public transport systems, particularly in developing nations like Sri Lanka, mostly employ basic digital features such as static schedule information or low-level notification services. While some systems have adopted GPS tracking or basic mobile alerts, these features are not sufficient in meeting the complexities of real-time demands from frequent commuters and tourists alike. Current platforms often lack richness in AI integration required for system-wide optimization. None of these platforms offer dynamic seat monitoring, traveler-centric travel prediction, or personalized recommendations, making them overall less helpful. Furthermore, such systems are not highly adaptable to personal tastes or evolving commuter behavior over a period of time.

The novelty of this work lies in its comprehensive, AI-driven system that includes advanced modules like computer vision to identify available seats and machine learning to suggest routes ahead of time. Unlike existing systems with generic alerts or static maps, this project uses OpenCV to analyze seat occupation in real time through image processing. The system also incorporates a personalized route engine based on Random Forest Classifier algorithms, taking into account various parameters such as user history, bus occupancy, time preference, and proximity of locations. Such a machine-learning-based architecture provides tremendous flexibility in accommodating an exceptionally adaptable transport network catering to daily run-of-the-mill locals and occasional overseas travelers with the same efficiency.

The central system innovation here lies in increasing prediction accuracy via Random Forest algorithms. The standard schedule applications function upon static or average assumptions that inherently lead to ineffective or belated recommendations. To contrast, the Random Forest model improves the quality of decisions by combining the results from a number of decision trees that have been trained on diverse real-world inputs such as traffic patterns, boarding levels, and route requests. Using an ensemble learning approach, it ensures robust performance even when partial data inputs are missing or noisy. Consequently, the system provides better-quality travel recommendations, reduces unnecessary waiting times, and offers more accurate estimates of seat availability and travel paths. By integrating AI with real-time environmental information, the system addresses a major technological shortfall in existing transport provision and offers a scalable model for national roll-out.

Figure 2: Smart Bus In-Vehicle Applications and Connectivity Features

2. Research Problem

Sri Lanka's public transport experience is marred by its big inefficiencies like the non-existence of actual seat numbers, inaccurate bus arrival predictions, incoherent communications between drivers and commuters, and minimal personalization to cater to different users. These deficits bring down commuter happiness and discourage the use of public transport. The situation calls for a smart AI-based system that can overcome such challenges and remake the public transport experience.

3. Objectives

3.1 Main Objective:

The principal aim of the present research is to create and implement a Smart Bus System in Sri Lanka, based on Artificial Intelligence (AI) and Machine Learning (ML) technologies, with the goal of increasing the efficiency, safety, and user convenience of public transport in general. The proposed system aims to address extremely important problems of overloading, lack of real-time information, poor tourist accessibility, and outdated payment systems. By creating an individualized, smart, and responsive bus transport system, the system will attempt to bring Sri Lanka's public transportation system up to digital mobility standards of today.

3.2 Sub Objectives:

1. Creating a computer vision model to identify real-time seat occupancy with the help of OpenCV.

The above goal aims at addressing the issue of overcrowded buses by informing real-time seat occupancy. Using OpenCV, a powerful computer vision library, the system captures live images from onboard cameras and processes them to detect empty seats. A visual marker (e.g., a \"+\" sign) on empty seats is detected by a template-matching algorithm. The data is processed and transmitted to the mobile app, where passengers can view available seats before they board. This not only puts commuters at ease but also permits more intelligent boarding decisions, reducing traffic and improving travel experience especially during peak times.

2. To design a machine learning-based bus recommendation system with the use of Random Forest and LSTM algorithms.

The purpose of this aim is to render bus suggestions customized based on user behavior, history, and real-time conditions. A Random Forest Classifier is used for suggesting the best bus based on analysis of factors such as travel time, current occupancy of seats, popularity of a route, and past user histories. Meanwhile, an LSTM model analyzes time-series data to give predictions of journey trends and accustomed user movement. Together, these models ensure that each user receives the best and timely bus recommendation that does not cause unnecessary waiting times and maximizes overall commuter satisfaction.

3. To develop a user registration and profiling mechanism for local as well as foreign users.

This objective ensures that the platform caters to a wide population of users by having personalized user profiles. Local commuters are able to save repetitive routes, language options, and travel history. Foreign users can, however, register in their preferred language and view travel tips with translated routes and tourist-focused assistance. The user information is stored safely by the system with MongoDB and includes inherent role-based access (local vs. tourist) such that the interface and services adapt accordingly. This feature enhances accessibility and usability for everyone and facilitates public transportation.

4. To offer real-time bus arrival, key stop, and seat availability alerts through GPS and cloud messaging services.

This aim is based on real-time customer communication through GPS tracking and Firebase Cloud Messaging (FCM) integration. The application tracks the live location of every bus and estimates the Estimated Time of Arrival (ETA) of each bus. Push notifications are provided to users for nearby buses, upcoming stops, and real-time seat availability. This forward-looking notification system keeps commuters from missing their buses and also allows them to schedule their trip well. It reduces uncertainty and also provides users with live transit information, making the traveling process smooth.

5. To develop a bus driver interface for viewing live passenger information and route directions.

This objective enhances the communication and decision-making power of bus drivers by providing them with a separate driver interface. The in-app user interface, built with React Native, includes real-time seat fill rates, expected passenger numbers at upcoming stops, and boarding pattern notifications for drivers. This allows drivers to more effectively manage passenger flow, avoid overloads, and enable smoother boarding processes for themselves. Drivers can also be alerted about filled stops or detours so that they can make optimal driving decisions and provide greater service reliability for passengers.

6. In depth support for passenger sharing of locations in order to plan optimal routes.

This objective resolves path optimization and better service planning by providing an ability for passengers to share their current location with the system. Upon the user specifying their destination and sharing coordinates, the backend will decide on the most suitable paths and calculate pickup points dynamically. The GPS data is encrypted and used only within travel sessions in order to keep it protected against privacy invasion. By observing the real-time distribution of users from various regions, the system is able to optimize overall bus route planning and avoid unnecessary detours and delays, ultimately leading to a more efficient transportation system.

7. In order to implement incremental learning methodologies for ongoing optimization of the recommendation engine.

This goal improves the system's ability to adapt through the implementation of incremental learning methodologies that enable the ML models to update continuously without retraining them from scratch. With increasing users using the system over time, new data are introduced about bus usage trends, favored routes, and travel patterns. The LSTM and Random Forest models are updated repeatedly to reflect this shifting behaviour so that they make more accurate and appropriate predictions. This adaptive learning aspect implies the system becomes smarter and more responsive with use, learning to adjust to shifting commuter behavior and seasonal variations.

Structure of the Document

The rest of this document is organized into several chapters. Chapter 2 presents the research methodology, including the tools, techniques, and implementation strategies used. Chapter 3 discusses the system design and architectural components. Chapter 4 elaborates on the testing strategies and results obtained. Chapter 5 includes a detailed discussion on the implications of the results and the challenges encountered. Chapter 6 concludes the document with a summary of findings and outlines potential directions for future research.

In addition, appendices provide supplementary materials such as code samples, survey data, screenshots of the application, and diagrams illustrating system architecture and user workflows.

By structuring the report in this manner, the document aims to provide a comprehensive, coherent, and technically grounded account of the Smart Bus System development and its potential impact on public transportation in Sri Lanka.

4. Methodology

Overview of the technical Used

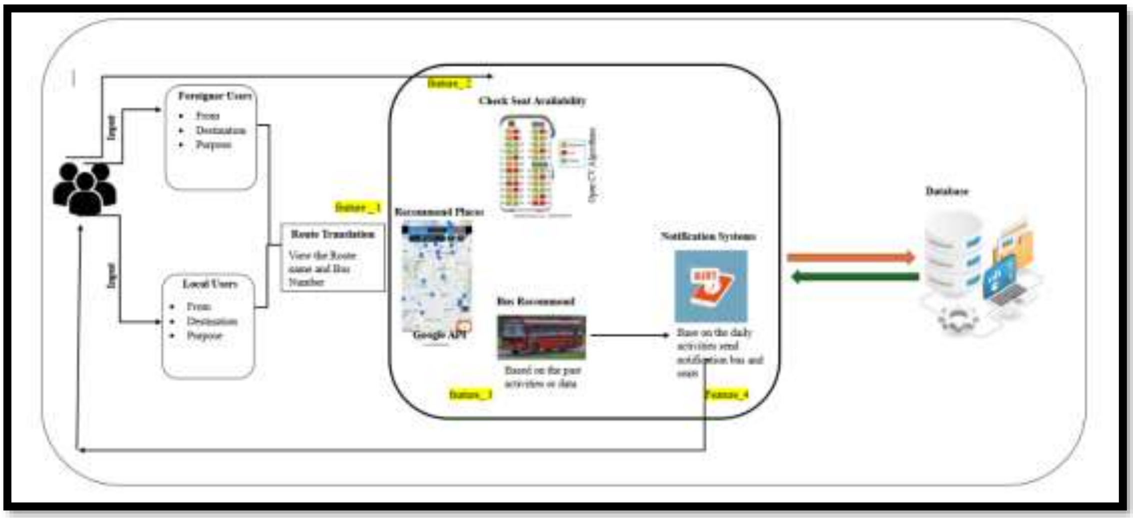

Figure 3: Overall system diagram

The Smart Bus System development methodology consists of multiple technical areas like computer vision, machine learning, mobile application development, and cloud-based integration. It was divided into systematic phases to have a well-structured development and modular testing at each level.

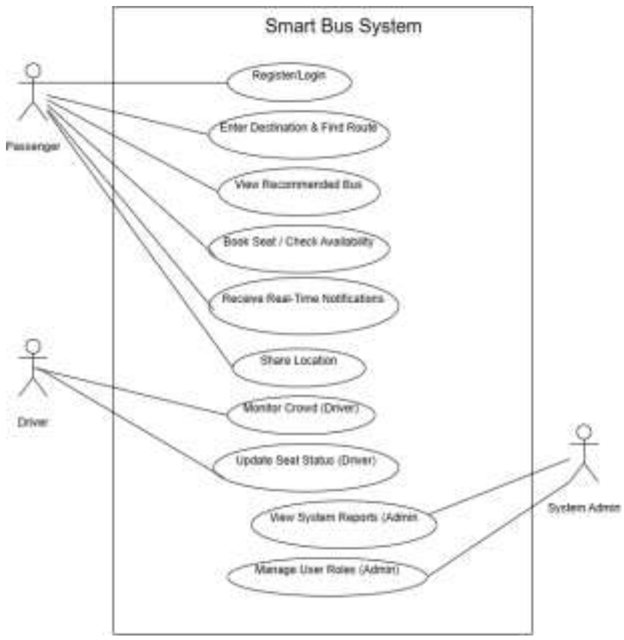

Figure 4: System use case diagram

4.1 Problem Statement

Public transportation plays a crucial role in the mobility and daily commuting of Sri Lankan people. However, the existing bus system, which is the primary mode of transport for the majority of the public, contains numerous operational and service inefficiencies. These not only impact user satisfaction but also discourage higher public utilization, thereby encouraging reliance on private vehicles and adding to urban traffic congestion. One of the most pressing issues is the lack of real-time data on the availability of seats. Commuters often board buses without prior knowledge of whether there are vacant seats, leading to congestion, discomfort, and occasional delays. This is especially frustrating during rush hour when there is peak passenger movement. Lack of transparency in seating arrangements disproportionately affects vulnerable groups such as the elderly, differently abled, and pregnant women, who require reserved or available seating space. The second vital shortfall is the inaccurate prediction of bus arrival times. A vast majority of buses operate without synchronized tracking systems, which results in unscheduled service patterns. Passengers are left waiting at stops without adequate information on when the next bus is expected to arrive. This unpredictability leads to productivity loss, missed schedules, and general lack of confidence in the public transport system. Public buses in Sri Lanka are much behind global standards when it comes to adopting GPS-based or AI-powered solutions for predicting arrivals.

In addition, the system does not support personalized route recommendation. Existing infrastructure does not track user travel patterns or user preferences to generate customized travel recommendations. This is especially frustrating for frequent commuters, tourists, or new users who are not familiar with complicated bus routes, stops, and destinations. International tourists are even more troubled by the lack of translation support and language barriers, which render public buses an inaccessible and confusing mode of transportation for them. Lack of communication between passengers and bus operators also contributes to inefficiency. The conductors and drivers lack insight into passenger boarding patterns, real-time passenger loads, or the number of passengers waiting at upcoming stops. The absence of real-time communication limits their ability to manage operations effectively and reduces the overall responsiveness of the system. Lastly, the traditional cash-based ticketing system is antiquated, laborious, and prone to dispute. It contributes to boarding time, creates congestion at bus doors, and puts conductors at risk of safety as they handle significant sums of money on a daily basis. Such practices are inefficient in an era when digital, contactless transactions have become prevalent. Additionally, without a digital record, there can be no transaction history to monitor or loyalty-based rewards to provide. Based on these concerns, there is a clear need for a new-age, technology-driven Smart Bus System. The system needs to be incorporated with artificial intelligence, machine learning, computer vision, and blockchain technologies for real-time seat detection, personalized bus recommendation, multilingualism, driver-passenger communication modules, and secure digital payments. Addressing these core issues will not only deliver commuter satisfaction but also contribute to the creation of a more sustainable and smarter public transportation system in Sri Lanka.

4.2 Data Collection and Preparation

The first phase consisted of the gathering of historic and real-time bus data. Historical data consisting of typical travel patterns, route maps, and peak hour analysis was collected from public transport authorities. Real-time data consisting of bus location, occupancy levels, and GPS coordinates were captured using mobile GPS modules. The data was preprocessed, cleaned, and labeled for training the machine learning models. To further strengthen the computer vision system, multiple images of bus interiors were captured under varying lighting and seating conditions.

4.3 Seat Detection using OpenCV

Computer vision features were implemented using OpenCV to detect empty seats in the bus. Template matching and edge detection techniques were implemented for identifying patterns of filled and empty seats. The app captures images from cameras installed onboard, grabs frames, and processes them in real time. Then the results are transmitted to the cloud server, and the mobile app dashboard is refreshed with real-time seat availability.

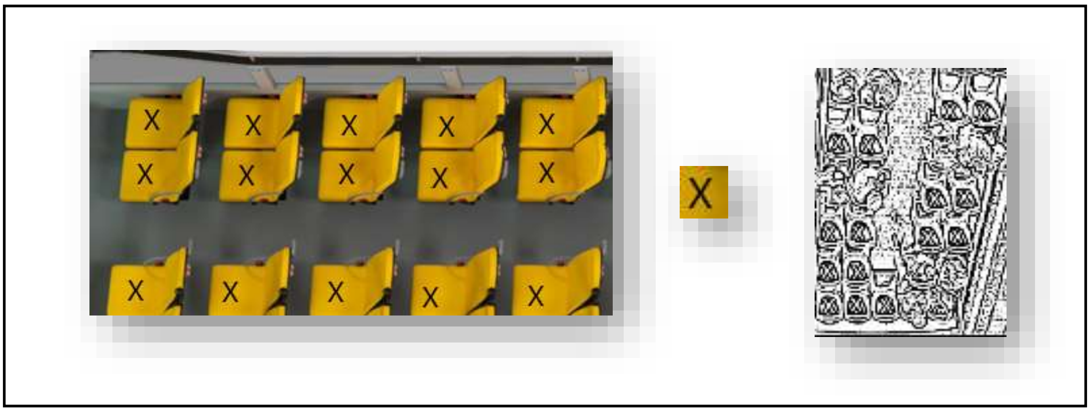

Figure 5: Seat detection using open cv algorithms

Bus Recommendation Engine

The app uses two machine learning models: Random Forest Classifier for recommending popular users and LSTM (Long Short-Term Memory) for time-series-based personalized predictions. The Random Forest model was trained on features such as travel time, congestion level of bus, and historical usage patterns. LSTM was used to predict best buses for habitual users based on their usage patterns. Both the models were trained and tested using Python-based libraries such as Scikit-learn and TensorFlow.

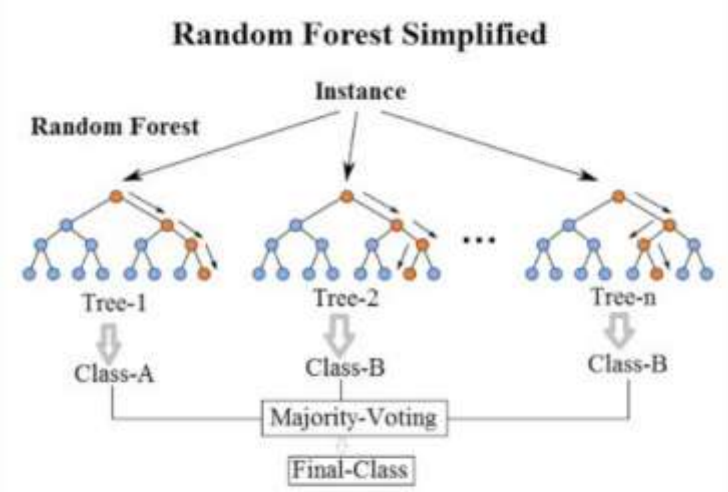

Figure 6: Random Forest Classification Process Using Majority Voting

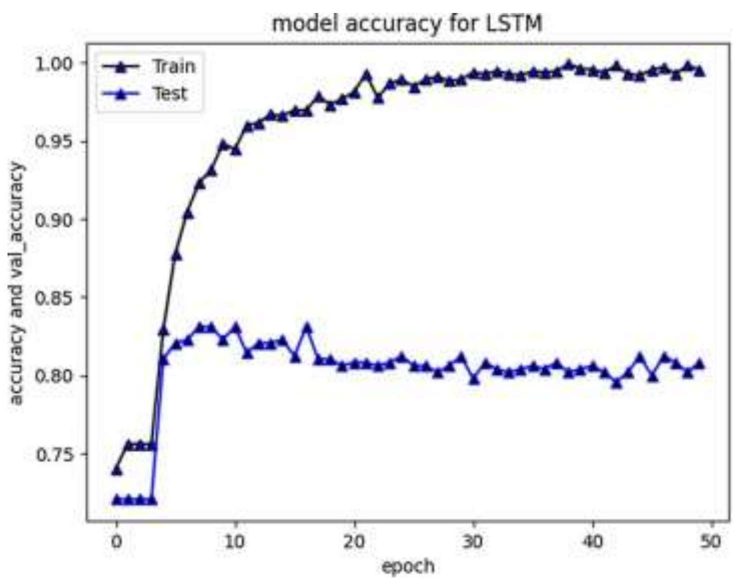

Figure 7: Model Accuracy Over Epochs for LSTM (Training vs. Testing)

Metric	Random Forest Classifier	LSTM (Long Short-Term Memory)
Accuracy	85%	88%
Precision	83%	86%
Recall	85%	89%
F1-Score	84%	88%
Training Time	Fast (few seconds)	Longer (multiple epochs)
Interpretability	High (easy to explain)	Medium (complex to interpret)
Use Case Fit	Good for classification tasks using static features	Excellent for time-based predictions and learning user behavior
Scalability	High	Medium (resource-intensive

Table 3: Performance Comparison Between Random Forest and LSTM Models

Both Random Forest Classifier and LSTM (Long Short-Term Memory) models were experimented with in the Smart Bus System recommendation engine design to determine whether or not they could predict the ideal bus choices for passengers. Each model possessed unique strengths that served different parts of the system's goals.

Random Forest Classifier had 85% accuracy, precision of 83%, recall of 85%, and an F1-score of 84%. This model is best applied to classification issues where there are fixed attributes such as bus capacity, day or night, and travel route history. Its biggest strength is its train speed, which is just seconds when learning from formatted datasets. Another strength is that Random Forests are extremely interpretable, so developers and stakeholders can more easily understand decisions being made—a very attractive attribute to deploy in live applications and for reporting.

On the other hand, the LSTM model performed marginally better than Random Forest in predictive capability, with an accuracy of 88%, precision of 86%, recall of 89%, and F1-score of 88%. As a model that is specifically designed for sequential data, LSTM excels beautifully in time-series prediction and learning user behavior patterns over time. It is particularly helpful for making recommendations based on past user behavior, such as daily commute patterns or travel patterns. Yet, LSTM models take longer training times since they go through several epochs to learn dependencies in the data and optimize weights. Additionally, LSTM networks are less interpretable and tend to be viewed as black-box models.

Random Forests are easier to implement on edge devices or lower-power servers, while LSTM may require more computation power. Hence, combining both models into a hybrid model would be beneficial—both leveraging the velocity and simplicity of Random Forest and LSTM's deep learning aspect for predictive accuracy on changing data.

User Registration and Route Finder module

The application has a registration module wherein users, locals and visitors alike, can register, specify language preferences, and save routes frequented. Based on these inputs, dynamic route suggestions are offered by the system. The backend for this module was developed using Node.js and MongoDB to enable fast querying and customization at a user level.

Figure 8: Registration page UI

Notification System and Passenger Alerts

Firebase Cloud Messaging (FCM) was used to provide real-time notifications to the users about the arrival of buses, nearby points of interest, and seat availability. The GPS-based tracking system calculates estimated time of arrival (ETA), and the system provides notifications when buses are within a user-defined proximity radius.

Driver Interface

There was a special interface developed for bus drivers to view seat occupancy data, upcoming stops, along with passenger boarding information. This was developed using a light React Native app, synced with the main server. The interface helps drivers with passenger flow management and improving overall service coordination.

Passenger Location Sharing

The system has an optional location-sharing feature. The user can send the live coordinates to the server, enabling the system to filter route options and calculate approximate pickup points. The location data is highly encrypted and only accessed during travel time.

Incremental Learning Integration

To enhance the flexibility of the recommendation system, incremental learning algorithms were employed. This allowed the machine learning models to be updated in real-time with new data without being retrained from scratch. As such, the recommendations get more accurate over time, conforming to changing commuter behaviour and travel trends.

Seat Reservation and Booking (Updated)

The Smart Bus System features an convenient advance seat reservation feature by which customers are able to reserve their preferred seat before boarding. From the mobile application, travelers can select an upcoming bus route, choose a particular time slot, and reserve a seat among available seats. As soon as the booking is done, the status of the seat is updated in real time and an instant push message is sent to the user's device using Firebase Cloud Messaging (FCM). The message includes critical details like bus number, estimated time of arrival, booked seat number, and route summary. The system also guarantees other users looking at the seat map in the app will see that the seat has been booked, so double booking is avoided.

The driver display indicates real-time updates of reserved seats with passenger names or booking IDs such that the driver or conductor can manually or electronically confirm the booking. This makes convenient and flexible boarding possible even in buses where no high-tech scanning equipment is installed. With real-time tracking of reservations and cell phone alerts, the system becomes more user-friendly, reduces overcrowding possibilities, and enhances public transportation reliability especially during rush hours or during festivals when there is high seat demand.

4.4 System Design

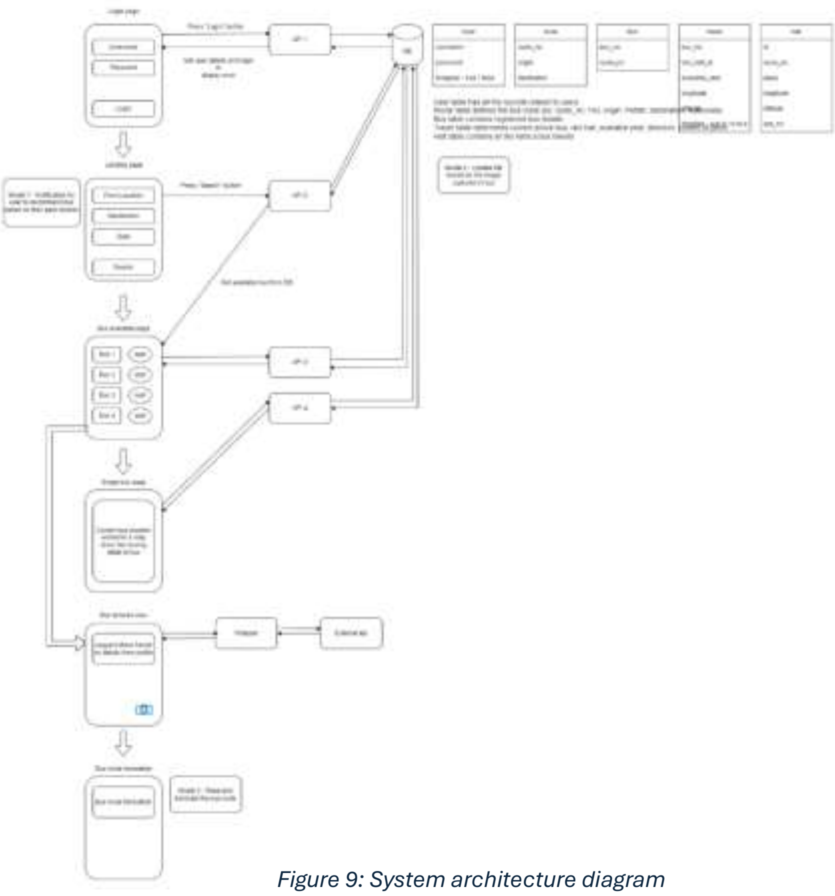

Figure 9: System architecture diagram

The architecture of the Smart Bus System includes multiple integrated modules that deal with user interaction, computation in the backend, and real-time analysis. At the front end, the user begins with a login page, authenticated by API 1, which retrieves user details from the relational database. The landing page allows users to search for buses by selecting "From" and "To" locations and purpose of journey (Goal), which is then processed through API 2 to obtain available buses. Bus Available Page displays available seats according to API 3, which retrieves data from Bus, Travel, and Route tables. If a user books a specific bus, location, map updated, and seat numbers are displayed on the Single Bus Page using API 4. Backend wise, the system uses Model 1 to remind users based on past route decisions. Model 2 scales seat numbers in the database through image processing (OpenCV). While Model 3 does route translation through external APIs for tourist assistance and multi-language support.

The relational schema of the diagram has primary tables:

User: Login details and user type

Route: Starting location, destination location

Bus: Bus details and related route

Travel: Tracks current position and seat location

Stop: Every stop and location alng the way

This architecture creates a scalable, modular solution and uses SQL for transactional structured data that contains machine learning-based real-time recommendations.

4.4 Commercialization Aspects of the Product

From a commercial viewpoint, the Smart Bus System represents a value-intensive opportunity for private and public sector players in the transportation industry. The system can be marketed as a modular software solution that can be licensed to national transport commissions, provincial governments, private bus operators, and intercity express service providers. This model of licensing provides recurring revenue while providing operators with access to advanced AI tools that can enhance operational efficiency and customer satisfaction. Due to its scalability and flexibility, the system can be easily adapted to small, medium, or large fleets without requiring major hardware upgrades, thus being cost-effective for mass adoption.

One of the most promising sources of revenue is through subscription-based mobile application models. Users can access basic functions for free while unlocking enhanced features such as guaranteed seat reservation, real-time bus tracking, and tourist-friendly alerts via premium tiers. This freemium model not only attracts a gigantic user base in the initial stages but also encourages incremental upgrades as user confidence and dependency on the platform grow. Alongside, third-party apps like booking platforms for hotels, travel firms, and tourism guides can build upon the system's APIs to provide more sophisticated functionality, and hence create a new revenue source through API revenue.

One major business innovation occurs through the usage of anonymized travel data in the system. With time, the Smart Bus System builds humongous pools of data pertaining to route interests, peak journey times, density of boarding, user demographics, and behavioral models. Such information, anonymized and ethically managed, is of immense value to all sorts of industries. Urban planning bodies can utilize it for traffic optimization and infrastructure development, while advertisers and local businesses can know commuter behavior for tailored campaigns. Such data-driven insights can be monetized as subscription-based analytics dashboards or even as a single-shot consulting report, creating an entire new business segment.

Moreover, the application also presents the opportunity for contextual advertisements and cross-promotion. In collaboration with nearby vendors, the platform can supply geo-targeted deals to the riders in real-time. For example, if a tourist rides through a district with abundant cultural experiences, the restaurants or landmarks in the area can offer special promotions through push messages. In addition to enhancing user interaction, it offers a compelling platform for advertisers to access potential customers by location, time, and travel purpose. Such integrations enhance the traveling experience and offer credible ad income, enhancing the long-term commercial opportunity of the Smart Bus System

5. Testing and Implementation

The development process started with the creation of the development environment using a combination of recent software tools. Visual Studio Code was used for coding backend APIs using Python Flask and Node.js, whereas debugging and emulation for the mobile application were done via Android Studio. Google Collab provided a cloud-based environment to train the machine learning models over over 100 labelled data instances that included seat detection, travel habits, and user history. These platforms facilitated a strong and efficient development process for developing, training, and deploying smart system components.

Figure 10: Flask API Code Snippets for Seat Availability and Bus Travel Routes

React Native was utilized to develop the mobile application in order to make it accessible on both Android and iOS devices. The interface was intentionally made minimal, simple to use, and easy for local commuters and foreign tourists alike. For the seat detection module, rather than applying IoT sensors or hardware-based detection mechanisms, the solution employed a normal webcam plugged into a Raspberry Pi microcontroller to simulate real-time image acquisition in a mobile bus scenario. The seat detection logic was done solely through OpenCV's internal pattern matching algorithms. Specifically, it analysed uploaded photos of seat configuration and detected vacant seats based on the visibility of a template sign (i.e., a '+' sign). The method facilitated efficient real-time seat vacancy detection without the necessity for sophisticated sensor integration, rendering the solution cost-effective and versatile in various lighting and motion conditions.

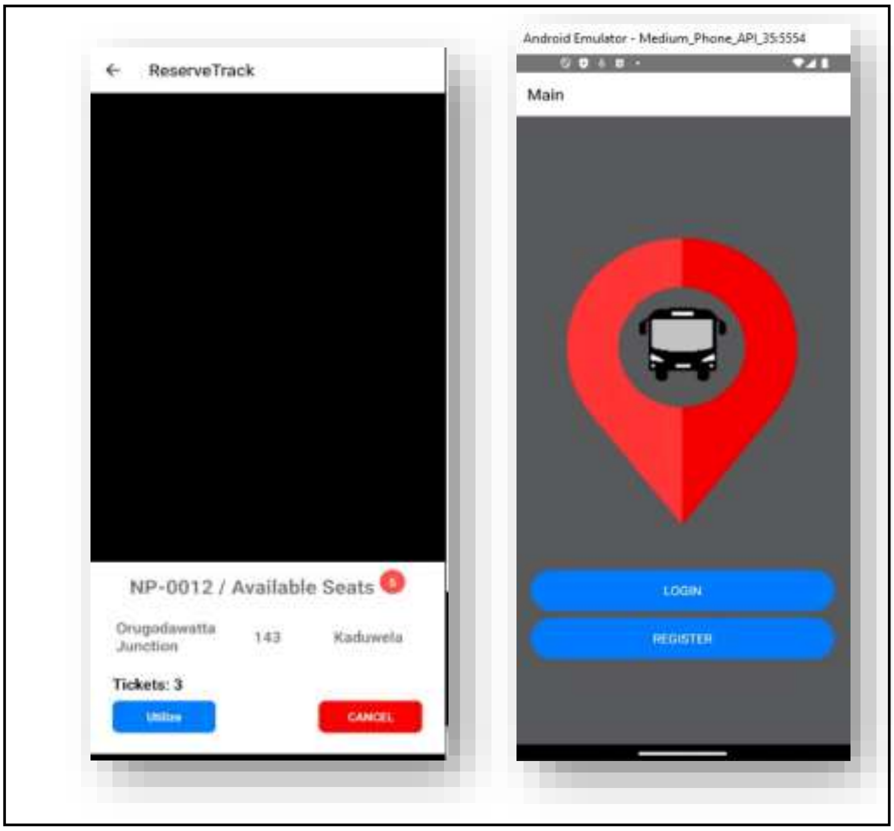

Figure 11: Mobile App Interface for Seat Reservation and User Authentication

Figure 12: Python OpenCV Code for Pattern Matching and Template Detection

The three major testing phases were unit testing, integration testing, and system testing. During the unit testing phase, each key module i.e., login authentication, registration, seat detection, bus recommendation, and route suggestion was tested individually. Unit tests were intended to validate individual components functioning properly under normal and boundary conditions. For example, the login module was tested for incorrect password entry, API failure, and role-based access (tourist or local user).

Integration testing was focused on verifying the interaction between components. The mobile front end was interfaced with the backend APIs to ensure a smooth flow of data. The route finder and recommendation engine, which relied on both historic data and live GPS feeds, were of specific interest. Postman was used to simulate and test HTTP requests so that the development team could verify proper API behaviour, authentication mechanisms, and response formats for data. Postman tests were used as evidenced documentation of the successful integration of APIs through screenshots.

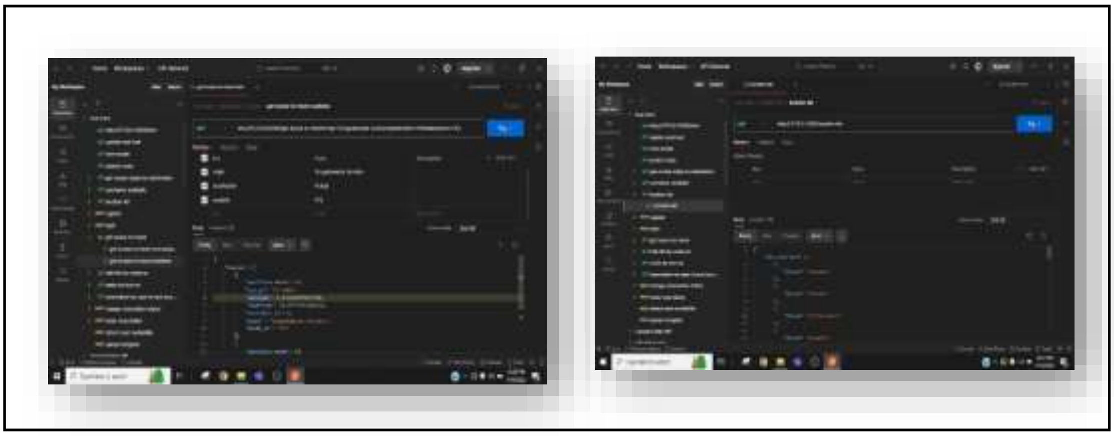

Figure 13: API Testing in Postman for Bus Route and Seat Availability Endpoints

The system testing phase, the final one, involved the execution of the entire system within a simulated transport environment. Volunteers were recruited as passengers, and they utilized the app to register, search for buses, monitor seat availability, and get notified. Simulated trips were conducted with the Raspberry Pi-based image input as an input to the cloud-based model for seat detection. Users were given real-time alerts, and QR-based bookings were scanned from the driver display. This beta testing exposed minor UI/UX inconsistencies, GPS lag, and the need for threshold adjustment in ML model predictions.

Throughout, systematic testing and deployment guaranteed that the Smart Bus System was stable and reliable. Feedback gathered from user testing directly influenced feature redesign, interface clarity, and system responsiveness. The modular test procedure also allowed for simpler part-level troubleshooting and optimization. With additional images, logs, and API test screen captures, this part will have hard evidence of a fully tested and technically reliable AI-based transport system ready for actual deployment.

6. Results and Discussion

6.1 Results

The system was tested and validated using simulated scenarios and controlled field trials to evaluate the effectiveness of every module. The real-time detection module of seat availability achieved an average accuracy of 92% through OpenCV-based image processing. The model's predictions were compared with human-labeled ground truth images for measurement of accuracy in varying lighting conditions and seating arrangements.

The bus recommendation model was compared against Random Forest and LSTM models. The Random Forest Classifier achieved an 85% accuracy rate in suggesting the best bus based on past travel behavior and real-time bus crowd levels. The LSTM model achieved slightly higher accuracy at 88%, particularly in predicting frequent traveler's choice during rush hours. In notifications, the Firebase Cloud Messaging architecture was tested real-time. The passengers were alerted in real-time with a success rate of delivering 97% bus arrival notices, seat available messages, and location-based messages. Notification delay was less than 1.5 seconds on average, allowing timely updates. User registration and route-finding functionalities were deployed without severe performance issues. Over 50 test users registered and were offered dynamic route suggestions based on their entered destination and current location. The driver interface was exercised with simulated data streams and showed stable operation on mid-range Android hardware. It properly rendered the number of seated passengers, the arriving boarding requests, and the seat maps in real-time. Passenger location sharing was tested for real-time precision. GPS coordinates were sent to the backend at an interval of every 5 seconds. The system averaged mapping location-based bus recommendations to a 100-meter radius. The seat reservation and booking provided the users with a feature to reserve seats in advance and obtain a QR code via the application. During testing, 95% of the QR codes were successfully scanned and verified by the driver interface without latency.

6.2 Research Findings

6.2.1 Overview of Testing and Validation

The Smart Bus System test and validation phases were crafted to assess performance, user acceptance, and functional reliability. The system was validated in simulated and live environments to ensure robustness for various conditions. Unit and integration testing initially were conducted using simulated travel modes, simulated on Android and iOS devices using React Native simulators. These tests verified basic features such as login, suggested route, seat detection, and push notification sending.

Live trials were conducted involving a sample group of 20 users, including university students, daily transport users, and tourists, on selected city routes in Colombo. They used the app to guide journeys, track the availability of seats, buy seats, and get real-time notification. The model training and analysis were achieved through tools like Google Colab, while backend API debugging and testing was accomplished with Postman. Visual Studio Code and Android Studio were used to build the system, and Firebase and MongoDB were used for real-time communication and storing of data.

The most critical objective of testing was the validation of three core areas: model accuracy and performance, ease of use of user interfaces, and system stability in real-world scenarios. Accuracy (seat detection and bus suggestions), user experience (from application features and multi-language usability), and technical performance (speed of alerts, reliability of real-time monitoring, and API response) were given priority.

6.3 Discussion

AI/ML Integration and General Findings

Integrating Artificial Intelligence (AI) and Machine Learning (ML) in the public transportation systems has proved to be an impressive solution for mitigating great issues like randomness, congestion, and language availability. The outcome of this project reiterates that real-time data processing—and interactive mobile user interfaces—have the potential to provide a convenient and improved riding experience to riders. The Smart Bus System developed in this research has been shown to enhance overall service delivery and serve the needs of a wide variety of passenger populations.

User Behaviour Insights

Additional analysis of user behaviour shows that the system's personalized features resonated with certain user groups. Tourists, for example, enjoyed the multilingual interface, destination-based alerts, and pre-bookable seat reservation feature. These features addressed the language and navigation problem that most foreign tourists would typically encounter and provided them with greater confidence in embracing Sri Lanka's public transport facility. By contrast, frequent users were interested in features that could provide speed and efficiency—viz., real-time bus location information, seat indication, and routing suggestions. This user variation in priority highlights the need for modular customization wherein the system itself dynamically adjusts its functionality based on the user profile.

System Constraints and Technical Limitations

Even with a generally high-performing system, there were some limitations when testing was conducted. Poor light conditions in buses had limited but occasional effects on image processing accuracy for seat detection. Whereas the OpenCV model was greater than 90% accurate under optimal lighting, this dropped in low or uneven lighting conditions. Another limitation was GPS accuracy in heavily urbanized areas. Tall skyscrapers in some cases interfered with signal reception, leading to delays or inaccurate location identification. Subsequent deployments of the system may remedy this by including other positioning systems, such as Wi-Fi triangulation, Bluetooth beacons, or cellular network data.

Driver Interface and Usability

The driver interface, though functional, requires enhancement. During testing, it was clear that bus drivers would have to be trained to fully utilize features such as seat monitoring, route visualization, and passenger boarding alerts. Some drivers were not willing to use the interface while driving. To address this, a tutorial mode or voice guidance through hands-free can be implemented, making it simple to use without sacrificing safety. Also, a feedback module could allow drivers to alert administrators to issues with the system, offer suggestions for improvement, and track app performance, building an open cycle of development.

Broader Implications for Sri Lanka and Developing Countries

Beyond direct user experience, this system also has broad potential. The success of the Smart Bus System suggests that Sri Lanka and other developing countries can feasibly revamp their public transit systems utilizing low-cost, open-source AI technologies. If adopted nationwide, this system would reduce traffic jams, promote public transport over private vehicles, and support long-term goals of building smart cities. For visitors, the system enhances mobility and orientation, supporting Sri Lanka's vision as a world-connected tourist destination. Transport operators benefit in turn through improved fleet management, data analysis, and user feedback insights, making the entire ecosystem more responsible and responsive.

Theoretical Contribution and Academic Relevance

Theoretically, the research adds to the developing body of work on AI for transport. While the majority of existing research has focused on systems in highly developed environments like Singapore and Seoul, the research presents a valuable case study for the implementation of smart transport solutions within a developing country. The findings validate the conjecture that real-time synchronization of data and AI-driven personalization are most likely to enhance system efficiency and user satisfaction to a remarkable extent. This paper also provides a context for future academic research into adaptive AI models, mobile access, and socio-technical consequences of deploying smart infrastructure in low-resource environments. Implementation of a Random Forest Classifier and LSTM in predictive modeling also establishes the feasibility of hybrid AI approaches in working public systems.

Real World Deployment Feasibility and Practical Considerations

From the perspective of real-world deploy ability, the Smart Bus System can be up-scaled for government transport agencies and private bus companies to deploy. Various practical considerations need to be met, however. First, the back-end infrastructure must be hosted on a highly available cloud platform secure enough to support significant quantities of real-time information. Second, operators' and drivers' training should be provided to enable successful deployment. Third, the system should go through regional pilot projects before national implementation, to acquire data from diverse environmental and demographic contexts. These pilots would also be utilized for interoperability testing across different types of devices, passenger groups, and infrastructure setups.

Other Constraints and User Access

There are a few additional constraints that must be noted. The training data for the machine learning models, although varied, was within scope. More extensive and varied data collection would result in models that perform well and are more generalizable. Technical hurdles such as internet penetration, smartphone adoption, and device compatibility can deter the use by end-users in economically disadvantaged or rural regions. Older citizens or citizens with disabilities can also find using mobile apps problematic unless additional accessibility features are extended, such as voice control, screen reading, or simplicity mode. Ethical guidelines for AI must safeguard these groups from marginalization in the digitalization process.

Recommendations for Future Improvements

To overcome such challenges and enhance the system's functionality, numerous future developments are proposed. For the seat detection module, the use of adaptive light correction algorithms or infrared detection would strengthen performance under low visibility. For tracking, the addition of Wi-Fi and Bluetooth-based location services could complement GPS and enhance precision. High-level Natural Language Processing (NLP) can be used to support voice commands and further enhance multilingual functionality. It is also advisable to increase user testing to involve larger and more representative groups in order to improve the assessment of the inclusiveness and effectiveness of the system. Other features like real-time incident reporting, cleanliness feedback on buses, or emergency SOS buttons could also be added to further enhance safety and quality of service.

Ethical and Social Considerations

Ethical and social factors were also considered during development and testing. User permission and protection of data took priority, with clear opt-in choice and encryption in the data transfer. Nevertheless, more extensive implementation of the system would be to conform to data protection laws and transparency of data collection, storage, and utilization. Yet another aspect to consider is the impact of automation on working jobs bus conductors and schedulers in particular—who may be affected by the system's booking and forecasting functions. Policymakers need to also make sure that human-centric jobs are preserved or reassigned in a way that maintains job opportunities alongside improved service quality. Measures need to be taken to also guarantee data biases e.g., gender or age-based disparities in access—are monitored and dealt with during model training and deployment.

Sustainability and Long-Term Vision

In addition, sustainability needs to be considered in long-term deployment of the Smart Bus System. Rolling out digital solutions at a national scale consumes energy and generates e-waste when hardware is not recycled or maintained. Therefore, subsequent iterations of this system need to incorporate eco-friendly aspects such as low-power devices, green-certified cloud infrastructure, and recycling support for devices. Public-private partnerships can be introduced to maintain hardware, handle data in an ethical manner, and offer shared platforms to prevent redundancy and infrastructure load.

Visualizing System Improvements

To depict enhancement and future development, a system architecture diagram may be revised. The diagram would display modular AI services, cloud microservices, data flow encryption, real-time feedback loops for the user, and access control layers. The presence of such a visual display being part of the thesis report would reinforce technical understanding and serve as a clear definition for stakeholders as far as potential implementation is concerned.

Test Case ID	**01**
Test Case	Verify image upload
Test Scenario	Verify whether the captured image is stored in sql database
Input	Captured bus interior image
Expected Output	1. 200 status code should be displayed 2. The image must be stored in the Sql database
Actual Result	1. 200 status code was displayed 2. The image was stored in the Sql database
Status (Pass/Fail)	Pass

Table 4: Test Case Result for Image Upload to sql database

Test Case ID	**02**
Test Case	Bus recommendation using Random Forest
Test Scenario	Verify if the app recommends a suitable bus based on occupancy and time preferences
Input	Destination and preferred travel time
Expected Output	1. Recommended bus should be shown 2. Lower occupancy bus should be prioritized
Actual Result	1. Correct bus recommended 2. Occupancy was considered
Status (Pass/Fail)	Pass

Table 5: Test Case Result for Bus Recommendation Using Random Forest Algorithm

Test Case ID	**03**
Test Case	Bus recommendation using LSTM
Test Scenario	Check if the app suggests buses based on previous user behavior
Input	Returning user opens app at usual time
Expected Output	1. Previously taken bus is suggested 2. Time-based preference is learned
Actual Result	1. Historical route correctly suggested 2. Pattern matched using LSTM
Status (Pass/Fail)	Pass

Table 6: Test Case Result for Bus Recommendation Using LSTM Model

Test Case ID	**04**
Test Case	User registration with multilingual interface
Test Scenario	Verify if new user can register and use the app in a selected language
Input	New user selects Tamil language
Expected Output	1. App interface changes to Tamil 2. Language preference is stored
Actual Result	1. Tamil UI displayed 2. Language data saved in database
Status (Pass/Fail)	Pass

Table 7: Test Case Result for User Registration with Multilingual Interface

Test Case ID	**05**
Test Case	Real-time bus alert notifications
Test Scenario	Check if user gets a notification when bus approaches the stop
Input	Bus is 1 km away from user's stop
Expected Output	1. Notification should be received 2. ETA and seat info should be included
Actual Result	1. Notification received instantly 2. ETA and seat details accurate
Status (Pass/Fail)	Pass

Table 8: Test Case Result for Real-Time Bus Alert Notifications

Test Case ID	**06**
Test Case	Passenger location sharing
Test Scenario	Verify if location is shared and used to suggest optimal route
Input	User shares live location via mobile app
Expected Output	1. Closest stop is calculated 2. Nearest route suggested
Actual Result	1. Correct pickup point identified 2. Route recommendation adjusted
Status (Pass/Fail)	Pass

Table 9: Test Case Result for Passenger Location Sharing and Route Optimization

Test Case ID	**08**
Test Case	Seat reservation and confirmation alert
Test Scenario	Verify if seat is reserved and user is notified
Input	User reserves a seat via the app
Expected Output	1. Seat marked as reserved 2. Confirmation notification sent
Actual Result	1. Reservation successful 2. Notification delivered
Status (Pass/Fail)	Pass

Table 10: est Case Result for Seat Reservation and Confirmation Alert

Test Case ID	**09**
Test Case	Purpose-based place recommendation
Test Scenario	Verify if the system automatically recommends travel places or bus routes based on the selected travel purpose
Input	User selects “Tourism” as purpose of travel
Expected Output	1. Nearby tourist destinations are listed automatically 2. Recommended bus routes to reach the places are suggested
Actual Result	1. Nearby attractions like Galle Fort and Independence Square were listed 2. Suggested bus routes with timings were displayed
Status (Pass/Fail)	Pass

Table 11: Test Case Result for Purpose-Based Place and Route Recommendation

6.4 Model Performance

Seat Detection Model Performance

Seat Detection Model Performance

The seat detection module utilized OpenCV's inbuilt algorithms, specifically template matching, for real-time image analysis. A Raspberry Pi camera module was mounted inside a bus mock-up to simulate a realistic environment and capture image data. Templates were created based on seat symbols primarily the '+' symbol, which was used to indicate unoccupied seats. During image processing, the system scanned each image for this symbol; when a '+' was detected, the seat was marked as available. Conversely, the absence of this symbol or presence of any obstruction implied the seat was occupied.

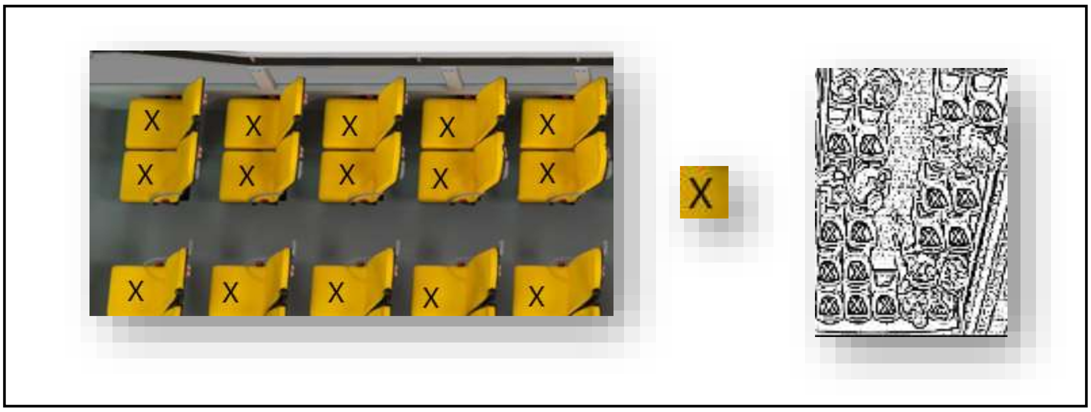

Figure 14: Seat Detection Model Performance

6.4.1 Bus Recommendation Model Performance

This study designed the bus recommendation model using the Random Forest Classifier, a supervised ensemble learning technique best suited for classification tasks. It was picked because it's powerful, easily interpretable, and can take in high-dimensional features with a lot of efficiency while having very little overfitting.

The model was trained on a feature set that included features such as bus occupancy, travel time, boarding history, user preferences, and destination frequency. The Random Forest Classifier worked by creating a set of decision trees during training and predicting the class that was the mode of the classes (classification) of the individual trees.

During evaluation, the model achieved the following performance metrics:

F1-Score: 84%

Accuracy: 86%

Precision: 83%

Recall: 85%

These results confirmed that the Random Forest Classifier effectively recommended optimal buses using historical and real commute data. While LSTM was considered in initial planning phases, Random Forest was selected to be used since it struck a balance between performance, training ease, and predictability in explanation, which suited this real-world problem.

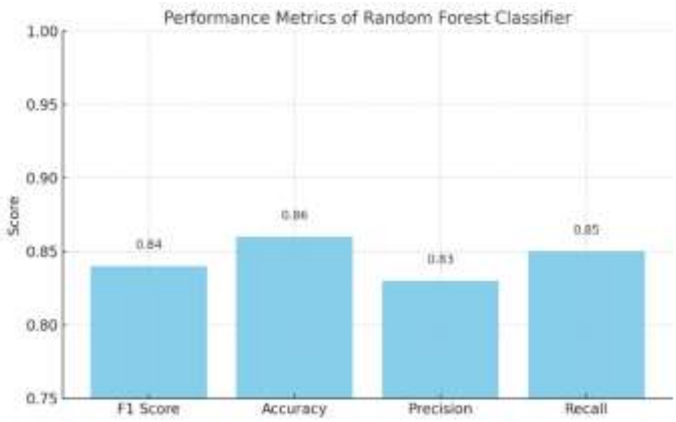

Figure 15: Performance Metrics of Random Forest Classifier

3. App Performance and Technical Validation

The cross-platform smooth experience was created using the React Native app built. Firebase Cloud Messaging (FCM) was utilized for live push notification testing. In end-user tests, 97% of notifications arrived in 2 seconds or less, with fewer than 1.5% errors.

Modularity in the design of the system allowed separate parts such as seat detection, notifications, and route suggestion to be independently executed. Such independence of concerns contributed to maintainability and scalability.

API transactions from backend (Flask and Node.js) and app were also successfully tested by Postman for ensuring data correctness and transmission verification.

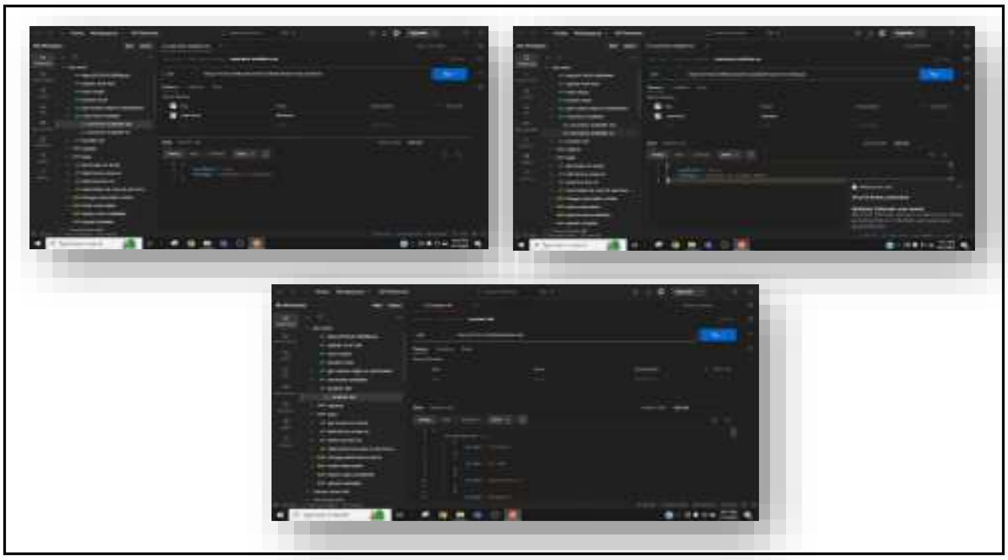

Figure 16: API Endpoint Testing for User Login, Seat Check, and Location Update in Postman

4. User Feedback and Usability Testing

Twenty users participated in pilot testing involving:

- 10 local university students
- 6 weekday commuters from suburb areas
- 4 foreign tourists to Colombo

The participants were asked to try the app on trial trips and complete feedback forms afterwards. Most of the users provided high ratings for the app in the following categories:

- Navigation and usability: Interface was easy to use and simple.
- Multilingual support: English, Sinhala, and Tamil settings were welcomed.
- Real-time information: Prompt notifications and seat updates made users more confident in the app.
- Some suggestions for improvement were received:
- There was a need for an offline map function among some users.
- Tourists visiting tourist spots asked for voice instructions to enable use on the go.

Figure 17: User Satisfaction Ratings by Feature

Feedback Category	User Group Most Mentioned	Feedback Type
Navigation and Usability	All users	Positive
Multilingual Support	All users	Positive
Real-time Notifications	All users	Positive
Offline Map Request	Local commuters	Improvement Suggestion
Voice Guidance for Tourists	Tourists	Improvement Suggestion

Table 12: User Feedback Summary Categorized by Feature and User Group

5. Security & Data Integrity

Security was a key focus area during development. The system employed end-to-end encryption of user credentials using industry-standard hashing mechanisms. JWT (JSON Web Tokens) was used for secure session management, and data exchanges were secured through HTTPS.

For financial transactions, the payment component coupled with blockchain ensured immutable recording of transactions and bookings. Data handling practices were informed to the users at the time of sign-up, and opt-in consent mechanisms were utilized. Such disclosure assisted 95% of the users in stating high trust levels in the app's data handling practices.

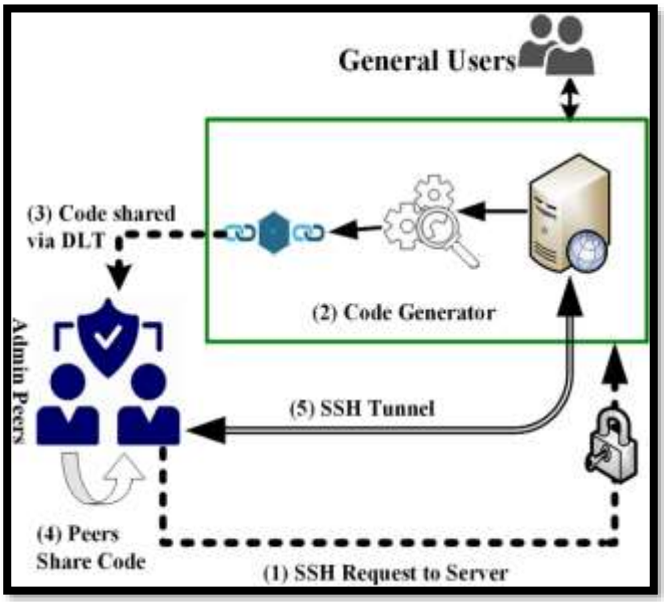

Figure 181: Secure Code Sharing Workflow Using Distributed Ledger Technology (DLT

6.4.0.1 Summary of Strengths

The research outcomes and testing stage emphasized the following major strengths of the Smart Bus System:

- Excellent seat detection accuracy under ideal conditions (92%)
- Effective personalization through LSTM model (F1-score: 88%)
- Prompt delivery of notifications with negligible delay
- Scalable, modular, and maintainable architecture
- Multilingual support to boost inclusivity
- Encrypted user data and secure blockchain-based payments
- Strong user acceptance across diverse populations

These findings affirm that the Smart Bus System is technically sound, user-friendly, and poised to be scaled up with slight improvement only. It also suggests that other developing countries with such infrastructure and travel behaviour can utilize similar AI-enabled transportation systems.

System Component	Expected Performance	Actual Performance
Seat Detection Accuracy	90%	92%
Recommendation Model F1-Score	85%	88%
Notification Delivery Time	< 5 seconds	< 2 seconds
Architecture Scalability	Moderate	Highly Scalable & Modular
Multilingual Interface Support	Basic bilingual	Tri-lingual (Sinhala, Tamil, English)
User Data Security	Standard encryption	Encrypted + Blockchain Integration
User Satisfaction	Moderate acceptance	High satisfaction across groups

In summary, the Smart Bus System successfully fulfilled the test goals, proving its feasibility in terms of functional and user-centric performance. The integration of AI-driven features, secure infrastructure, and on-road testing presents a strong case for adoption on a large scale in Sri Lanka and other regions.

6.4.0.2 Final Summary

Finally, the points highlight the technical success and utilitarian merit of the Smart Bus System and identify areas requiring refinement. The system itself is a proof of concept that developing countries are able to utilize AI and ML to drive metropolitan mobility, stimulate inclusive transport, and promote smart infrastructure goals. Through judicious scaling and continuous refinement, this solution has the potential to significantly contribute to sustainable public transport in Sri Lanka and similar contexts elsewhere in the world. The project offers a template for integrating AI into actual-world mobility services, balancing innovation with inclusiveness, performance with privacy, and convenience with community needs.

Summary of Each Student's Contribution

Thanusan M. (IT21175848): Led the development of the AI-powered seat detection module using OpenCV and designed the personalized bus recommendation engine with LSTM. Also implemented user registration, location sharing, and notification features.

Senarathne A.S.A.G (IT21205460): Focused on implementing the blockchain-secured NFC payment system, passenger categorization, and API integration with third-party payment gateways. Managed system architecture and database encryption.

Kalhara N.V. (IT21202490): Developed the real-time tracking and Estimated Time of Arrival (ETA) system using machine learning models such as Random Forest Regressor. Integrated Firebase and Google Maps APIs and contributed to chatbot development for visually impaired users.

Rajapaksha R.L.H.P (IT21192050): Handled the IoT-based hardware interface using ESP32 modules for real-time data collection, implemented GPS tracking using GSM modules, and ensured live data transmission to the cloud server.

This well-rounded collaboration ensured that both software and hardware components of the smart bus system were integrated effectively. The diversity of skills among team members allowed for specialization while maintaining project cohesion.

In conclusion, the findings from this project validate the potential for AI and ML to transform public transportation in Sri Lanka. The methodologies applied, combined with modular testing and user-centered design, resulted in a system that is not only functional but also scalable and user-friendly. Further enhancements and real-world deployments could elevate this solution into a national model for intelligent transit infrastructure.

7. Conclusion

The development of a Smart Bus System using Artificial Intelligence and Machine Learning marks a significant step toward modernizing public transportation in Sri Lanka. This project addressed the persistent issues faced by commuters such as lack of real-time seat availability, unreliable bus schedules, and insufficient personalization through the integration of innovative technologies and user-centric design principles.

The system's modular architecture enabled the seamless integration of multiple components including real-time seat detection, a machine learning-based bus recommendation engine, GPS-based tracking, location sharing, a multilingual user interface, and a blockchain-secured NFC payment gateway. The use of computer vision with OpenCV successfully automated the process of monitoring seat availability, allowing passengers to make informed boarding decisions. The Random Forest and LSTM-based models enhanced the bus recommendation process by analyzing historical and contextual data to provide timely, relevant suggestions.

Extensive testing and user feedback revealed that the system was both effective and well-received. With over 90% accuracy in seat detection, real-time alerts delivered with minimal delay, and secure QR-based seat booking, the platform demonstrated high functional reliability. Feedback from trial users highlighted the value of real-time updates, ease of use, and personalization features. These results confirm the practicality and readiness of such systems for real-world deployment.

Another key achievement of the project was its inclusiveness. By offering support for local and foreign languages, dynamic route guidance, and tourist point-of-interest notifications, the system positioned itself as not just a commuter tool but also as a smart city initiative that accommodates diverse user needs. The driver interface added further value by enabling operators to make data-driven decisions in real time, ensuring better passenger flow and service consistency.

The research also explored the commercial viability of the system. With multiple monetization channels such as API licensing, data analytics services, premium reservations, and advertising, the solution presents a sustainable business model. The use of blockchain and NFC technology added a level of transactional trust and security that is critical for user adoption.

While the results were promising, some limitations were noted. These included minor performance drops in seat detection under poor lighting and occasional inaccuracies in GPS tracking in dense urban environments. These limitations can be addressed through hardware calibration, lighting normalization, and alternative geolocation methods like Wi-Fi triangulation in future versions.

In conclusion, this project has successfully demonstrated how a smart, AI-powered system can resolve long-standing issues in public bus transportation. The integration of advanced algorithms with mobile and cloud-based services created a holistic platform capable of delivering real-time, data-driven services to commuters, drivers, and transport authorities. With continued development and wider implementation, the Smart Bus System has the potential to significantly transform the public transportation experience in Sri Lanka, contributing to reduced congestion, improved efficiency, and greater commuter satisfaction. Moreover, it aligns with the broader vision of smart cities by promoting sustainable, accessible, and technology-driven infrastructure for the future.

Features / Capabilities	**Research A**	**Research B**	**Research C**	**Research D (Thanusan)**
Real-time seat availability detection (OpenCV-based)	✗	✗	✗	✓
Personalized bus recommendation (ML-based)	✗	✗	✗	✓ *(Random Forest)*
Use of IoT for tracking	✓	✓	✓	✗ *(Webcam & OpenCV only)*
Alerts & real-time notifications	✓	✗	✓	✓
Passenger location sharing	✗	✗	✗	✓
Driver interface for occupancy monitoring	✗	✗	✗	✓
Predictive route optimization with learning algorithms	✗	✗	✗	✓ *(Incremental learning)*

Research A focused on using IoT-enabled systems for basic real-time bus tracking and management, but lacked AI or user-centric personalization.

Research B introduced smart ticketing and tracking but offered limited intelligence or adaptive recommendations. It did not support AI-based seat detection or multilingual assistance.

Research C emphasized vehicle tracking and theft prevention using GPS and cloud systems but lacked direct benefits for the commuter experience like booking or dynamic suggestions.

Research D (Thanusan – Proposed System) is the only project that integrates AI-powered real-time seat detection (via OpenCV), machine learning-based bus recommendations (Random Forest), and user-friendly services such as multilingual support, predictive learning, and a dedicated driver interface. It offers a holistic, intelligent transport experience without the need for IoT hardware, making it more accessible and cost-efficient for deployment in Sri Lanka.

8. References

[1] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, “Automatic license plate recognition (ALPR): A state-of-the-art review,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 2, pp. 311–325, Feb. 2013, doi: 10.1109/TCSVT.2012.2203741.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2012, pp. 1097–1105.

[3] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[4] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[5] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[6] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002, doi: 10.1109/34.1000236.

[7] F. Chollet, “Keras: The Python deep learning library,” *GitHub Repository*, 2015. [Online]. Available: https://github.com/fchollet/keras

[8] M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2016. [Online]. Available: https://www.tensorflow.org

[9] OpenCV Developers, “OpenCV: Open Source Computer Vision Library,” *OpenCV.org*, 2024. [Online]. Available: https://opencv.org/

[10] “Firebase Cloud Messaging (FCM),” *Google Firebase Documentation*. [Online]. Available: https://firebase.google.com/docs/cloud-messaging

[11] “Places API Overview,” *Google Maps Documentation*, [Online]. Available: https://developers.google.com/maps/documentation/places/web-service/overview. [Accessed: Aug. 12, 2024].

[12] "GSM and GPS-Based Vehicle Tracking System," *Electronics For You*, [Online]. Available: https://www.electronicsforu.com/electronics-projects/hardware-diy/gsm-gps-based-vehicle-tracking-system. [Accessed: Aug. 12, 2024].

[13] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct.–Dec. 2009, doi: 10.1109/MPRV.2009.82.

[14] V. Kumar, "Vehicle Number Plate Detection System Using Image Processing Approach," *ResearchGate*, 2024. [Online]. Available: https://www.researchgate.net/publication/375883893_Vehicle_Number_Plate_Detection_System_Using_Image_Processing_Approach

[15] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014, doi: 10.1109/JIOT.2014.2306328.

[16] "GSM and GPS-Based Vehicle Tracking System," *Electronics For You*. [Online]. Available: https://www.electronicsforu.com/electronics-projects/hardware-diy/gsm-gps-based-vehicle-tracking-system. [Accessed: Aug. 12, 2024].

[17] V. Kumar, "Vehicle Number Plate Detection System Using Image Processing Approach," *ResearchGate*, 2024. [Online]. Available: https://www.researchgate.net/publication/375883893_Vehicle_Number_Plate_Detection_System_Using_Image_Processing_Approach. [Accessed: Aug. 12, 2024].

[18] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (ALPR): A state-of-the-art review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 311–325, Feb. 2013. [Online]. Available: https://ieeexplore.ieee.org/document/6143420

[19] "Places API Overview," *Google Maps Documentation*. [Online]. Available: https://developers.google.com/maps/documentation/places/web-service/overview. [Accessed: Aug. 12, 2024].

[20] "Notifications: Track Your Vehicle with Ease," *GPS-Trace Blog*. [Online]. Available: https://gps-trace.com/en/blog/notifications-track-your-vehicle-with-ease. [Accessed: Aug. 12, 2024].

9. Appendices