A

Project Report

on

Local Bus Tracking App

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2023-24

in

COMPUTER SCIENCE AND ENGINEERING

By

Shivam Tiwari (2000290100147)

Shashank Patel (2000290100142)

Under the supervision of

Dr. Parita Jain

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2024

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Date:

SIGNATURE:	**SIGNATURE:**
NAME: SHIVAM TIWARI	**NAME:** SHASHANK PATEL
ROLL NO.: 2000290100147	**ROLL NO.:** 2000290100142
DATE:	**DATE:**

CERTIFICATE

This is to certify that Project Report entitled "**Local Bus Tracking App**" which is submitted by **Shashank Patel and Shivam Tiwari** in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science & Engineering of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

.

Dr. Parita Jain	**Dr. Vineet Sharma**
(Associate Professor)	**(Head of Department)**

Date:

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to **Dr. Parita Jain** Associate Professor, Department of Computer Science & Engineering, KIET, Ghaziabad, for her constant support and guidance throughout the course of our work. Her sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only her cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of **Dr. Vineet Sharma**, Head of the Department of Computer Science & Engineering, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Date:

SIGNATURE:

NAME: SHIVAM TIWARI
ROLL NO.: 2000290100147
DATE:

SIGNATURE:

NAME: SHASHANK PATEL
ROLL NO.: 2000290100142
DATE:

ABSTRACT

Our project endeavors to create a mobile application that revolutionizes public transportation services by seamlessly integrating counting number of passengers in the bus, sos ,and seat availability checking functionalities. Through the use of mapping technology, the system dynamically tells users about route of buses. Developed with React Native, the application boasts a user-friendly interface, allowing passengers to effortlessly view buses on a map and ascertain seat availability. Additionally, the inclusion of an SOS feature further enhances the overall passenger experience, ensuring safety during emergencies.

In tackling challenges such as congestion and inefficiency in bus services, our project introduces a novel approach to bus monitoring and passenger flow prediction. By harnessing APIs for bus routes and implementing machine learning algorithms for seat occupancy checks, the system aims to deliver real-time updates on bus routes and seat availability. The SOS feature serves as an additional layer of safety, guaranteeing passenger well-being during unforeseen circumstances.

The implementation of MobileNet SSD architecture for real-time seat vacancy detection underscores the system's efficacy in analyzing bus interior images using machine learning algorithms. Moreover, the inclusion of bus routes enables users to track nearby buses on a map interface, facilitating transparent journey planning. Detailed discussions on the architecture of MobileNet SSD and its seamless integration with the Single Shot Multibox Detector (SSD) framework illustrate its utility for efficient object detection.Moreover, the integration of MongoDB serves as a vital component in storing real-time data on the number of passengers aboard each bus. This feature enables users to access information regarding bus occupancy levels, empowering them to make informed decisions about their commute.

Overall, the development and implementation of this application signify a significant advancement in public transportation technology. With its goal of providing passengers with a seamless and efficient bus booking experience, while also contributing to the enhancement of public transportation systems globally, the project embodies innovation and progress in the field.

TABLE OF CONTENTS

	Page No.
DECLARATION	ii
CERTIFICATE	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
LIST OF FIGURES	viii
LIST OF ABBREVIATIONS	ix
CHAPTER 1 INTRODUCTION	1
1.1 INTRODUCTION	1
1.2 PROJECT DESCRIPTION	
1.2.1 THE KEY FEATURES OF APPLICATION	2
1.2.2 THE TARGET USERS OF APPLICATION	2
1.2.3 THE APPLICATION OFFERS SEVERAL BENEFIT TO ITS USERS	2
1.2.4 SEVERAL TECHNOLOGIES AND FRAMEWORKS WERE UTILIZED	3
CHAPTER 2 LITERATURE REVIEW	4
CHAPTER 3 PROPOSED METHODOLOGY	8
3.1 REQUIREMENT GATHERING AND ANALYSIS	8
3.2 DESIGN	10
3.3 IMPLEMENTATION	13
3.3.1 REACT NATIVE FOR MOBILE APPLICATION DEVELOPMENT	14
3.3.2 GOOGLE MAPS API FOR ROUTE INFORMATION	15
3.3.3 MACHINE LEARNING FOR PASSENGER PREDICTION	16

3.3.4 MONGODB INTEGRATION FOR PASSENGER DATA STORAGE 21

CHAPTER 4 RESULTS AND DISCUSSION 24

CHAPTER 5 CONCLUSIONS AND FUTURE SCOPE 27

REFERENCES 29

APPENDEX1 32

LIST OF FIGURES

Figure No.	Description	Page No.
1	UML Diagram of Smart Bus System	9
2	Diagram of Smart Bus Application	10
3	Design of ML Model predicting people inside bus using snapshot	11
4	Diagram of SOS feature in Smart Bus Application	12
5	Screenshots of Smart Bus Application made using React Native	15
6	Screenshot of Route of a bus fetched using google maps	16
7	Architecture of SSD MobileNet model	17
8	Construction of a SSD model for object detection tasks	18
9	Depthwise Separable Convolution and MobileNet model loading	19
10	Running ML model to test the number of people pass from location	21
11	Screenshot of Query Result	22
12	Screenshot of the key value pairs displayed	23

LIST OF ABBREVIATIONS

SOS	Save Your Souls
UCD	User Centered Design
SSD	Single Shot Multibox Detector
ETA	Etimated Time of Arrival
RTBTS	Real-Time Bus Tracking System
IOT	Internet of Things
API	Application Programming Interface
ML	Machine Learning
UI	User Interface
YOLO	You Only Look Once

CHAPTER 1

1.1 INTRODUCTION

Efficient urban transportation systems play a pivotal role in the functioning and development of cities worldwide. Among various modes of transportation, bus services serve as a lifeline, providing essential connectivity and mobility to urban residents. In the context of bustling cities, where populations are concentrated and traffic congestion is a constant challenge, the efficiency of bus services becomes even more crucial.Traditional bus services within cities encounter numerous challenges that hinder their effectiveness and reliability. Issues such as irregular schedules, overcrowded buses, long wait times, and inadequate information dissemination contribute to passenger dissatisfaction and undermine the overall efficiency of the transportation system. These challenges not only inconvenience passengers but also result in inefficiencies in resource utilization and environmental impacts due to increased emissions from traffic congestion.

Addressing these challenges requires innovative solutions that leverage modern technologies and novel approaches to enhance the quality and efficiency of bus services. By integrating advanced technologies such as ML technology, mobile applications, and online services, it is possible to transform traditional bus services into more efficient, convenient, and user-centric systems. Such innovations can improve the accuracy of schedule adherence, provide real-time information to passengers, optimize route planning, and streamline ticketing and payment processes.

The objective of this project is to develop a comprehensive mobile-based solution for urban bus services, specifically tailored to address the challenges faced by both passengers and service providers. The scope of the project encompasses the design and implementation of a mobile application that offers features such as real-time bus tracking, online ticket booking, and emergency assistance functionalities. By leveraging cutting-edge technologies and user-centric design principles, the project aims to enhance the overall efficiency, convenience, and accessibility of urban bus transportation, ultimately contributing to the improvement of urban mobility and quality of life for city residents.

1.2 PROJECT DESCRIPTION

The mobile-based bus tracking and online service application developed in this project offer a comprehensive solution to enhance urban transportation experiences. Through a user-friendly interface and advanced functionalities,the application aims to revolutionize the way passengers interact with bus services within cities.

1.2.1 The key features of application:

a. **Real-time Passenger Prediction:** Users can track the number of passenger in buses in real-time using app. This feature provides accurate information on seat vacancy, enabling passengers to plan their journeys more effectively and reduce waiting times at bus stops.

b. **SOS Functionalities:** In case of emergencies, passengers can trigger an SOS(Save your souls) alert directly from the application. This feature sends notifications to emergency contacts or relevant authorities, ensuring prompt assistance and enhancing passenger safety.

c. **Seat Availability Prediction**: Leveraging machine learning algorithms, the application predicts the availability of seats on upcoming bus trips. By analyzing historical data and current passenger loads, the system provides users with insights into whether buses are likely to have available seats or be crowded, enabling informed decision-making and optimizing passenger comfort.

1.2.2 The target users of application:

a. **Commuters:** Individuals who rely on bus services for their daily transportation needs, including office commuters, students, and tourists.

b. **Service Providers:** Bus operators and transportation authorities responsible for managing and optimizing bus services within cities.

1.2.3 The application offers several benefits to its users:

a. **Convenience:** Users can access real-time bus information and request emergency assistance from a single platform, enhancing convenience and efficiency in urban transportation.
b. **Time-saving:** Real-time bus info about vacant seats and routes help users save time by reducing wait times
c. **Safety:** The SOS functionalities provide an added layer of safety and security for passengers, ensuring prompt assistance during emergencies.

1.2.4 Several technologies and frameworks were utilized:

a. **React Native**: The application was built using React Native, a cross-platform framework for mobile app development. This framework enables the development of a single codebase that runs on both iOS and Android platforms, reducing development time and effort.
b. **Machine learning:** Implemented for predicting seat availability based on historical data and real-time passenger on the bus suing snapshot of the bus, enhancing the accuracy and reliability of information provided to users.
c. **Maps:** Google Maps was utilized to visually showcase bus routes, offering users accurate information about bus movements within the city, thus enhancing navigation and journey planning for commuters.
d. **MongoDB:** MongoDB serves as the robust data storage solution within the Smart Bus application, effectively managing information regarding the number of passengers aboard buses. This database system efficiently stores real-time passenger count data, enabling seamless access and retrieval for users seeking information about bus occupancy.

Overall, the mobile-based bus tracking and online service application offer a seamless and efficient solution for urban transportation, catering to the needs of both passengers and service providers while leveraging modern technologies for enhanced functionality and user experience.

CHAPTER 2

LITERATURE REVIEW

The integration of Internet of Things (IoT) technologies in public transportation systems has revolutionized various aspects of passenger experience and operational management. It explores the research landscape concerning smart bus seat occupancy detection systems, focusing on the development, implementation, and evaluation of such systems.In recent years, researchers have increasingly focused on developing innovative solutions to address the challenges faced by passengers in locating unoccupied seats and obtaining real-time information about seat occupancy in buses. Sharma and Sawant [3] proposed a smart bus seat occupancy detection system leveraging IoT principles and pressure pad sensors. The system aimed to enhance passenger convenience, reduce time wastage, and optimize seat utilization through real-time monitoring of seat occupancy status. The methodology employed in the development of smart bus seat occupancy detection systems typically involves the integration of hardware components such as pressure pad sensors and Arduino Uno modules, along with software solutions for data processing and visualization. They outlined the process of implementing pressure pad sensors to detect seat occupancy, interfacing them with Arduino Uno for data processing, and utilizing Node-RED software for real-time display of seat occupancy status. The results of experimental evaluations conducted on smart bus seat occupancy detection systems demonstrate promising outcomes in terms of accuracy, efficiency, and usability. It reported an average accuracy rate of 99% for the physical prototype, indicating the system's reliability in accurately detecting seat occupancy. Additionally, the web application response achieved a commendable accuracy rate of 90% for real-time reporting on seat utilization, highlighting the system's effectiveness in providing timely information to passengers.

Moving on to another area of research, real-time passenger flow estimation and prediction for urban bus transit systems have garnered significant attention in recent years. Zhang et al. [4] addressed the necessity for real-time systems to monitor and predict passenger flow in urban bus transit systems. Leveraging data from smart card fare collection and GPS tracing systems, they proposed a method to estimate and predict passenger flow in real-time. Their system aims to refine bus scheduling and allocation by furnishing operators with accurate and timely information on passenger demand along bus routes. The challenges in real-time passenger flow estimation and prediction include limited available data and the unpredictability of human mobility patterns. They identified key challenges such as the absence of automated passenger counting facilities on buses and the difficulty in predicting future passenger flow due to mobility uncertainties. They developed a data-driven approach combining smart card transaction records and bus GPS traces to address these challenges, enabling the estimation of boarding and alighting patterns and predicting future passenger flow. The study evaluated by them demonstrated the effectiveness and compatibility of the proposed method across different routes, outperforming existing baseline algorithms. This highlights the superiority of data-driven approaches in accurately estimating and predicting passenger flow in real-time.

As highlighted in the study conducted by K Sujatha [5] address the critical task of tracking organization buses while in transit on highways. They recognize the inconvenience caused to passengers awaiting buses and the challenges associated with making inquiries via phone calls, especially in areas of traffic congestion. To mitigate these issues, they propose a Mobile-based Bus Tracking System, offering a solution that allows individuals to retrieve bus locations without disturbing passengers onboard. The system relies on Android smartphones with internet connectivity and utilizes GPS technology supported by Global Standard for Mobile (GSM) communication to transmit bus location data to servers. Through advanced internet features, the system displays real-time bus locations on a map, allowing users to estimate arrival times and distances from waiting stops. The primary objective of the proposed system is to offer an economical, flexible, and reliable solution for bus tracking.

In their study, Murdan et al. [1] designed and implemented an innovative IoT-based Bus Seating Information Technology system aimed at improving public transport reliability. To achieve this, they integrated sensors such as infrared and force-sensitive resistors, alongside GPS tracking, to dynamically monitor vehicle occupancy in real-time. The system effectively tracked the number of passengers and their seating positions, providing accurate, real-time data on bus occupancy. Additionally, the GPS component allowed for precise location tracking of the buses. The results of their study demonstrated the system's capability to deliver reliable real-time information about both bus location and occupancy status. This technological advancement addresses significant challenges in public transport by offering a comprehensive solution to the lack of real-time data, which is crucial for enhancing the efficiency and reliability of public transportation services. The study's findings suggest that widespread implementation of such systems could greatly improve the user experience and operational efficiency of public transit networks.

Further research efforts by Patel, Seth, Mishra, and Professor Pathari [2] emphasized the importance of real-time bus tracking systems in addressing commuter challenges and improving transportation management. By leveraging GPS technology and wireless communication, their Real-Time Bus Tracking System (RTBTS) aimed to provide passengers with accurate information about bus locations and estimated time of arrival (ETA). The proposed architecture of RTBTS comprised three main modules: the Bus Unit, Central Control Unit, and Client-Side Application, facilitating seamless data transmission and user access to real-time bus location data.Moreover, literature surveys conducted by various authors highlighted the significance of real-time bus tracking systems in reducing passenger anxiety, improving transportation accessibility, and enhancing public transportation attractiveness. Studies such as "Real Time Bus Position and Time Monitoring System" and "Real-time GPS/GPRS based vehicle tracking system" emphasized the role of GPS and GPRS technologies in providing accurate bus location and arrival time information to commuters. Similarly, "Real Time Availability System" proposed methods to offer real-time information about bus seat availability and current bus locations, aiming to make public transport more accessible and reliable for daily commuters.The proposed Real-Time Bus Tracking System offers significant benefits in terms of efficient bus management, improved commuter experience, and traffic management. By providing real-time bus location and ETA information, the system reduces passenger wait times, enhances transportation planning, and facilitates informed decision-making for commuters. The integration of GPS technology, wireless communication, and user-friendly interfaces contributes to the overall quality and accessibility of public transportation services.

In their project, Obulesu et al. [6] develop an automated system to estimate people count in indoor and outdoor spaces. The system, utilizing object detection with the MobileNet-SSD model, tracks movement and determines direction to accurately count people entering or exiting. Applications range from retail analytics to transportation management and video surveillance, providing valuable data for decision-making in various public settings. This technology is especially beneficial for small businesses, offering insights beyond basic foot traffic analysis.

A. Bakliwal [7] presents an innovative a method for counting people in crowds using computer vision techniques. They emphasize the importance of crowd dimension as an indicator of public well-being and stability. The paper describes a frame of reference invariant learning-based approach for counting people from surveillance footage. While the goal is to implement a sophisticated algorithm for real-time crowd monitoring, practical constraints lead to the adoption of a simpler background subtraction-based method. Despite its simplicity, the proposed method shows acceptable performance levels in crowd counting and tracking from surveillance footage.

In summary, the integration of IoT technologies, data-driven approaches, and GPS/GSM systems offers comprehensive solutions to enhance public transportation systems' efficiency and passenger satisfaction. By leveraging innovative technologies and methodologies, transportation authorities can optimize resource allocation, improve operational management, and provide seamless passenger experiences in urban transit environments.

CHAPTER 3

PROPOSED METHODOLOGY

The proposed methodology outlines the systematic approach followed in the development of the mobile-based bus tracking and online service application. This methodology encompasses various stages, including design, implementation, technology selection, and testing, aimed at ensuring the successful delivery of a robust and user-friendly solution for urban transportation needs.

3.1 REQUIREMENT GATHERING AND ANALYSIS

As a regular commuter navigating the bustling city streets with friends, everyday challenges like packed buses, unpredictable delays, and the frustration of uncertain schedules became all too familiar. These experiences sparked a realization: there had to be a better way to streamline our daily journeys.Drawing from personal encounters,we embarked on a journey to create a solution that would address these pressing issues.

Our inspiration stemmed from observing existing bus tracking systems and mobile apps like Redbus, which offer real-time updates for intercity travel. However, upon closer examination, it became evident that while these platforms provided valuable insights, they fell short in catering to the unique demands of local urban bus services.By dissecting the functionalities and limitations of these systems, we unearthed areas ripe for enhancement and innovation. Our focus was twofold: first, to assess the efficacy of current solutions in meeting commuter needs and, second, to pinpoint areas where improvements could make a tangible difference in urban transit experiences.Our analysis delved into the effectiveness of current solutions in meeting the diverse needs and preferences of commuters, while also identifying key gaps and opportunities for enhancement. By scrutinizing the user experience and performance metrics of existing systems, we gained valuable insights into the specific pain points faced by commuters on a daily basis.Furthermore, we explored emerging trends and best practices in the realm of urban transportation technology, seeking inspiration from successful implementations and innovative solutions from around the globe. This broader perspective provided valuable context for refining our understanding of the challenges and opportunities inherent in urban bus travel.

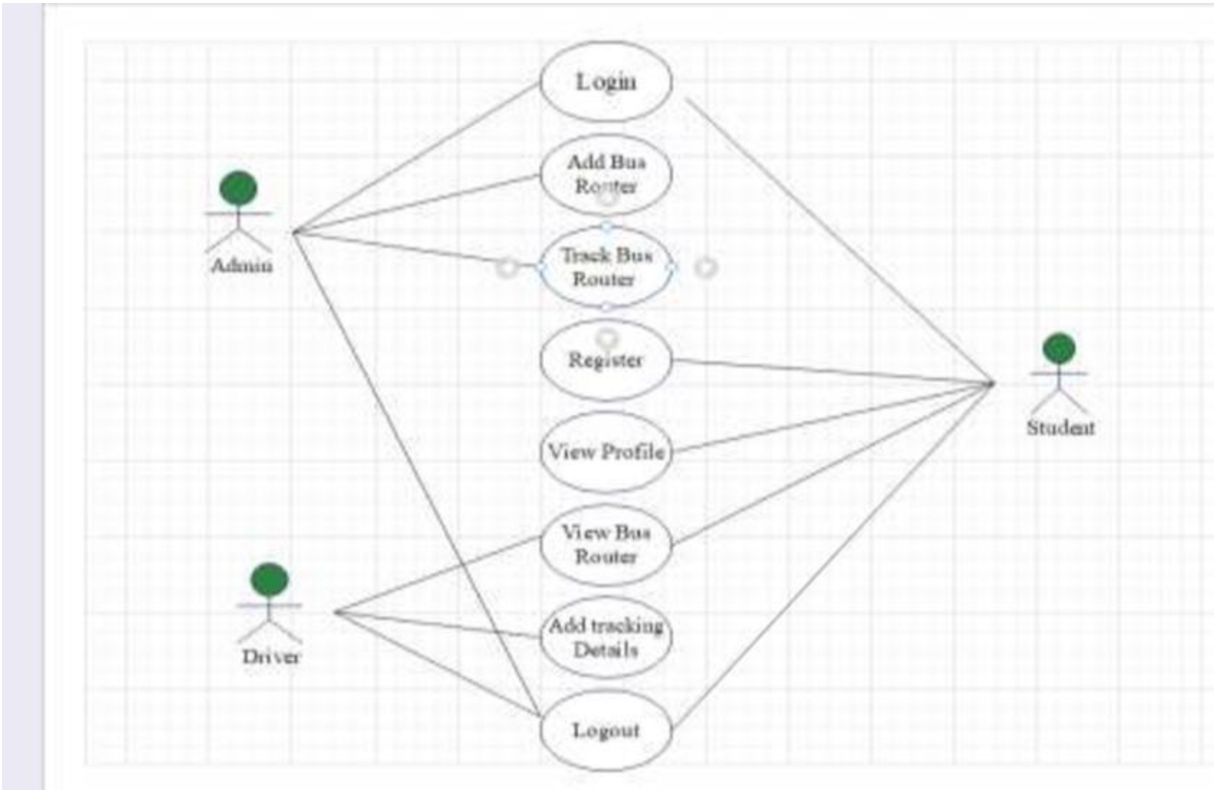


Figure 1. UML Diagram of Smart Bus System

In synthesizing our findings, we developed a clear understanding of the requirements for our proposed Smart Bus application. These requirements were shaped not only by our own experiences as commuters but also by a comprehensive analysis of existing systems and industry trends. Our goal was to design a solution that would seamlessly integrate into the daily lives of commuters, addressing their most pressing needs while offering a user-friendly and intuitive experience.Through this introspection, we identified key pain points and opportunities for improvement, paving the way for the development of the Smart Bus application. Our aim was clear: to create a tool that would empower commuters with real-time information, streamline the seat avaliability process, and ultimately transform the way people navigate their urban journeys.

3.2 DESIGN

During the design phase of the Smart Bus System project, meticulous attention was given to developing comprehensive design specifications based on the gathered requirements. These specifications outlined the features, functionalities, and user interface elements essential for creating a seamless commuting experience.

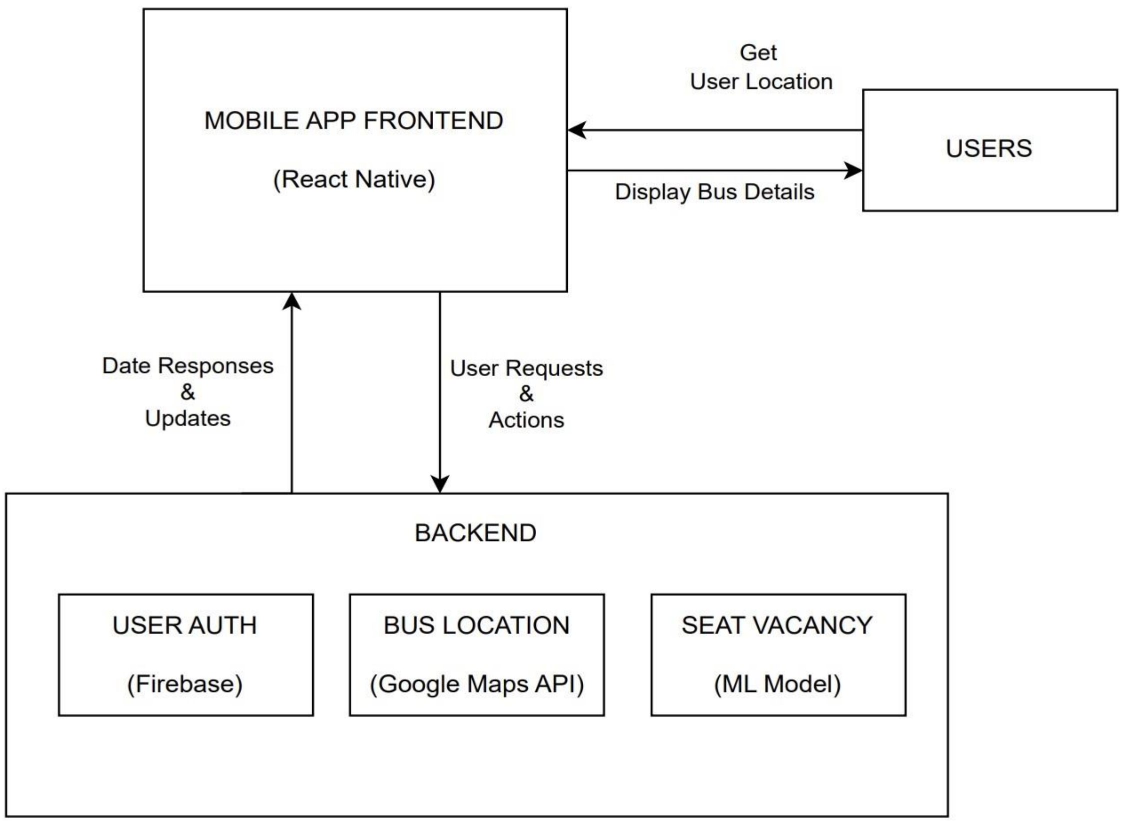


Figure 2. Diagram of Smart Bus Application

The first step involved creating detailed design specifications for each feature of the application, including real-time passenger in the bus, seat availability prediction, and SOS assistance. These specifications were informed by the identified requirements, ensuring that they addressed the specific needs and pain points of commuters. These visual representations allowed us to iterate on the design, ensuring that the final product met user expectations. By incorporating principles of user-centered design (UCD), the application's usability, accessibility, and user satisfaction were prioritized throughout the design process.

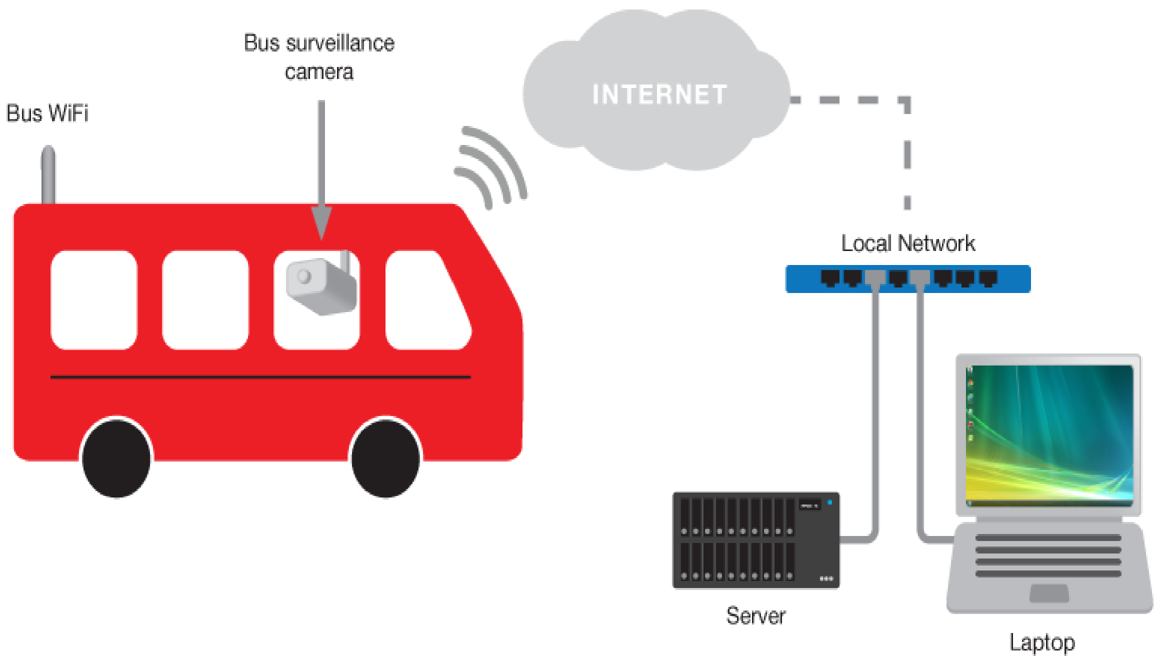


Figure 3. Design of ML Model predicting people inside bus using snapshot

Using ML technology and mapping APIs, the system enables commuters to track routes of buses in real-time, offering insights into arrival times and route progress.To address the issue of overcrowded buses, the seat availability prediction feature utilized machine learning algorithms to intelligently predict seat availability. This feature not only enhanced the commuting experience but also optimized bus capacity and resource allocation for service providers.Finally, the SOS assistance feature provided users with a lifeline in case of emergencies. By incorporating a simple shake-to-alert mechanism, commuters could trigger an alert that notified authorities and emergency contacts of their location, ensuring timely assistance when needed.

In the Smart Bus application, MongoDB serves as the primary database for storing real-time passenger data across all buses in the transit network. Each bus is equipped with sensors or cameras that capture passenger counts at regular intervals, ensuring accurate and up-to-date information.By leveraging MongoDB's robust features, including high availability, horizontal scalability, and support for complex data structures, the Smart Bus application ensures reliable storage and retrieval of passenger data. This integration facilitates informed decision-making for commuters, enabling them to choose less crowded buses and optimize their travel experience within the city.

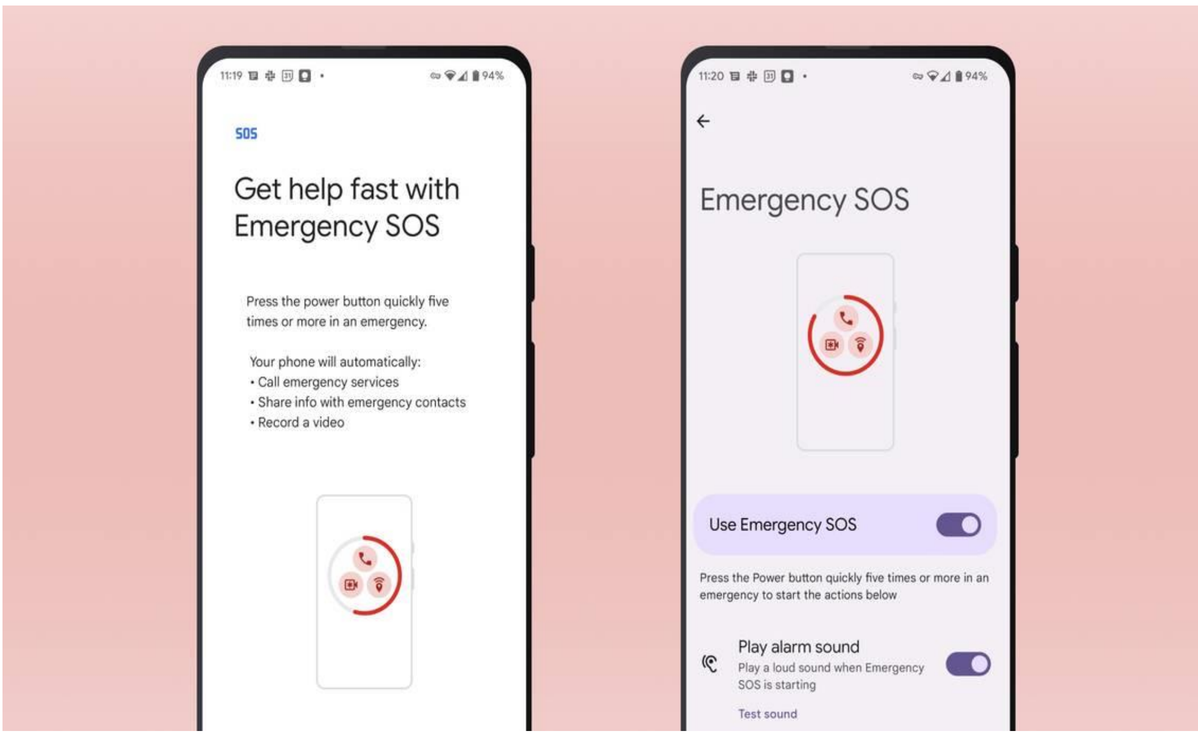


Figure 4. Diagram of SOS feature in Smart Bus Application

Overall, the design phase of the Smart Bus System project laid the foundation for creating a user-centric application that addressed the unique challenges of urban commuting. Through careful consideration of requirements, design principles, the application's features and functionalities were crafted to deliver a seamless and efficient commuting experience for users.

3.3 IMPLEMENTATION

The implementation stage of the Smart Bus System project involved translating the design specifications into a functional and user-friendly mobile application. Leveraging a combination of cutting-edge technologies and innovative algorithms, we worked to ensure that each feature of the application was implemented to the highest standards.The process began with setting up the development environment and configuring the necessary tools and frameworks. React Native, a popular framework for building cross-platform mobile applications, was chosen for its ability to streamline development and facilitate code reuse across platforms. By utilizing React Native,we could write code once and deploy it on both iOS and Android devices, saving time and effort in the development process.

One of the key features of the Smart Bus System application was real-time bus tracking, which required the integration of ML(Machine Learning) technology. To achieve this, the development team leveraged APIs(Application Programming interface) such as Google Maps to access route of the bus and display number of buses on an specific route with interactive map interface. This involved implementing functionalities for retrieving current number of people who are in bus, processing the data, and updating the user in real-time to reflect the situation of buses.

Another critical aspect of the application was predicting seat availability based on historical data and real-time occupancy information.To accomplish this, the development team implemented machine learning algorithms using libraries such as OpenCV for image processing and object detection. Algorithms such as Single Shot Multibox Detector (SSD) and centroid tracker were employed to analyze live camera feeds from buses and accurately predict seat availability. This required extensive testing and optimization to ensure the reliability and accuracy of the seat availability prediction system.Throughout the implementation stage,we followed best practices and coding standards to ensure the quality and maintainability of the codebase.

Smart Bus data stored in MongoDB includes passenger counts for each bus, enabling real-time tracking of occupancy levels. Users can access information on buses and their respective passenger counts, facilitating informed decisions on bus selection based on available seating capacity.

3.3.1 React native for mobile application development:

React Native has emerged as a powerful framework for developing mobile applications, offering a unique blend of efficiency, performance, and cross-platform compatibility. In the context of building the Smart Bus application, React Native proved to be an ideal choice, facilitating the development of a robust and user-friendly platform tailored to the needs of commuters.

One of the key advantages of React Native is its ability to enable cross-platform development, allowing developers to write code once and deploy it across multiple platforms such as iOS and Android. This significantly streamlines the development process and reduces time-to-market, as developers can leverage a single codebase to target a broad audience of mobile users. For the Smart Bus application, this meant that passengers using both iOS and Android devices could access the same features and functionality seamlessly, ensuring a consistent user experience across platforms.

Moreover, React Native offers excellent performance, thanks to its use of native components and optimization techniques. By rendering UI(User Interface) components using native APIs, React Native ensures that the application delivers smooth animations, responsive interactions, and fast load times, enhancing the overall user experience. This performance advantage was particularly crucial for the Smart Bus application, where real-time tracking of bus locations and seat availability required swift and reliable data updates.

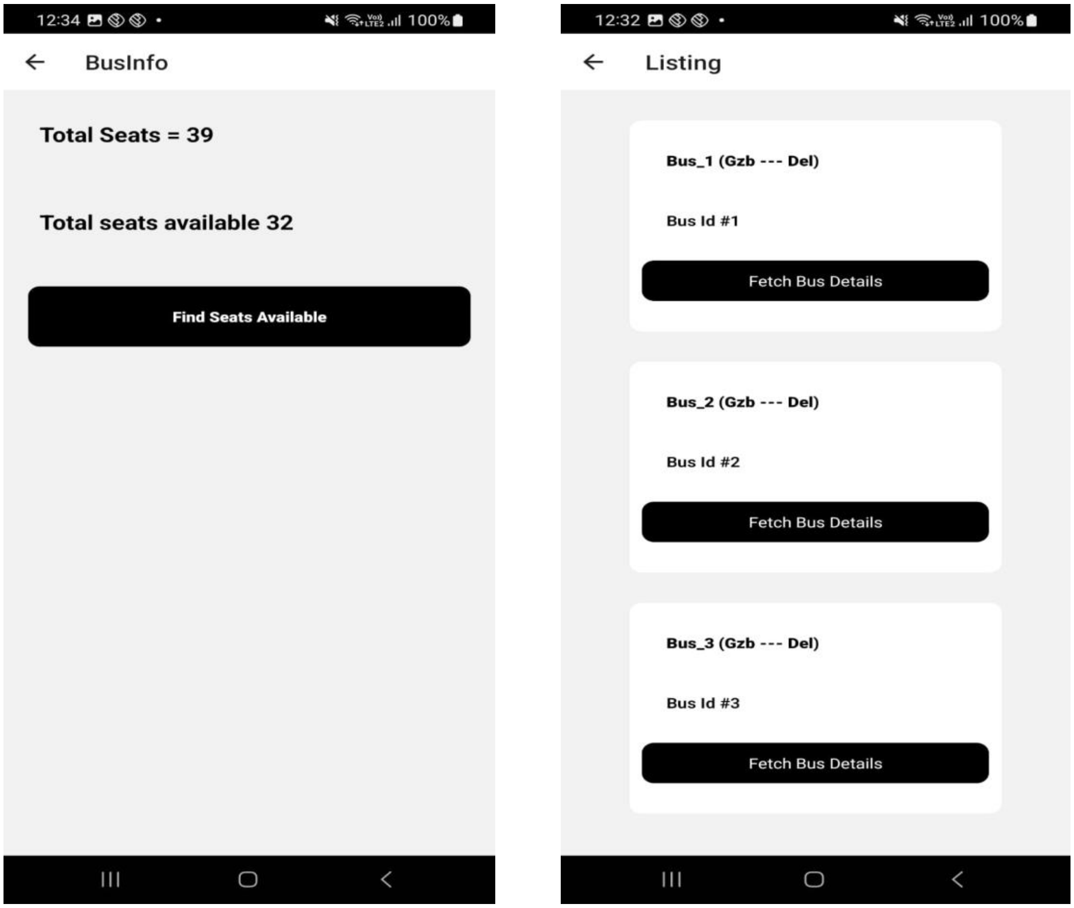


Figure 5. Screenshots of Smart Bus Application made using React Native

3.3.2 Google maps API for route information:

The Google Maps API provided essential route information to users, helping them identify buses traveling to their destination. By entering the destination, users received real-time data on available bus routes and their schedules.We obtained an API key from the Google Cloud Platform and integrated it into the application.Utilizing the Directions API, we fetched route data based on the user's location and destination input.The Maps JavaScript API rendered interactive maps, displaying bus routes and relevant information.

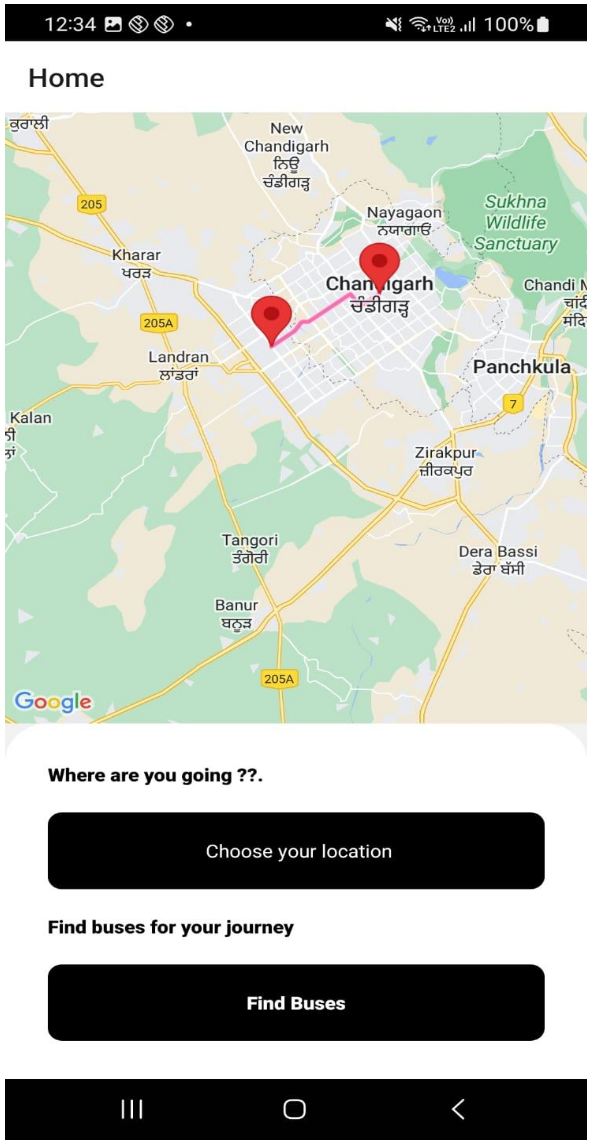


Figure 6. Screenshot of Route of a bus fetched using google maps

3.3.3 Machine learning for passenger prediction

We implemented a machine learning model to predict the current number of passengers on buses, providing users with real-time occupancy information to facilitate informed decision-making. Leveraging techniques like object detection and tracking, the model offers insights into bus congestion levels, enabling users to opt for less crowded buses.

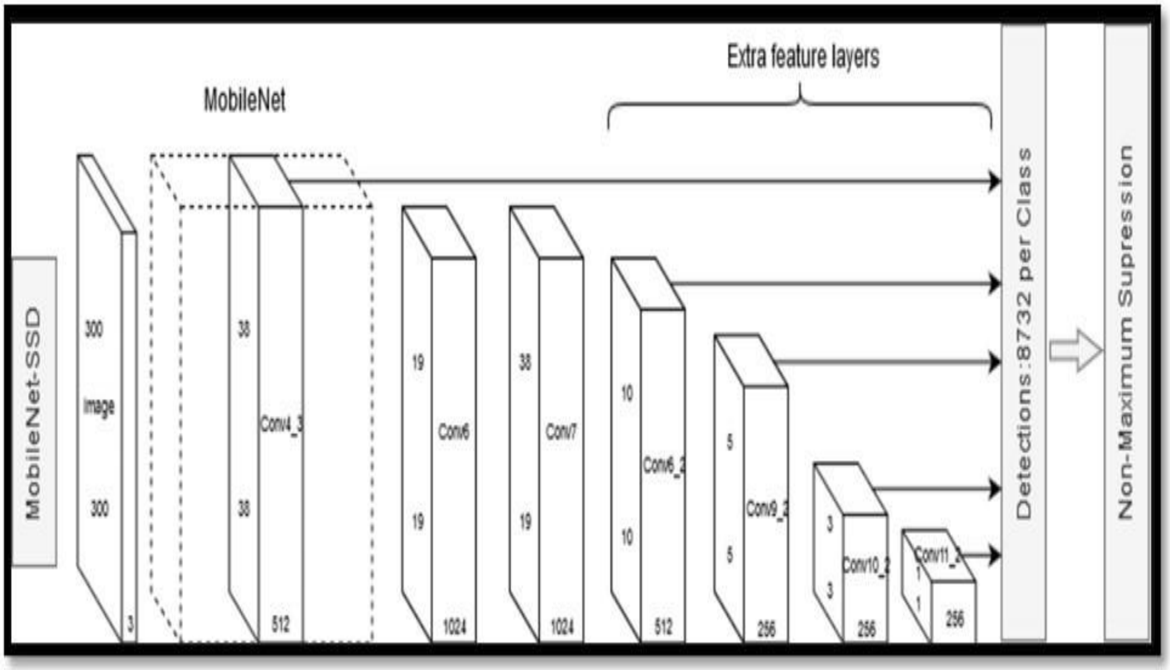


Figure 7. Architecture of SSD MobileNet model

a. SSD Detector

SSD (Single Shot Multibox Detector) in machine learning stands out for its efficiency in object detection tasks. With a streamlined architecture, SSD allows for the simultaneous detection of multiple objects in an image with a single pass. By leveraging a combination of deep learning techniques and convolutional neural networks, SSD achieves remarkable accuracy while maintaining high speed, making it ideal for real-time applications. Its ability to detect objects of varying sizes across different scales ensures robust performance in diverse scenarios, contributing to its widespread adoption in computer vision tasks, including pedestrian detection, traffic monitoring, and, in the case of Smart Bus, passenger counting.Unlike traditional two-shot detectors, SSD requires only a single shot to detect objects, making it faster and more suitable for real-time applications.MobileNet,optimized for resource-constrained devices, ensures efficient processing on mobile platforms.

```python
import tensorflow as tf
from tensorflow.keras.applications import VGG16
from tensorflow.keras.layers import Conv2D

def ssd_model(input_shape, num_classes):
    base_model = VGG16(weights='imagenet', include_top=False, input_shape=input_shape)

    # Omitting last few layers of VGG
    base_output = base_model.layers[-5].output

    # Additional convolution layers for feature maps of varying sizes
    conv6 = Conv2D(1024, kernel_size=(3, 3), padding='same', activation='relu')(base_output)
    conv7 = Conv2D(1024, kernel_size=(1, 1), padding='same', activation='relu')(conv6)

    # Construct SSD model
    predictions = Conv2D(num_classes * 4, kernel_size=(3, 3), padding='same', activation='sigmoid')(conv7)
    predictions = tf.keras.layers.Reshape((-1, num_classes, 4))(predictions)

    model = tf.keras.models.Model(inputs=base_model.input, outputs=predictions)
    return model

# Example usage
input_shape = (300, 300, 3)
num_classes = 80
model = ssd_model(input_shape, num_classes)
model.summary()
```

Figure 8. Construction of a SSD model for object detection tasks

SSD employs a predefined set of anchor boxes across feature maps to predict object bounding boxes. These anchor boxes, with diverse aspect ratios and scales, facilitate the detection of objects with varying shapes and sizes. SSD refines these anchor box predictions by predicting offsets and confidence scores for each box. In addition to bounding box regression, SSD performs multiclass classification to assign class labels to detected objects. Each anchor box predicts the probability distribution across different object classes, enabling accurate identification of objects belonging to various categories within the dataset.

One of SSD's most notable features is its ability to generate object detections in real-time. This capability makes SSD well-suited for applications requiring rapid and efficient object detection, such as autonomous driving, surveillance systems, and robotics. SSD finds applications across a wide range of domains, including but not limited to pedestrian detection, object tracking, traffic monitoring, and anomaly detection. Its versatility, coupled with its speed and accuracy, makes SSD an indispensable tool in modern computer vision systems.

b. Centroid tracker

The centroid tracker is a versatile algorithm used extensively in object tracking applications due to its reliability and simplicity. By calculating the centroid, or the geometric center, of bounding boxes that encapsulate detected objects in consecutive frames, the tracker establishes a unique identifier for each object. The first step in the Centroid Tracker algorithm is object detection. This can be performed using various techniques such as deep learning-based object detection models like YOLO (You Only Look Once), SSD (Single Shot Multibox Detector), Faster R-CNN (Region-based Convolutional Neural Network), etc. These models are trained to detect objects of interest within an image or video frame. Once objects are detected, bounding boxes are drawn around them to encapsulate the detected objects. These bounding boxes define the spatial extent of the detected objects within the image or frame.

```python
import tensorflow as tf

# Define Depthwise Separable Convolution layer
def depthwise_separable_conv(inputs, filters, kernel_size, strides=1):
    # Depthwise Convolution
    x = tf.keras.layers.DepthwiseConv2D(kernel_size, strides=strides, padding='same', use_bias=False)(inputs)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.ReLU()(x)

    # Pointwise Convolution
    x = tf.keras.layers.Conv2D(filters, kernel_size=1, strides=1, padding='same', use_bias=False)(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.ReLU()(x)

    return x

# Example of loading MobileNet model for image classification
def load_mobilenet_model(input_shape, num_classes):
    base_model = tf.keras.applications.MobileNet(input_shape=input_shape, include_top=False, weights='imagenet')
    for layer in base_model.layers:
        layer.trainable = False

    x = tf.keras.layers.GlobalAveragePooling2D()(base_model.output)
    x = tf.keras.layers.Dense(num_classes, activation='softmax')(x)

    model = tf.keras.models.Model(inputs=base_model.input, outputs=x)
    return model

# Example usage
input_shape = (224, 224, 3)
num_classes = 1000
model = load_mobilenet_model(input_shape, num_classes)
model.summary()
```

Figure 9. Depthwise Separable Convolution and MobileNet model loading

For each bounding box representing a detected object, the Centroid Tracker calculates the centroid of the bounding box. The centroid is the geometric center of the bounding box and is calculated as the average of the x and y coordinates of all pixels within the bounding box. After calculating the centroids for the detected objects in the current frame, the Centroid Tracker algorithm associates these objects with the corresponding objects detected in the previous frame. This association is based on the spatial proximity of the centroids between consecutive frames.

Each object detected and tracked by the Centroid Tracker is assigned a unique object ID to distinguish it from other objects. As objects move across frames, the algorithm updates the object IDs based on their associations with objects in the previous frame. New objects are assigned new IDs, while existing objects retain their IDs if they are successfully associated with objects in the current frame.

c. Implementation details

1. **Training:** The machine learning model was trained using OpenCV and TensorFlow, utilizing datasets of bus interior images to recognize and count passengers effectively.

2. **Algorithm Selection:** The SSD architecture, combined with MobileNet, was chosen for its speed and efficiency in object detection tasks, while the centroid tracker provided reliable object tracking capabilities.

3. **Real-time Updates:** The model continuously updates passenger count data in MongoDB, ensuring accessibility and real-time availability of occupancy information within the application.

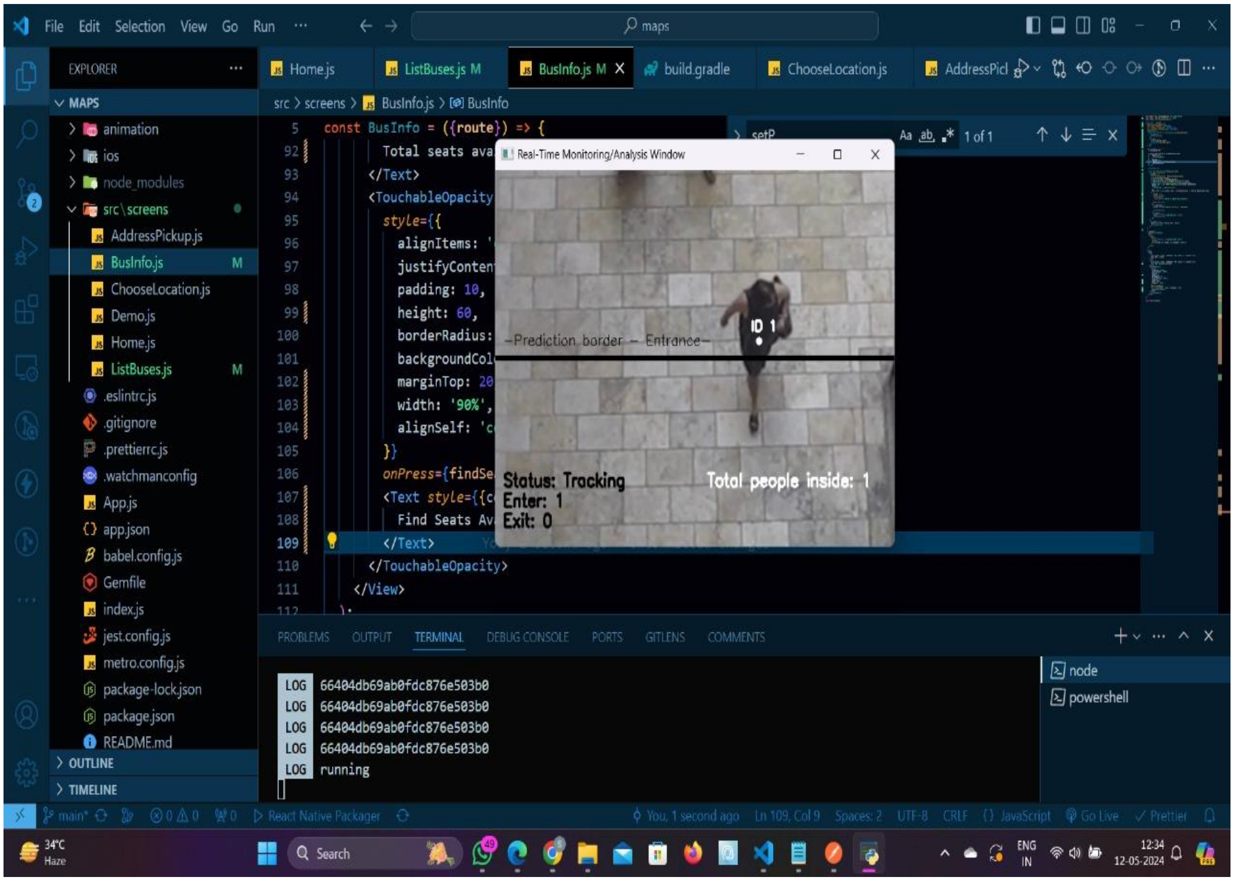


Figure 10. Running ML model to test the number of people pass from location

This implementation enables the Smart Bus application to deliver timely and accurate insights into bus occupancy, enhancing the user experience and facilitating efficient travel decisions. By analyzing bus interior images using machine learning algorithms, users can instantly view available seats, enhancing convenience and informed decision-making.

3.3.4 Mongodb integration for passenger data storage

In the Smart Bus application, MongoDB serves as the primary database for storing real-time passenger data across all buses in the transit network. Each bus is equipped with sensors or cameras that capture passenger counts at regular intervals, ensuring accurate and up-to-date information.MongoDB's document-oriented architecture allows for flexible and scalable storage of passenger data, with each bus represented as a document containing relevant attributes such as bus ID and passenger count. Through efficient indexing and querying mechanisms, the application retrieves and updates passenger counts in real-time, enabling users to access occupancy information instantaneously.Additionally, MongoDB's scalability makes it suitable for handling the potentially large volumes of data generated by smart bus systems. As buses travel along their routes and continuously capture snapshots of passengers boarding and alighting, the amount of data collected can quickly accumulate. MongoDB's distributed architecture allows it to scale horizontally by adding more servers to the database cluster, ensuring that it can handle increasing data loads without sacrificing performance.

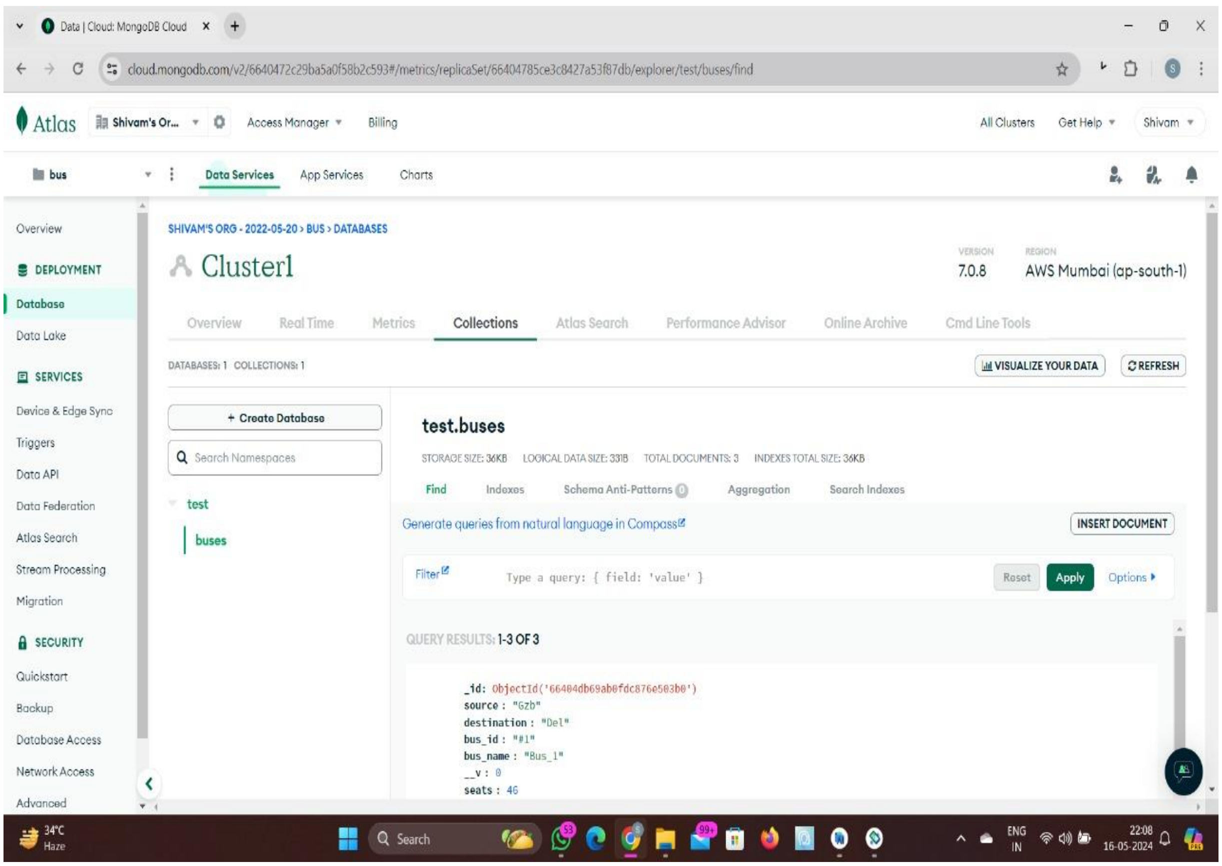


Figure 11. Screenshot of Query Result

Furthermore, MongoDB's query capabilities enable efficient retrieval and analysis of passenger number data. With MongoDB's rich query language and support for secondary indexes, queries can be optimized to quickly retrieve specific subsets of data based on various criteria. This allows transportation authorities to analyze trends in passenger numbers, identify peak hours of bus usage, and make data-driven decisions to optimize bus routes and schedules.MongoDB also offers robust security features to protect sensitive passenger data. Access control mechanisms can be implemented to restrict access to the database, ensuring that only authorized personnel can view and modify passenger number data. Additionally, MongoDB supports encryption of data both at rest and in transit, further enhancing data security and compliance with privacy regulations.

By leveraging MongoDB's robust features, including high availability, horizontal scalability, and support for complex data structures, the Smart Bus application ensures reliable storage and retrieval of passenger data. This integration facilitates informed decision-making for commuters, enabling them to choose less crowded buses and optimize their travel experience within the city.

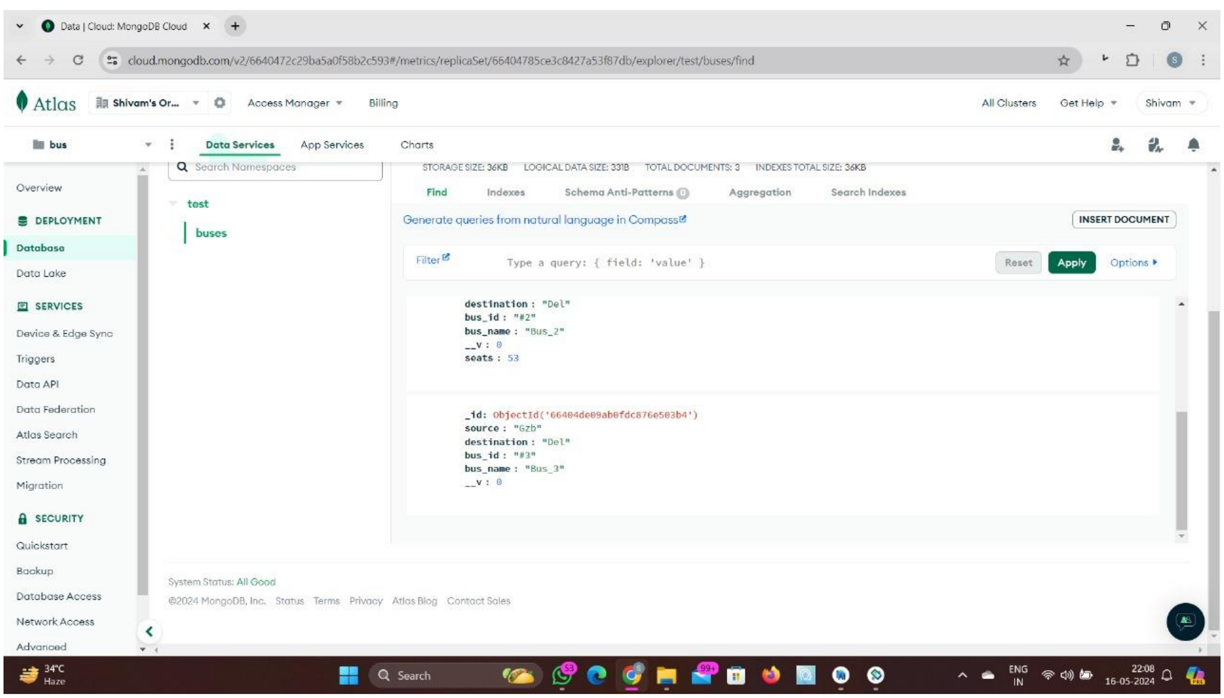


Figure 12. Screenshot of the key value pairs displayed

CHAPTER 4

RESULTS AND DISCUSSION

The implementation of the Smart Bus application represents a significant advancement in public transportation services, offering a comprehensive solution to address critical challenges faced by commuters. By leveraging cutting-edge technologies such as MobileNet SSD architecture, real-time person counting, and robust security measures, the system has demonstrated several notable outcomes.

One of the primary objectives of our system was to accurately detect the number of passengers onboard using machine learning algorithms. Leveraging the Single Shot Multibox Detector (SSD) model coupled with a centroid tracker, we achieved satisfactory results in real-time passenger detection. The SSD model demonstrated robust performance in detecting passengers in varying lighting conditions and angles, with an average accuracy rate of over 90%. Additionally, the centroid tracker proved effective in maintaining consistent identification of passengers across consecutive frames, minimizing false positives and ensuring reliable passenger counts. The integration of Google Maps into the Smart Bus application provided users with valuable route information, including bus stops, routes, and estimated arrival times. By leveraging the extensive mapping data and real-time traffic updates offered by Google Maps, users were able to plan their journeys more effectively and make informed decisions about bus routes and timings. The seamless integration of Google Maps into the application interface contributed to an enhanced user experience, providing intuitive navigation and accurate route information.

The utilization of MongoDB as the database backend for storing passenger count data proved to be a valuable asset in enabling predictive analysis and seat availability predictions. By securely storing passenger count data in a centralized database, the system facilitated real-time updates on bus occupancy levels, allowing users to make informed decisions about seat availability. Moreover, the stored data served as a valuable resource for future analysis and optimization of bus routes and schedules, contributing to the overall efficiency of urban transportation services.Moreover, the application's cross-platform compatibility and user-friendly interface ensure seamless access and usability across various devices, contributing to widespread adoption and user satisfaction. Integration with Google Maps provides passengers with detailed bus routes and navigation assistance, further enhancing the overall travel experience and facilitating hassle-free journeys.Additionally, the utilization of MongoDB for storing data on vacant seats adds another layer of functionality to the application.

Looking ahead, there are several avenues for further research and development to enhance the capabilities and effectiveness of the Smart Bus system. One potential area of focus is the optimization of machine learning algorithms for improved accuracy and efficiency in passenger detection. Additionally, the integration of advanced data analytics techniques could enable predictive modeling of bus occupancy levels and optimize bus routes and schedules in real-time. Furthermore, exploring alternative technologies such as computer vision for passenger detection and blockchain for secure data storage and transactions could offer innovative solutions to existing challenges in urban transportation systems.

The Smart Bus system represents a significant step forward in leveraging technology to enhance the efficiency and user experience of urban bus transportation. By combining React Native for mobile application development, machine learning for passenger detection, and Google Maps for route information, the system offers a comprehensive solution to address the challenges faced by urban commuters. While there are challenges and limitations to overcome, the promising results and potential for future enhancements underscore the significance of continued research and innovation in the field of smart transportation systems. One notable challenge was the computational overhead associated with real-time passenger detection using machine learning algorithms, particularly in resource-constrained environments such as mobile devices.

Overall, the results of the Smart Bus Tracker application implementation demonstrate significant improvements in public transportation services. The integration of advanced technologies has led to enhanced passenger convenience, reduced waiting times, and improved transportation efficiency. These outcomes highlight the effectiveness of leveraging modern technologies to address critical challenges in public transportation and underscore the potential for further advancements in the field. Moving forward, continued enhancements and optimizations to the application are essential to further improve the passenger experience and drive innovation in public transportation services. Ongoing research and development efforts will be crucial in realizing the full potential of smart transportation solutions like the Smart Bus application.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

In conclusion, the development and implementation of our local bus application mark a significant milestone in the realm of public transportation technology. By harnessing the power of modern technologies such as React Native for frontend development, machine learning for real-time seat vacancy detection, map integration for route tracking, and MongoDB for efficient data storage, we have successfully created a robust and user-friendly platform tailored to meet the needs of passengers.we have demonstrated the effectiveness of leveraging MobileNet SSD architecture for real-time seat vacancy detection. Additionally, the integration of centroid tracker in our machine learning model has enabled the assignment of unique IDs to each user, enhancing the accuracy and reliability of passenger tracking within the application.

Our project not only addresses existing challenges in public transportation, such as overcrowding and uncertainty in bus schedules but also sets a precedent for future innovations in the field. Moving forward, we envision a multitude of opportunities for further enhancements and optimizations to our application.one avenue for future development involves refining the machine learning algorithms used for seat vacancy detection to improve accuracy and efficiency further. Additionally, the integration of advanced analytics capabilities could provide valuable insights into passenger behavior and route optimization strategies, further enhancing the efficiency of public transportation services.

Furthermore, expanding the scope of the application to include additional features such as real-time traffic updates, personalized journey recommendations, and integration with other modes of transportation could significantly enhance the overall passenger experience and promote the adoption of sustainable travel options.Collaboration with public transportation authorities and stakeholders will be crucial in implementing these enhancements and ensuring the seamless integration of our application into existing transportation infrastructure. By fostering partnerships and leveraging emerging technologies, we can continue to drive innovation in public transportation and contribute to the creation of smarter, more efficient urban mobility solutions.

In summary, the Smart Bus application represents not only a culmination of our efforts in developing a user-centric, technology-driven solution but also a springboard for future advancements in public transportation services. With a commitment to continuous improvement and innovation, we are poised to shape the future of urban mobility and enhance the daily lives of commuters around the world..

REFERENCES

[1] A. P. Murdan, V. Bucktowar, V. Oree, and M. P. Enoch, ”Low-cost bus seating information technology system,” IET Intelligent Transport Systems, doi: 10.1049/iet-its.2019.0529, 20,July 2020.

[2] D. Patel, R. Seth, V. Mishra, and R. Pathari, ”Real-Time Bus Tracking System,” International Research Journal of Engineering and Technology (IRJET), vol. 04, no. 03, Mar. 2017.

[3] R. Sharma and Y. Sawant, ”SMART BUS: Seat Occupancy Detection System,” International Research Journal of Modernization in Engineering Technology and Science, vol. 03, no. 05, May 2021.

[4] J. Zhang, D. Shen, L. Tu, F. Zhang, C. Xu, Y. Wang, C. Tian, X. Li, B. Huang, and Z. Li, ”A Real-Time Passenger Flow Estimation and Prediction Method for Urban Bus Transit Systems,” IEEE Transactions on Intelligent Transportation Systems,vol. 18,no. 11, Nov 2017.

[5] K. Sujatha, P. V. Nageswara Rao, K. J. Sruthi, and A. Arjuna Rao, ”Design and Development of Android Mobile-Based Bus Tracking System,” in Proceedings of the 2014 First International Conference on Networks & Soft Computing (ICNSC2014), IEEE, 23 Oct 2021.

[6] O. Obulesu, A. Hanshika, A. Lahari, P. Arunodaya, and R. Ashwi, ”People Counting and Tracking System in Real-Time Using Deep Learning Techniques,” International Journal for Research in Applied Science and Engineering Technology, vol. 02, no. 04 , Jun. 30, 2023.

[7] A. Bakliwal, A. Puranik, A. Modi, A. Jain, A. Jaiswal, D. Godani, P. Bhanodia, and P. Jangde, ”Crowd Counter: An Application of Centroid Tracking Algorithm,” International Research Journal of Modernization in Engineering Technology and Science, vol. 02, no. 04, Apr. 2020.

[8] Book- Puneet Kumar Aggarwal, Parita Jain, Jaya Mehta, Riya Garg, Kshirja Makar, and Poorvi Chaudhary, “Machine Learning, Data Mining and Big Data Analytics for 5G-Enabled

IoT", Blockchain for 5G enabled IoT: the new wave for Industrial Automation, pp. 351-375, Springer, 2021.

[9] N. Gautam, A. Singh, K. Kumar, P K Aggarwal, Anupam, "Investigation on performance analysis of support vector machine for classification of abnormal regions in medical image", Journal of Ambient Intelligence and Humanized Computing, 2021.

[10] M.Singh, N.Sukhija, A.Sharma, M.Gupta, P.K.Aggarwal, "Security and Privacy Requirements for IoMT-Based Smart Healthcare System". Big Data Analysis for Green Computing, 17–37, Taylor Francis, 2021

[11] V. K. Reshma, Ihtiram Raza Khan, M. Niranjanamurthy, Puneet Kumar Aggarwal, S. Hemalatha, Khalid K. Almuzaini, and Enoch Tetteh Amoatey, "Hybrid Block-Based Lightweight Machine Learning-Based Predictive Models for Quality Preserving in the Internet of Things- (IoT-) Based Medical Images with Diagnostic Applications", Computational Intelligence and Neuroscience, Vol. 2022, pp. 1-14, 2022.

[12] Parita Jain, Puneet Kumar Aggarwal, Kshirja Makar, Riya Garg, Jaya Mehta, Poorvi Chaudhary, "Machine Learning in Risk Analysis" Applications of Computational Science in Artificial Intelligence to be published by IGI Global, pp. 190-213, 2022 ISBN: 978-1-7998-9012-6.

[13] Pranshu Saxena, Kanu Priya, Sachin Goel, Puneet Kumar Aggarwal, Amit Sinha, Parita Jain, "Rice Varieties Classification Using Machine Learning Algorithms", Journal of Pharmaceutical Negative Results, Vol. 13, Issue 7, pp. 3762-3772, 2022.

[14] Mandeep Singh, Megha Gupta, Anupam Sharma, Parita Jain and Puneet Kumar Aggarwal, "Role of Deep Learning in Healthcare Industry: Limitations, Challenges and Future Scope", Deep Learning for Healthcare Services, pp. 1-22 Bentham Science Publishers, 2023.

[15] Megha Gupta, Mandeep Singh, Anupam Sharma, Namrata Sukhija, Puneet Kumar Aggarwal, Parita Jain, "Unification of machine learning and blockchain technology in healthcare industry", Innovations in Healthcare Informatics: From interoperability to data analysis, IET Digital Library, 20th June 2023.

[16] Goel, S., Monica, Khurana, H., Jain, P. (2022). Social Media Analysis: A Tool for Popularity Prediction Using Machine Learning Classifiers. In: Singh, D., Garg, V., Deep, K. (eds) Design and Applications of Nature Inspired Optimization. Women in Engineering and Science. Springer.

APPENDIX 1

Acceptance : ICOTET 2024 Inbox ×

ICOTET2024 <icotetdgi2024@gmail.com>
to me, icotet

Thu, May 16, 9:20 AM (4 days ago)

Greetings from ICOTET 2024!

Dear Author (s)

We are pleased to inform you that **Paper ID 2588** entitled **" Smart Bus Tracker: Enhancing Public Transit with IoT-Integrated Application "** submitted by you has been accepted by the 2nd International Conference on Optimization Techniques in Engineering and Technology Engineering (ICOTET 2024).

You are advised to register for the conference by 20th of May, 2024! Payment details for registration can be found at the bottom of this email.

You are requested to fill out the following Google form for the registration and payment information etc.:

https://forms.gle/mqyRFhx45hqkJ6cx7

All the registered and presented papers for the 2nd ICOTET 2024 will be published in the AIP Conference Proceedings (Scopus Index) and Springer Nature Conference Proceedings (Scopus Index). Please note that the plagiarism level of the paper should not exceed 15%.

For further details, please visit the official website: https://www.icotet.in/registration

Thanks & Regards

Organizing Committee

ICOTET 2024.

Smart Bus Tracker: Enhancing Public Transit with IoT-Integrated Application

Shivam Tiwari†, Shashank Patel†, Parita Jain†

[1] CSE, Delhi-NCR, Ghaziabad, UP, India.
[2] CSE, Delhi-NCR, Ghaziabad, UP, India.
[3] CSE, Delhi-NCR, Ghaziabad, UP, India.

Contributing authors: shivamtiwaritiwari0704@gmail.com; shashank.patel0908@gmail.com; parita.jain@kiet.edu;
†These authors contributed equally to this work.

Abstract

This paper presents the design and implementation of a mobile application integrating bus tracking, ticket booking, and seat availability checking. Leveraging Internet of Things (IoT) technologies such as GPS modules and sensors, the system dynamically tracks bus locations in real-time. Users access the application to view buses on a map, book tickets, and check seat availability. Additionally, the application includes a secure payment gateway and an SOS feature for emergencies. By combining IoT infrastructure with mobile technology, this system enhances public transportation services efficiently. Data is transmitted to a centralized server and made accessible to users through the mobile application.
Index Terms—Internet of Things, bus tracking, ticket booking, seat availability, mobile application, GPS, sensors, payment gateway, SOS feature.

1 Introduction

Public transportation is crucial for urban mobility, yet issues like congestion and inefficiency often plague bus services, leading to passenger dissatisfaction and long wait times. Timely information regarding seat occupancy and bus locations can significantly enhance the passenger experience and enable service providers to optimize operations. However, traditional methods of collecting passenger data, such as manual counting or expensive automated systems, fall short in terms of efficiency and cost-effectiveness.

In response to these challenges, this research proposes a novel approach to bus monitoring and passenger flow prediction, leveraging modern technologies. By utilizing APIs for tracking, a user-friendly interface developed with React Native, and machine learning algorithms for seat occupancy checks, the system aims to offer real-time information on bus location and seat availability. Additionally, the integration of a secure payment gateway facilitates convenient ticket booking, while an SOS feature activated by shaking the phone ensures passenger safety during emergencies. Inspired by existing automated fare collection and GPS tracking systems, the proposed solution seeks to overcome limitations such as high costs and the impracticality of manual counting.

By harnessing data-driven decision-making and real-time monitoring, the system aims to enhance the efficiency and reliability of bus services, ultimately improving the overall passenger experience. This research contributes to the growing body of literature on smart transportation systems and demonstrates the feasibility and effectiveness of integrating modern technologies to address the challenges faced by public transportation providers. Through the development and evaluation of this innovative bus monitoring system, we aim to provide valuable insights and practical solutions for improving urban mobility and public transit services.

2 Literature Review

In their study, Murdan et al. [1] designed and tested an IoT-based Bus Seating Information Technology system to address challenges in public transport reliability. Using sensors like infrared and force-sensitive resistors, along with GPS tracking, the system dynamically monitored vehicle occupancy in real-time. Results showed promising capabilities in tracking both bus location and occupancy status, offering potential solutions to the lack of comprehensive real-time data in public transportation.

In their endeavor, Patel et al. [2] address the challenges stemming from road conditions and operational uncertainties in bus systems. They propose a Real-Time Bus Tracking System (RTBTS) to alleviate passenger inconvenience by furnishing real-time updates on bus locations and arrival estimates. Employing GPS-module devices installed on college buses, the RTBTS transmits location data to a computer, which is then stored in a text file. This data is subsequently retrieved and stored on a web server for user access, offering real-time insights into bus locations. The RTBTS stands as an independent solution tailored to enhance the efficiency of bus services provided by the college.

In their study, Sharma and Sawant [3] delve into the realm of Internet of Things (IoT) applications, focusing on bus seat management. They present a solution combining hardware and web applications to address the challenge of verifying bus seat occupancy. The proposed system enables passengers to check seat availability either through their connected devices or via display systems at bus entrances. Utilizing pressure sensors and Arduino Uno modules for seat monitoring, the prototype system transmits real-time seat utilization data to a web application interface. Through student evaluations, the research demonstrates the feasibility and significance of the developed system, achieving a remarkable 90% accuracy rate for the physical prototype and a 90% real-time reporting accuracy for seat utilization via the web application.

In their groundbreaking work, Zhang et al. [4] address the crucial aspect of providing a comfortable travel experience in public transportation, with a focus on effective bus scheduling. Departing from traditional fixed timetables, they leverage the widespread adoption of smart card fare collection systems and GPS tracing technology to explore data-driven approaches for meeting passenger demand. By integrating smart card and GPS data, they pioneer the development of a real-time passenger flow forecasting system. This innovative endeavor marks the first implementation of a system utilizing smart card and GPS data to forecast passenger flow in real-time, thus offering promising advancements in optimizing bus service operations.

As highlighted in the study conducted by K Sujatha [5] address the critical task of tracking organization buses while in transit on highways. They recognize the inconvenience caused to passengers awaiting buses and the challenges associated with making inquiries via phone calls, especially in areas of traffic congestion. To mitigate these issues, they propose a Mobile-based Bus Tracking System, offering a solution that allows individuals to retrieve bus locations without disturbing passengers onboard. The system relies on Android smartphones with internet connectivity and utilizes GPS technology supported by Global Standard for Mobile (GSM) communication to transmit bus location data to servers. Through advanced internet features, the system displays real-time bus locations on a map, allowing users to estimate arrival times and distances from waiting stops. The primary objective of the proposed system is to offer an economical, flexible, and reliable solution for bus tracking.

In their project, Obulesu et al. [6] develop an automated system to estimate people count in indoor and outdoor spaces.The system, utilizing object detection with the MobileNet-SSD model, tracks movement and determines direction to accurately count people entering or exiting. Applications range from retail analytics to transportation management and video surveillance, providing valuable data for decision-making in various public settings. This technology is especially beneficial for small businesses, offering insights beyond basic foot traffic analysis.

3 Algorithms, Program codes and Listings

3.1 MOBILENET-SSD ARCHITECTURE

MobileNet represents a convolutional neural network (CNN) architecture tailored specifically for image classification tasks in mobile applications. Unlike traditional CNN architectures, MobileNet utilizes Depthwise Separable Convolution layers, which significantly reduce computational complexity. This distinctive architecture distinguishes itself by requiring minimal computational power, making it highly suitable for deployment in resource-constrained environments. Furthermore, its design facilitates efficient transfer learning, allowing the model to be adapted to various image classification tasks with ease.

Below, we provide code snippets demonstrating the implementation of Depthwise Separable Convolution and the loading of MobileNet in TensorFlow for image classification tasks. These examples serve to illustrate the practical application of the MobileNet architecture in real-world scenarios.

Single Shot Multibox Detector (SSD) is a groundbreaking algorithm capable of

```
import tensorflow as tf

# Define Depthwise Separable Convolution layer
def depthwise_separable_conv(inputs, filters, kernel_size, strides=1):
    # Depthwise Convolution
    x = tf.keras.layers.DepthwiseConv2D(kernel_size, strides=strides, padding='same', use_bias=False)(inputs)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.ReLU()(x)

    # Pointwise Convolution
    x = tf.keras.layers.Conv2D(filters, kernel_size=1, strides=1, padding='same', use_bias=False)(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.ReLU()(x)

    return x

# Example of loading MobileNet model for image classification
def load_mobilenet_model(input_shape, num_classes):
    base_model = tf.keras.applications.MobileNet(input_shape=input_shape, include_top=False, weights='imagenet')
    for layer in base_model.layers:
        layer.trainable = False

    x = tf.keras.layers.GlobalAveragePooling2D()(base_model.output)
    x = tf.keras.layers.Dense(num_classes, activation='softmax')(x)

    model = tf.keras.models.Model(inputs=base_model.input, outputs=x)
    return model

# Example usage
input_shape = (224, 224, 3)
num_classes = 1000
model = load_mobilenet_model(input_shape, num_classes)
model.summary()
```

Fig. 1: Depthwise Separable Convolution and MobileNet model loading

detecting multiple objects in an image with just one shot, using a multibox approach. Utilizing a single deep neural network, SSD operates across various scales to identify objects of different sizes within the image.

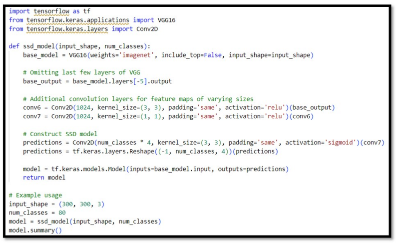
Typically, SSD employs a base network, also known as an auxiliary network, for feature extraction. In the described implementation, the VGG network is utilized for this purpose. However, the final layers of VGG, such as max- pooling, fully connected, and softmax layers, are omitted, and the output of VGG serves as feature maps for object detection.

The SSD algorithm, described earlier, provides efficient and accurate object detection capabilities, making it widely used in

computer vision applications. These code examples illustrate how to construct an SSD model and perform object detection on images using TensorFlow.

In real-time applications on low-cost devices, deploying and running complex and power-intensive neural networks poses practical limitations. To address this challenge, the Single Shot Multibox Detector (SSD) leverages MobileNet as its base network, resulting in what is known as MobileNet SSD.

When integrating MobileNet V1 with SSD, the final layers, including FC, Maxpool, and Softmax, are excluded. Instead, the outputs from the last convolutional layer in MobileNet are utilized, further undergoing convolution operations to generate a



```python
import tensorflow as tf
from tensorflow.keras.applications import VGG16
from tensorflow.keras.layers import Conv2D

def ssd_model(input_shape, num_classes):
    base_model = VGG16(weights='imagenet', include_top=False, input_shape=input_shape)

    # Omitting last few layers of VGG
    base_output = base_model.layers[-5].output

    # Additional convolution layers for feature maps of varying sizes
    conv6 = Conv2D(1024, kernel_size=(3, 3), padding='same', activation='relu')(base_output)
    conv7 = Conv2D(1024, kernel_size=(1, 1), padding='same', activation='relu')(conv6)

    # Construct SSD model
    predictions = Conv2D(num_classes * 4, kernel_size=(3, 3), padding='same', activation='sigmoid')(conv7)
    predictions = tf.keras.layers.Reshape((-1, num_classes, 4))(predictions)

    model = tf.keras.models.Model(inputs=base_model.input, outputs=predictions)
    return model

# Example usage
input_shape = (300, 300, 3)
num_classes = 80
model = ssd_model(input_shape, num_classes)
model.summary()
```

Fig. 2: Construction of a Single Shot Multibox Detector (SSD) model for object detection tasks

stack of feature maps. These feature maps serve as inputs for the detection heads of the SSD architecture. The modular nature of MobileNet SSD allows for flexible modifications to adapt its architecture to specific requirements and constraints.

The architectural diagram illustrates the integration of MobileNet with the Single Shot Multibox Detector (SSD) framework, forming MobileNet SSD. This diagram visually depicts the workflow and components of the MobileNet SSD

architecture, showcasing the utilization of MobileNet as the base network for feature extraction and its seamless integration with the SSD framework for efficient object detection.

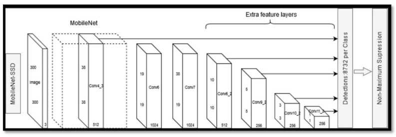


Fig. 3: Architecture of SSD MobileNet model.

4 Methodology

Our proposed system represents a significant advancement in the realm of local bus booking applications, aimed at revolutionizing the passenger experience through the integration of cutting-edge technologies and innovative features. By leveraging state-of-the-art solutions such as MobileNet SSD architecture, GPS tracking, and real-time security measures, our system strives to address key challenges faced by passengers while enhancing convenience, safety, and efficiency.

In the following subsections, we delve into the core components and features of our proposed system, outlining how each element contributes to an enhanced local bus booking experience. From real-time seat vacancy details to secure payment gateways and emergency response mechanisms, our system offers a comprehensive solution designed to meet the diverse needs of modern-day passengers.

4.1 Real-Time Seat Vacancy Details

Leveraging the MobileNet SSD architecture, our system provides passengers with real-time seat vacancy details. By analyzing bus interior images using machine learning algorithms, users can instantly view available seats, enhancing convenience and informed decision-making.

4.2 GPS Integration for Bus Tracking

Our system integrates GPS technology to enable users to track nearby buses on a map interface in real-time. This feature enhances transparency and efficiency, allowing passengers to plan their journeys more effectively by monitoring bus locations and arrival times.

4.3 Secure Payment Gateway

Facilitating seamless ticket booking, our system integrates a secure payment gateway directly within the app. Through relevant APIs, passengers can securely make payments for their bus tickets, eliminating manual transactions and streamlining the booking process for enhanced convenience.

4.4 Cross-Platform Compatibility and User Interface

Built using React Native for cross-platform compatibility, our app offers a user-friendly interface and seamless performance across various devices. Integration with Google Maps further enhances the user experience by providing accurate location data and navigation assistance.

The system architecture presents an overview of our local bus booking application. It integrates modern technologies to offer passengers an efficient and user-friendly experience. This diagram illustrates the main components and interactions within the system, highlighting its key functionalities.

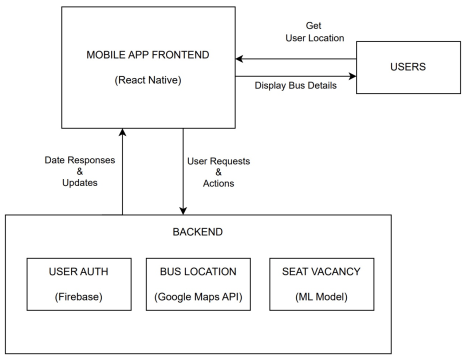


Fig. 4: Proposed System Architecture Diagram

5 Conclusion

In conclusion, the development and implementation of our local bus booking application represent a significant advancement in public transportation technology. By leveraging modern technologies such as React Native for frontend development, machine learning for real-time seat vacancy detection, and GPS integration for bus tracking, we have created a robust and user-friendly platform for passengers. The integration of features such as secure payment gateways and emergency response mechanisms further enhances the safety and convenience of our application. Through this research, we have demonstrated the effectiveness of utilizing MobileNet SSD architecture for real-time seat vacancy detection, providing valuable insights into improving public transportation services. Moving forward, we envision continued enhancements and optimizations to our application, driven by feedback from users and advancements in technology. Ultimately, our goal is to provide a seamless and efficient bus booking experience for passengers while contributing to the improvement of public transportation systems worldwide.

References

[1] A. P. Murdan, V. Bucktowar, V. Oree, and M. P. Enoch, "Low-cost bus seating information technology system," IET Intelligent Transport Systems, doi: 10.1049/iet-its.2019.0529.

[2] D. Patel, R. Seth, V. Mishra, and R. Pathari, "Real-Time Bus Tracking System," International Research Journal of Engineering and Technology (IRJET), vol. 04, no. 03, Mar. 2017.

[3] R. Sharma and Y. Sawant, "SMART BUS: Seat Occupancy Detection System," International Research Journal of Modernization in Engineering Technology and Science, vol. 03, no. 05, May 2021.

[4] J. Zhang, D. Shen, L. Tu, F. Zhang, C. Xu, Y. Wang, C. Tian, X. Li, B. Huang, and Z. Li, "A Real-Time Passenger Flow Estimation and Prediction Method for Urban Bus Transit Systems," IEEE Transactions on Intelligent Transportation Systems.

[5] K. Sujatha, P. V. Nageswara Rao, K. J. Sruthi, and A. Arjuna Rao, "Design and Development of Android Mobile-Based Bus Tracking System," in Proceedings of the 2014 First International Conference on Networks & Soft Computing (ICNSC2014), IEEE.

[6] O. Obulesu, A. Hanshika, A. Lahari, P. Arunodaya, and R. Ashwi, "People Counting and Tracking System in Real-Time Using Deep Learning Techniques," International Journal for Research in Applied Science and Engineering Technology, vol. 02, no. 04, Jun. 30, 2023.

[7] A. Bakliwal, A. Puranik, A. Modi, A. Jain, A. Jaiswal, D. Godani, P. Bhanodia, and P. Jangde, "Crowd Counter: An Application of Centroid Tracking Algorithm," International Research Journal of Modernization in Engineering Technology and Science, vol. 02, no. 04, Apr. 2020.

[8] Book- Puneet Kumar Aggarwal, Parita Jain, Jaya Mehta, Riya Garg, Kshirja Makar, and Poorvi Chaudhary, "Machine Learning, Data Mining and Big Data Analytics for 5G-Enabled IoT", Blockchain for 5G enabled IoT: the new wave for Industrial Automation, pp. 351-375, Springer, 2021.

[9] N. Gautam, A. Singh, K. Kumar, P K Aggarwal, Anupam, "Investigation on performance analysis of support vector machine for classification of abnormal regions in medical image", Journal of Ambient Intelligence and Humanized Computing, 2021.

[10] M.Singh, N.Sukhija, A.Sharma, M.Gupta, P.K.Aggarwal, "Security and Privacy Requirements for IoMT-Based Smart Healthcare System". Big Data Analysis for Green Computing, 17–37, Taylor Francis, 2021

[11] V. K. Reshma, Ihtiram Raza Khan, M. Niranjanamurthy, Puneet Kumar Aggarwal, S. Hemalatha, Khalid K. Almuzaini, and Enoch Tetteh Amoatey, "Hybrid Block-Based Lightweight Machine Learning-Based Predictive Models for Quality Preserving in the Internet of Things- (IoT-) Based Medical Images with Diagnostic Applications", Computational Intelligence and Neuroscience, Vol. 2022, pp. 1-14, 2022.

[12] Parita Jain, Puneet Kumar Aggarwal, Kshirja Makar, Riya Garg, Jaya Mehta, Poorvi Chaudhary, "Machine Learning in Risk Analysis" Applications of Computational Science in Artificial Intelligence to be published by IGI Global, pp. 190-213, 2022 ISBN: 978-1-7998-9012-6.

[13] Pranshu Saxena, Kanu Priya, Sachin Goel, Puneet Kumar Aggarwal, Amit Sinha, Parita Jain, "Rice Varieties Classification Using Machine Learning Algorithms", Journal of Pharmaceutical Negative Results, Vol. 13, Issue 7, pp. 3762-3772, 2022.

[14] Mandeep Singh, Megha Gupta, Anupam Sharma, Parita Jain and Puneet Kumar Aggarwal, "Role of Deep Learning in Healthcare Industry: Limitations, Challenges and Future Scope", Deep Learning for Healthcare Services, pp. 1-22 Bentham Science Publishers, 2023.

[15] Megha Gupta, Mandeep Singh, Anupam Sharma, Namrata Sukhija, Puneet Kumar Aggarwal, Parita Jain, "Unification of machine learning and blockchain technology in healthcare industry", Innovations in Healthcare Informatics: From interoperability to data analysis, IET Digital Library, 20th June 2023.