MUTMA’INN: An AI-Driven Edge–Cloud Framework for Safe and Intelligent School Bus Transportation

Ahad Alotaibi
Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia.
Department of Computer Information System.

Rayana Aldulaijan
Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia.
Department of Computer Information System.

Aljoharah Alabdulmohsen
Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia.
Department of Computer Information System.

Danah Aljowaiser
Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia.
Department of Computer Information System.

Rawdah Alhindi
Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia.
Department of Computer Information System.

Asiya Abdus Salam
Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia.
Department of Computer Information System.

Mona Albinali
Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia.
Department of Computer Information System.

Rabab Alkhalifa
Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia.
Department of Computer Engineering.

ABSTRACT
Student safety during daily school transportation remains a major concern, particularly in systems that rely mainly on GPS tracking and manual supervision. Existing approaches often lack proactive safety mechanisms for monitoring both student attendance and driver condition in real time. This paper presents **MUTMA’INN** derived from the Arabic word **“مطمئن”**, meaning *being reassured, at peace,* or *tranquil,* reflecting the system’s role in ensuring the safety and security of students during transportation. The proposed system is an AI-powered school bus safety framework designed to improve the security and reliability of daily student transportation in alignment with Saudi Vision 2030’s Quality of Life Program. The proposed system consists of two integrated components: a cross-platform Flutter mobile application for parents, drivers, and school administrators, and a python-based edge system connected to Firebase for real-time synchronization. The framework automates student attendance through facial recognition at the bus gate, reducing manual effort and the risk of human error. In addition, it monitors the driver using contactless remote photoplethysmography and facial analysis techniques to estimate heart rate and detect signs of fatigue or emotional distress. When abnormal conditions are detected, immediate alerts are sent to administrators to support timely intervention. By combining mobile computing, edge intelligence, computer vision, and cloud services into a unified platform, **MUTMA’INN** provides a proactive approach to school transportation safety. The proposed framework demonstrates how AI can support safer and more intelligent student transit systems.

Keywords
Remote Photoplethysmography (rPPG), Driver Monitoring System, Computer Vision, Facial Recognition, Emotion Detection, Smart Transportation, Internet of Things (IoT).

1. INTRODUCTION
The safety of students during daily transportation to and from school remains a significant concern worldwide. According to the *World Health Organization*, road traffic injuries are the leading cause of death among children and young people aged 5–29 years [1]. In Saudi Arabia, road safety continues to be a critical issue; recent statistics from the *General Authority* for Statistics report 17,231 serious traffic accidents, 4,282 fatalities, and 24,077 injuries in 2024 [2].

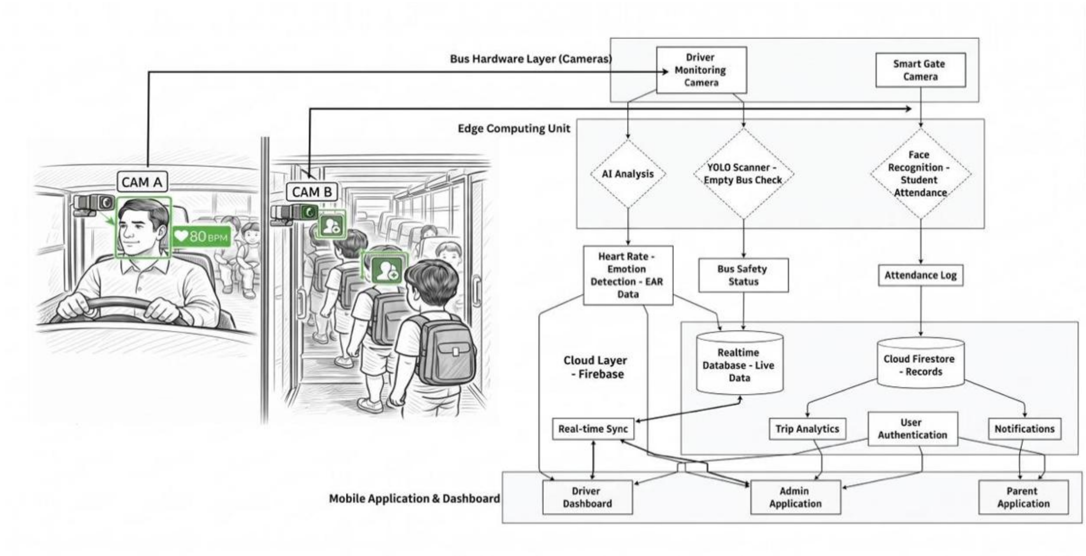

Figure 1: Overview of the proposed system architecture, illustrating the end-to-end workflow integrating in-bus camera modules with edge computing and cloud services. The system captures driver and student data through dedicated cameras, enabling real-time processing for face recognition (attendance), object detection (bus safety), and rPPG-based physiological monitoring of the driver. Processed data is transmitted to the cloud for storage, analysis, and synchronization with the mobile dashboard, supporting continuous monitoring, alert generation, and overall transportation safety management.

At the national level, student transportation is supported through formal services provided by the *Ministry of Education*, aiming to ensure safe and organized travel for students [3]. In addition, regulatory bodies such as the *Transport General Authority* emphasize compliance with safety standards in educational transport operations [4]. These efforts align with Saudi Vision 2030, which promotes safer and more efficient transportation systems as part of improving overall quality of life [5]. Despite these initiatives, many existing school transportation systems remain largely reactive, relying primarily on GPS tracking and post-event monitoring rather than proactive, real-time safety mechanisms. This limitation highlights the need for intelligent systems capable of preventing incidents before they occur.

1.1 Problem Statement

Current school transportation systems face several critical limitations. First, student attendance is often verified manually or through basic tracking methods, making it prone to human error and increasing the risk of serious incidents, such as students being left unattended inside buses.

Second, these systems typically lack continuous and non-invasive monitoring of the driver's physiological and emotional condition. This creates a significant safety gap, as fatigue, stress, or sudden health issues may go undetected until an accident or unsafe event occurs. Recent studies in driver monitoring systems [6] highlight the growing importance of real-time detection of driver state using computer vision and contactless sensing techniques.

Furthermore, while technologies such as facial recognition, driver monitoring, and object detection have been widely studied, they are often implemented as standalone solutions. The absence of an integrated platform that combines these capabilities limits the effectiveness of current systems in delivering proactive and comprehensive safety measures.

1.2 Research Objectives and Contributions

To address the limitations of existing school transportation systems, this paper proposes **MUTMA'INN**. As shown in Figure 1, the framework combines in-bus cameras, edge-based AI processing, cloud services, and mobile applications to support student protection and driver monitoring. Unlike conventional tracking systems, it enables real-time event analysis, automated attendance logging, rapid alert generation, and synchronized access to live transportation data.

The main objective of this work is to develop a unified system that can automatically record student boarding and alighting, continuously monitor the driver's physiological and behavioral state, detect unsafe in-bus situations, and provide real-time visibility to administrators, drivers, and parents through connected mobile interfaces.

The main contributions of this work are as follows. First, it develops an edge-based driver monitoring module that processes live video from a driver-facing camera to estimate heart rate using contactless rPPG and support real-time analysis of the driver's condition. Second, it implements a smart-gate facial recognition mechanism to automatically record student boarding and alighting, reducing reliance on manual attendance and minimizing human error. Third, it integrates computer vision-based bus safety analysis for occupancy checking and post-trip verification to detect unsafe situations, such as a student remaining inside the bus after the trip ends. Fourth, it connects edge-generated data to Firebase cloud services for real-time synchronization, centralized record management, alert delivery, and role-based access. Fifth, it designs a cross-platform Flutter mobile application that enables administrators, drivers, and parents to monitor attendance, alerts, and trip activity through role-specific interfaces. Finally, it presents a unified end-to-end architecture that combines driver monitoring, student attendance automation, bus safety verification, cloud synchronization, and mobile access into a cohesive framework for proactive school transportation safety management.

2. RELATED WORK

School transportation safety has received increasing attention because of the need to protect students during daily trips and improve communication between schools, parents, and drivers.

Existing smart school bus systems commonly use GPS, IoT devices, and mobile applications to track bus locations and notify parents about boarding, arrival, or delays [7]. These systems improve visibility and reduce communication gaps between stakeholders. However, most of them remain focused on tracking rather than proactive safety. They usually do not verify the driver's physical condition, detect unsafe driver behavior, or confirm whether a student has been left inside the bus after the trip ends. This creates a need for systems that go beyond location monitoring and provide real-time safety intelligence.

Computer vision has also been widely used in biometric identification and intelligent transportation systems. Face recognition, supported by recent deep learning approaches such as convolutional neural networks and embedding-based models, has become a reliable method for contactless identification [8]. It is commonly applied in attendance systems and access control because it reduces manual effort and human error. In the context of school buses, face recognition can support automated student boarding and alighting records. However, bus environments introduce practical challenges such as changing lighting, motion blur, camera angle variation, and partial occlusion. Therefore, face recognition in this setting must be implemented as a lightweight and real-time pipeline suitable for edge processing at the bus gate.

Driver monitoring is another important area related to school transportation safety. Existing driver monitoring systems use computer vision techniques such as eye tracking, head-pose estimation, facial landmark analysis, and facial expression recognition to detect fatigue, distraction, or unsafe behavior. In parallel, remote photoplethysmography allows non-contact estimation of physiological signals such as heart rate from video by detecting subtle skin color changes caused by blood flow [9]. These technologies are useful for identifying early signs of fatigue, stress, or abnormal driver condition. However, both visual and physiological monitoring can be affected by real-world conditions such as motion, vibration, illumination changes, and camera noise. This is especially relevant inside moving vehicles, where the system cannot depend on controlled recording conditions.

Object detection has also become a key component in intelligent safety systems. Models such as YOLO provide real-time object detection capabilities and have been widely used in transportation applications [10]. While these models are often applied to road-scene analysis, vehicle detection, and pedestrian monitoring, their use inside school buses for post-trip safety verification remains limited. This is an important gap because one of the most critical risks in school transportation is the possibility of a child being left inside the bus after the trip ends. A post-trip object detection module can support automated inspection and reduce reliance on manual checking alone.

IoT, edge computing, and cloud-based communication provide the infrastructure needed to connect these safety functions into a practical system. Edge computing allows time-sensitive video and signal processing tasks to be performed locally, reducing latency and limiting unnecessary data transmission [11]. Cloud services, on the other hand, support centralized storage, real-time synchronization, and communication with mobile applications [12]. A hybrid edge–cloud architecture is therefore suitable for school transportation because urgent decisions can be handled locally, while processed results and alerts can be shared with parents, drivers, and administrators through a mobile platform.

Table 1: Research Gap Comparison Table

Research Reference	Student Tracking	Driver Vitals (rPPG)	Emotion Detection	Post-Trip Scan (YOLO)	Edge Computing	Integrated Platform
Al-Quraishi [6]	×	✓	×	×	×	×
Chen et al. [8]	✓	×	×	×	×	×
Alghamdi et al. [9]	×	×	×	×	✓	×
Li et al. [10]	×	✓	✓	×	×	×
Redmon et al. [11]	×	×	×	✓	×	×
Mutma'inn (Ours)	Full	Full	Full	Full	Full	Full

Despite advances in smart transportation, driver monitoring, physiological sensing, object detection, and cloud communication, existing studies often address these functions as separate components. As summarized in Table 1, prior work typically covers selected capabilities, such as student tracking, driver vitals, drowsiness detection, or object detection, but does not provide a fully integrated school bus safety framework. This limits the ability of existing systems to deliver real-time and proactive protection.

MUTMA'INN addresses this gap through a unified edge–cloud framework that combines facial recognition for student attendance, rPPG-based heart-rate estimation, EAR-based drowsiness detection, emotion analysis, and YOLO-based post-trip occupancy scanning. Our work differs from prior work by integrating multiple AI-driven safety modules into one practical end-to-end school transportation monitoring platform.

3. METHODOLOGY

This section explains the design and implementation of the proposed system. It focuses on how the different components work together, including student attendance, driver health monitoring, and driver state analysis. The methodology combines computer vision and signal processing techniques to provide a reliable and real-time safety solution.

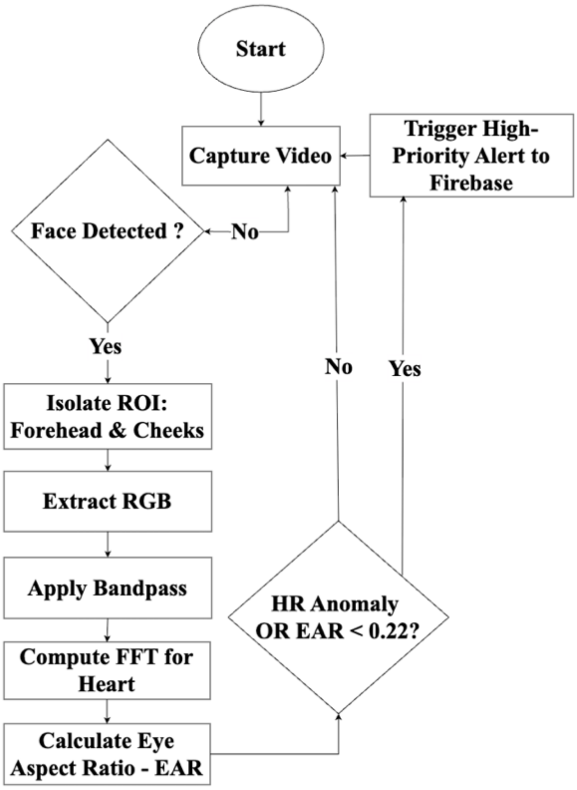

Figure 2: Flowchart of the driver monitoring system and rPPG pipeline: The process begins with video capture, followed by face detection. When a face is detected, the region of interest (ROI) is extracted to obtain RGB signals, which are then filtered and analyzed using FFT to estimate heart rate. In parallel, they Eye Aspect Ratio (EAR) is calculated to assess drowsiness. If a heart rate anomaly or low EAR is detected, a high-priority alert is triggered and sent to Firebase.

Table 2: Comparative analysis of the proposed Edge-Cloud architectural framework against traditional approaches.

Our Proposed Edge-Cloud	**Cloud Only**	**Edge Only**	**Feature**
Very Low (real-time)	High (Depends on Net)	Low (Fast)	**Latency**
Partial/Robust	Required always	Not required	**Connectivity**
High (Local Processing)	Low	High	**Privacy**
High	High	Limited	**Scalability**

As illustrated in Table 1, the Edge-Cloud hybrid framework achieves better performance than traditional single-tier systems by maintaining low-latency requirements while scaling. The system maintains its real-time response capabilities through Edge-based mission-critical task inference for rPPG-based fatigue detection and facial recognition. The Cloud integration provides a strong solution for data synchronization and historical analysis and long-term storage because it overcomes the resource limitations of Edge devices. The hybrid design improves system reliability and safety while preventing bandwidth bottlenecks in the school bus transportation monitoring system

3.1 Student Attendance through Facial Recognition Pipeline

This subsection describes how the system automatically records student attendance using facial recognition. The process is designed to work efficiently in real-time, even with changes in lighting or movement at the bus gate.

First, the system detects faces from the camera feed using lightweight models such as MTCNN or Haar Cascades [13] [14]. These models are chosen because they are fast and suitable for real-time use.

Once a face is detected, it is aligned and processed before being passed to a deep learning model. This model extracts a 128-dimensional embedding that represents the unique features of the face.

To identify the student, the system compares this embedding with stored embeddings in the database using cosine similarity. If the similarity score is higher than a predefined threshold ($\lambda > 0.85$), the student is successfully recognized. The system then records the event as either boarding or alighting, depending on the situation. This approach reduces manual work and helps avoid common errors in attendance tracking.

3.2 Non-Contact Driver Vital Monitoring rPPG

The core physiological shield extracts the Blood Volume Pulse (BVP). The algorithmic steps include:

1. ROI Localization: In order to prevent any possible artifacts due to blinks or talking by the subject, the localization process extracts two ROIs, namely, the forehead and the top part of cheeks [15] (see Figure 1).
2. Spatial Averaging: In each video frame t, the RGB pixel intensities of these ROIs are averaged to produce raw color signals denoted as CR(t), CG(t), CB(t).
3. Signal Detrending and Filtering: There will be a lot of unwanted noise in the obtained signals due to vehicle vibration [16]. The use of Butterworth bandpass filtering [17] with frequencies at 0.75 Hz and 2.5 Hz (or more specifically, 45 bpm and 150 bpm) removes such noise.
4. Pulse Rate Determination: After the filtering process is complete, fast Fourier transform (FFT) [18] is used to convert the signal from time domain to frequency domain and the Heart Rate (HR) is derived from the frequency with the maximum power spectrum (fmax):

$$HR = 60 \times fmax$$

3.3 Driver Emotion and Fatigue Detection

To detect driver drowsiness, the system monitors the ocular region using facial landmarks. Specifically, six landmarks around the eye, denoted as p_1 to p_6, are used to compute the Eye Aspect Ratio (EAR) [19] for each video frame:

$$EAR = (||p2 - p6|| + ||p3 - p5||) / (2||p1 - p4||)$$

Figure 3 illustrates the EAR landmarks for open- and closed-eye states. EAR is used as a geometric indicator of eye openness. Under normal conditions, the EAR remains relatively stable when the eyes are open. As the eyes close, the vertical distances between the eyelids decrease, resulting in a lower EAR value. This property makes EAR an effective and computationally efficient measure for real-time drowsiness detection.

Figure 3: Eye Aspect Ratio (EAR) landmarks showing the difference between open-eye and closed-eye states.

To improve reliability, the system combines the EAR threshold with temporal logic to distinguish prolonged eye closure from normal blinking. In the current implementation, a drowsiness alert is generated when the EAR remains below 0.22 for 10 consecutive frames (approximately 0.33 seconds). With this approach, **MUTMA'INN** system achieves its accident prevention system through its combined geometric and temporal analysis method which achieves precise safety alert timing through its connection to the Firebase dashboard.

4. OUR SYSTEM ARCHITECTURE

The architecture follows a three-tier structure, consisting of the mobile application layer, the edge computing layer, and the cloud layer. Each layer plays a specific role, and together they form a unified system

4.1 Flutter Mobile Application Tier

The **MUTMA'INN** ecosystem adopts a decoupled three-tier architecture to ensure low-latency processing, reliable communication, and cross-platform accessibility. Its presentation layer is developed using the Flutter UI toolkit [20]. which provides near-native performance across both iOS and Android platforms. Through efficient state management, the application supports continuous synchronization of real-time data streams. As illustrated in Figure 4, administrators can access a centralized dashboard displaying live bus locations, driver vital status, and high-priority alerts, while parents can monitor their children's boarding and alighting events in real time. Drivers interact with a simplified interface that supports trip management and route monitoring.

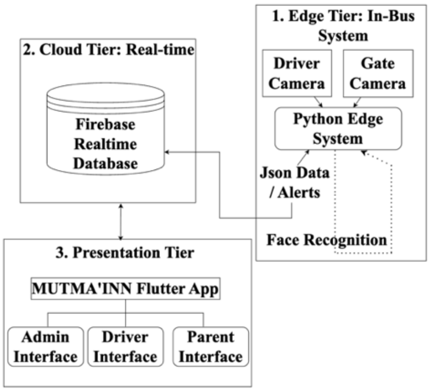

Figure 4: Flowchart of the Proposed System Architecture of MUTMA'INN. The architecture follows a three-tier design consisting of an edge layer, a cloud layer, and a presentation layer. At the edge tier, in bus cameras (driver and gate) feed data into a Python based edge system for real time processing, including face recognition and driver monitoring. Processed data and alerts are transmitted to the Firebase Realtime Database, enabling continuous cloud synchronization. Finally, the presentation layer provides a unified Flutter based mobile application for administrators, drivers, and parents to monitor and interact with the system in real time.

4.2 Python-Based In-Bus Edge System

This layer, which uses multithreading to manage two separate video feeds without frame blocking, is installed on onboard computing hardware [11]. While Thread 'B' processes the driver-facing camera for continuous rPPG and emotion analysis, Thread 'A' processes the gate-facing camera for student facial recognition. Payload sizes sent via mobile networks are significantly reduced when inferences are made at the edge.

4.3 Firebase Cloud Synchronization Services

Firebase serves as the real-time intermediary [21] The solution avoids the cost associated with conventional HTTP polling by using WebSockets and a NoSQL document structure. All connected Flutter clients immediately receive an asynchronous status update when the edge system reports a physiological abnormality and writes to the database.

4.4 Physical Actuation and Hardware Layer

The system utilizes an ESP32 microcontroller to interface AI-driven decisions with physical hardware within the bus. For smart gate control, successful student identification via the edge-based facial recognition pipeline triggers a signal to the ESP32, which actuates a servo motor to open the gate at a 90-degree angle. Verified student check-ins are visually indicated by a green LED, while unrecognized or unverified entries activate a yellow LED.

The driver health signaling system operates in tandem with the physical layer to provide immediate visual feedback regarding the driver's condition. A red LED activates to signal critical states, such as prolonged drowsiness (based on EAR threshold violations) or physiological anomalies where heart rate abnormalities exceed the 10-second threshold. During normal operations, a white LED remains illuminated to indicate a safe state. This hardware integration creates a local, responsive feedback loop, ensuring that safety alerts are physically acknowledged within the bus environment in real time.

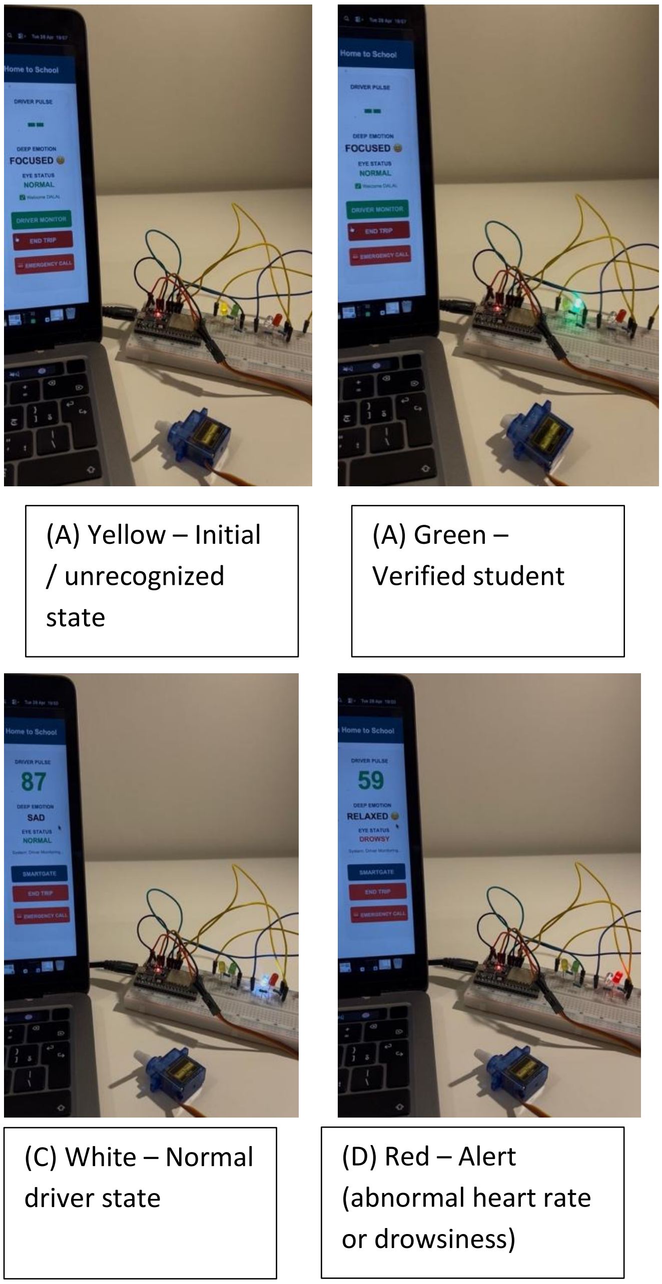

Figure 5: LED-based visual feedback of the system under different operating conditions: (a) green LED indicating a verified student check-in, (b) yellow LED representing the initial system state or unrecognized input, (c) red LED indicating an alert condition due to abnormal heart rate or drowsiness, and (d) white LED representing normal operation.

5. IMPLEMENTATION AND RESULTS

This section presents the implementation of the proposed **MUTMA'INN** framework and its operational setup. It first describes the software environment used to develop the edge computing and mobile application components, then outlines the threshold-based alert mechanism and the evaluation setup adopted in simulated transit scenarios. Finally, it presents the mobile application interfaces developed to support administrators, parents, and drivers.

5.1 Software Development Environment

The edge computing pipeline was implemented using Python 3.10 to support real-time video processing within the bus environment. Computer vision techniques were applied to develop an automated student attendance system based on facial recognition, enabling a fully contactless identification process. NumPy was used to perform the required signal processing for heart rate estimation.

The mobile application was developed using Flutter SDK 3.x to provide a role-based interface for different users. Firebase Realtime Database was utilized to enable real-time communication between the edge computing system and the mobile application.

5.2 Operational Thresholds and Alert Mechanism

To prevent alert fatigue among school administrators, the system employs threshold-based logic before escalating physiological warnings to the cloud.

In the current implementation, alerts are triggered only when the driver's heart rate falls outside the normal range for a sustained period.

Table 3. Heart Rate Thresholds for Driver Monitoring

Parameters	Normal Range	Critical Threshold (Alert Trigger)	Duration Required
Heart Rate (HR)	60-100 bpm	< 55 bpm OR > 110 bpm	> 10 seconds

5.3 System Evaluation Setup

The proposed system was tested in simulated transit scenarios to examine its operation in a school transportation setting. This evaluation focused on the main functional modules of the framework, including student attendance, driver health monitoring, drowsiness detection, emotion analysis, and cloud-based communication.

The attendance module uses facial recognition to match the captured student face during boarding and alighting with the corresponding face record stored in Firebase, enabling automated attendance logging. The driver monitoring module estimates heart rate from the driver-facing camera using contactless rPPG. In the current implementation, the first heart rate reading becomes available after approximately 4–5 seconds (120 frames), which are required to fill the FFT buffer, and subsequent readings are updated every 2 seconds (60 frames). The drowsiness detection module computes the EAR ratio every 5 frames, and a drowsiness condition is flagged when the EAR remains below 0.22 for 10 consecutive frames (approximately 0.33 seconds). In parallel, the emotion analysis module is performed every 60 frames (approximately 2 seconds) to monitor the driver's emotional state and highlight potentially negative conditions. To support system-wide communication, Firebase Realtime Database synchronizes outputs from the edge system with the mobile application, enabling continuous updates and alert delivery [22].

5.4 Quantitative Performance Evaluation

The results show that the proposed system can estimate heart rate with a reasonable level of accuracy under real-time conditions. Some differences between the two measurements were observed, mainly due to motion artifacts, lighting changes, and the nature of camera-based monitoring.

In addition to accuracy evaluation, the system's decision-making capability was tested using threshold-based conditions. The system considers heart rate values below 55 bpm or above 110 bpm as abnormal conditions for more than 10 seconds.

To verify this behavior, controlled and simulated scenarios were used. Table 4 presents the results of these tests, where the system correctly triggered alerts in abnormal cases and remained stable during normal conditions.

Overall, the results confirm that the proposed system provides reliable heart rate estimation and effective real-time alerting in dynamic environments.

Table 4: Comparison between rPPG-based heart rate measurements and smartwatch readings at different time intervals.

Time (s)	RPPG (bpm)	Smart Watch (bpm)	Error (%)	Accuracy (%)
0	76	78	2.6	97.4
2	72	76	5.3	94.7
4	106	76	39.5	60.5
6	90	77	16.9	83.1
8	86	76	13.2	86.8
10	83	73	13.7	86.3
12	73	73	0.0	100.0
14	65	73	11.0	89.0
16	81	73	11.0	89.0
18	81	78	3.8	96.2

The output indicates that the accuracy level of the suggested method was found to be approximately 88.3% as compared to the accuracy level attained using the Apple Watch Series 9.

A few discrepancies in the results have been observed, mainly due to some sharp peaks observed in the rPPG signal. This was normal since external factors such as motion, ambient light levels, and other signal disturbances could lead to variations in the measurement of heart rate using a camera sensor.

Generally, all of the readings were fairly similar, thereby indicating that the system works consistently and can hence be employed in dynamic environments.

To further evaluate the system, the alert mechanism was tested under different conditions, including abnormal heart rate and drowsiness detection.

Table 5: Validation of the alert mechanism based on heart rate and drowsiness conditions, including combined scenarios

Condition	Heart Rate (bpm)	Drowsiness	Duration (s)	System Response
Normal	75	No	10>	No Alert
High HR	115	No	10<	Alert Triggered
Low HR	50	No	10<	Alert Triggered
Drowsiness	80	Yes	10<	Alert Triggered
High/Low HR + Drowsiness	115/50	Yes	10<	Alert Triggered

The system triggers an alert whenever either the abnormal heart rate condition or the drowsiness condition persists for over 10 seconds. Both conditions are considered individually; hence an alert can be issued when any of the two happens.

Furthermore, whenever both conditions happen concurrently, the system issues an alert as expected. As depicted in Table 5, the system performs as expected in all the test cases.

5.5 Robustness Testing

To evaluate the robustness of the MUTMA'INN system, several real-world scenarios were simulated to assess system performance under varying environmental and operational conditions. These scenarios aim to ensure that the system remains reliable, accurate, and responsive in practical deployment settings.

First, the system was tested under different lighting conditions, including low-light environments, bright daylight, and variable illumination inside the bus. The facial recognition module maintained acceptable accuracy, although minor performance degradation was observed under extremely low lighting conditions.

Second, motion-related challenges were evaluated by simulating bus movement and camera vibrations. The rPPG-based heart rate estimation demonstrated stable performance due to the applied filtering techniques; however, sudden excessive motion introduced minor noise in the signal.

Third, variations in student behavior were considered, such as partial face occlusion, different head angles, and fast movement during boarding. The system was still able to detect and recognize faces in most cases, confirming its adaptability in dynamic scenarios.

Additionally, driver monitoring was tested under different facial expressions and head positions. The EAR-based drowsiness detection remained effective in identifying prolonged eye closure, while the emotion detection module successfully captured general emotional states.

Finally, network variability was examined by simulating intermittent internet connectivity. The edge computing layer ensured continuous operation of critical functionalities, while

Firebase synchronization resumed automatically once the connection was restored.

These results demonstrate that the proposed system is capable of operating under realistic conditions and is suitable for real-world deployment in school transportation environments.

5.6 User Interface Mobile Application

The system includes a mobile application with three main user roles: administrator, parent, and driver. Each interface is designed to meet the requirements of its intended user. The administrator interface supports system monitoring and management, the parent interface provides access to student tracking and attendance information, and the driver interface assists with trip operation and execution. Figure 5 presents selected mobile application screens, including the route tracking interface, the emergency alerts section, and the main user dashboard, to illustrate how real-time transportation information is delivered to users. In addition, Figure 6 shows the driver drowsiness detection interface, where eye closure is monitored and the driver's state is displayed in real time. Finally, Figure 7 illustrates the object detection module used after the trip ends to verify that no students remain inside the bus.

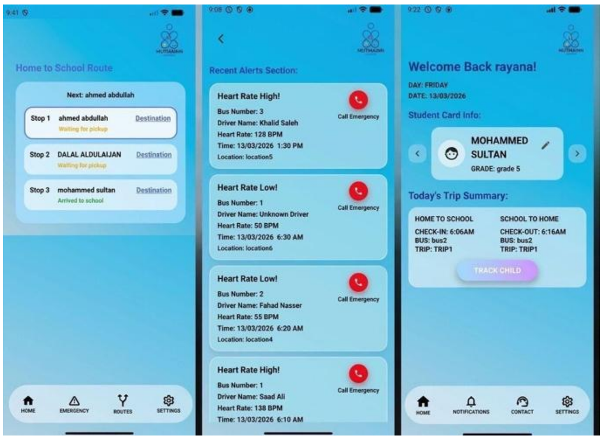

Figure 6: Comprehensive overview of the MUTMA'INN mobile application interfaces. From left to right: (1) Route Tracking Interface, displaying real-time bus stops and student boarding status; (2) Emergency Alerts Section, highlighting critical driver health notifications (Heart Rate Monitoring) with quick-access emergency call features; (3) Main User Dashboard, providing a summarized daily trip log and student identification details.

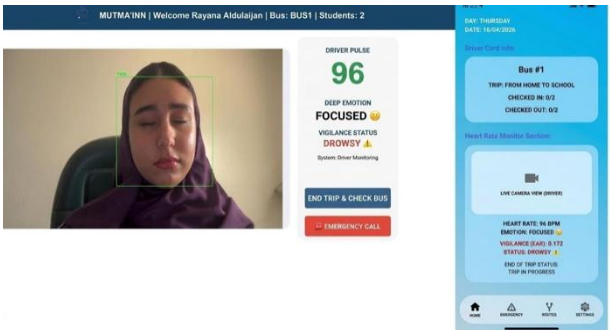

Figure 7: The system includes a drowsiness detection feature that monitors eye closure and displays the driver's state in real time.

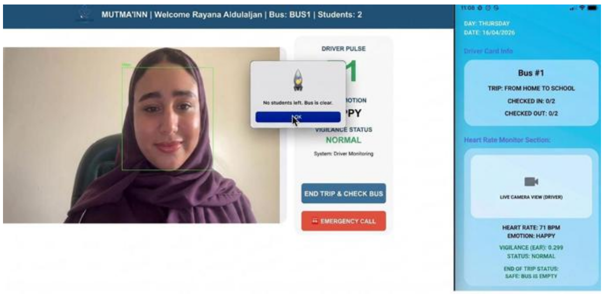

Figure 8: Object detection performed after the driver ends the trip to verify that no students remain on the bus.

5.7 Hardware Setup

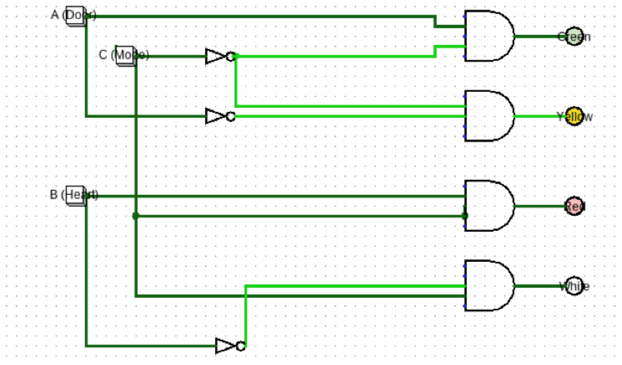

Figure 9: The combinational circuitry represented consists of 3 inputs(A= Door, B = Heart, and C= Mode) to control four outputs(Green, Yellow, Red, and White) using basic combine gates(AND and NOT). The input C would give a selection mode to the circuit. If input C = 0(purpose= door) AND A = 1 Green would be energized. If A = 0yellow would be activated. If C = 1 (Purpose= Heart) When B = 1, Red energizes; and when B = 0, White energizes. Therefore we have a simple example of being able to separate two related subsystems by a single selection device and control multiple outputs through basic combinational logic.

6. CONCLUSION AND FUTURE WORK

This paper presented **MUTMA'INN**, an AI-driven edge–cloud framework for intelligent school transportation safety. The proposed system integrates facial recognition for automated student attendance, contactless rPPG for vital driver monitoring, and computer vision techniques for drowsiness and in-bus safety analysis within a unified architecture. By combining a Python-based edge system with a Flutter mobile application and Firebase cloud services, the framework enables real-time monitoring, alert generation, and continuous communication among administrators, drivers, and parents. Unlike conventional GPS-based systems, **MUTMA'INN** adopts a proactive approach by supporting real-time decision-making and early detection of potential safety risks.

Future work will focus on enhancing the robustness and reliability of the system under real-world conditions. Advanced deep learning-based rPPG models will be explored to improve signal quality and reduce the impact of motion and illumination variations in dynamic environments. In addition, further validation in real deployment scenarios and the incorporation of quantitative performance evaluation will be considered. Finally, integrating the system with national educational and transportation platforms could support large-scale analytics and contribute to the smart mobility objectives of Saudi Vision 2030.

Ethical Note. The displayed images are included with prior consent from the participant and are used for demonstration purposes only.

7. ACKNOWLEDGMENTS

We extend our most sincere thanks to ***Dr. Saad Alhasaniah*** who works as a cardiologist and electrophysiologist at Johns Hopkins Aramco Healthcare for his essential medical advice about vital signs. The rPPG algorithms underwent precise development through his expertise which led to better physiological alert systems and helped our system achieve compliance with medical requirements.

The special recognition of ***Dr. Nahier Ghaleb Aldhafeeri*** who works as Associate Professor and Vice Dean for Graduate Affairs at the College of Computer Science and Information Technology requires our appreciation. The project core idea and main development methods were determined through his assistance during the initial project period.

8. REFERENCES

[1] W. H. Organization, "Global status report on road safety 2023," World Health Organization, Geneva, 2023.

[2] G. A. f. Statistics, "Road Transport Statistics 2024," General Authority for Statistics, Riyadh, 2024.

[3] M. o. Education, "Student Transport Services," Ministry of Education, [Online]. Available: https://www.moe.gov.sa/en/knowledgecenter/eservices/pages/studentransport.aspx. [Accessed 10 April 2026].

[4] S. P. Agency, "TGA Urges Educational Transport Sector to Comply with Safety Standards," Saudi Press Agency, 2024. [Online]. Available: https://www.spa.gov.sa/en/N2153260. [Accessed 10 April 2026].

[5] T. G. Authority, "TGA and Vision 2030," Transport General Authority, [Online]. Available: https://www.tga.gov.sa/en/AboutTGA/TGAand2030Vision. [Accessed 10 April 2026].

[6] M. S. Al-Quraishi, "Technologies for detecting and monitoring drivers' states," *Heliyon,* vol. 10, no. 8, p. e15623, 2024.

[7] L. Chen, H. Ai, Z. Zhuang and C. Shang, "Real-Time Multiple People Tracking with Deep Learning-Based Re-Identification," *IEEE Access,* vol. 12, 2024.

[8] J. Deng, J. Guo, N. Xue and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 44, no. 10, p. 5962–5979, 2022.

[9] Z. Li, X. Wu, C. Álvarez Casado, V. Lindholm, K. Mikkonen, Z. Xia, X. Feng and M. Bordallo López, "A comprehensive survey on contactless vital sign monitoring using vision-based, radio-based, and fusion approaches," *Neurocomputing,* vol. 674, p. 132877, 2026.

[10] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016.

[11] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal,* vol. 3, no. 5, p. 637–646, 2016.

[12] M. Satyanarayanan, "The Emergence of Edge Computing," *Computer,* vol. 50, no. 1, p. 30–39, 2017.

[13] K. Zhang, Z. Zhang, Z. Li and Y. Qiao, "Joint Face Detection and Alignment Using Multi-task Cascaded Convolutional Neural Networks," *IEEE Signal Processing Letters,* vol. 23, p. 1499–1503, 2016.

[14] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Kauai, HI, USA, 2001.

[15] G. de Haan and V. Jeanne, "Robust Pulse Rate From Chrominance-Based rPPG," *IEEE Transactions on Biomedical Engineering,* vol. 60, no. 10, p. 2878–2886, 2013.

[16] M. Lewandowska, J. Rumiński, T. Kocejko and J. Nowak, "Measuring Pulse Rate with a Webcam – A Non-contact Method," *Biomedical Engineering Online,* vol. 10, no. 10.1186/1475-925X-10-82, p. 82, 2011.

[17] M.-Z. Poh, D. J. McDuff and R. W. Picard, "Advancements in Noncontact, Multiparameter Physiological Measurements Using a Webcam," *IEEE Transactions on Biomedical Engineering,* vol. 58, no. 1, p. 7–11, 2011.

[18] A. V. Oppenheim and R. W. Schafer, Discrete-Time Signal Processing, Upper Saddle River, NJ, USA: Pearson, 2010.

[19] T. Soukupová and J. Čech, "Real-Time Eye Blink Detection using Facial Landmarks," in *Proceedings of the Computer Vision Winter Workshop (CVWW)*, Rimske Toplice, Slovenia, 2016.

[20] M. Palmieri, I. Singh and A. Cicchetti, "Comparison of Cross-Platform Mobile Development Tools," in *Proceedings of the IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, San Francisco, CA, USA, 2012.

[21] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall and W. Vogels, "Dynamo: Amazon's Highly Available Key-value Store," in *Proceedings of the ACM Symposium on Operating Systems Principles (SOSP)*, Stevenson, WA, USA, 2007.

[22] A. Alghamdi, M. Alotaibi and S. Alqahtani, "Real-Time Driver Drowsiness Detection Using Computer Vision Techniques," *Sensors,* vol. 23, no. 5, 2023.