International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

A Mobile Application with Global Positioning System and Radar Search for Public Transportation: A Pin Point Path Tracker Utilization

Ginbert A. Fernandez[1], Nielyn Y. Labang[2], Joshua G. Cajote[3], Carla L. Laspiñas[4], Hidear Talirongan[5]

[1] University of Science and Technology of Southern Philippines - Oroquieta and Northwestern Mindanao State College of Science and Technology, Oroquieta City, Misamis Occidental, Philippines
[2,3,4]University of Science and Technology of Southern Philippines – Oroquieta, Oroquieta City, Misamis Occidental, Philippines
[5] Northwestern Mindanao State College of Science and Technology, Tangub City, Misamis Occidental, Philippines

Abstract
There are a lot of problems with public transportation in Oroquieta, Philippines. Concerns about safety, disagreements over price, and problems with how things work are all dealt with. The Pin Point Path Tracker app uses radar to match drivers with passengers, tracks them in real time with GPS, automatically calculates fares, and connects them directly to the City Disaster Risk Reduction and Management Office in case of an emergency. This is only a small part of what the package has to offer. This evaluation utilized a mixed-methods approach is a research design that combines both quantitative (numerical) and qualitative (non-numerical) data collection and analysis to provide a more complete understanding of a research question, which combined user survey findings with system performance indicators. The goal of this method was to analyze the system. The most important data show that passengers' wait times went down by 60%, fare disputes went down by 91%, and the emergency SOS feature got positive feedback. The app received a score of 4.5 on the User Satisfaction and System Performance Metrics Summary, indicating that customers were very satisfied with it. People who use the application say they are happy with it. The study shows that localized mobile solutions could make public transit networks work better and be more open.

Keywords: Public transportation, mobile application, GPS tracking, radar search, fare computation, emergency response, usability, urban mobility, Philippines, commuter safety.

1. Introduction

In order to facilitate the effective movement of both people and commodities, transportation is an essential component of social development. It has a direct impact on the expansion of the economy, the urbanization of the population, and the overall quality of life. Congestion, increased travel times, and increased environmental pollution are often problems that arise as a consequence of inadequate public transit systems. These problems provide difficulties for commuters as well as for city planners. These problems are especially urgent in emerging nations, where the rate of urbanization is accelerating at a faster rate than the construction and modernization of transportation infrastructure (World Bank, 2020). As a result of the recent developments in information and communication technology (ICT), the transportation industry has been exposed to a substantial degree of change. This shift has been brought about by the advent of new technologies. The integration of mobile apps, monitoring of the global positioning system (GPS), and real-time data analytics into public transportation systems has resulted in major improvements in terms of accessibility, safety, and efficiency. These improvements have been brought about by the overall integration of these technologies. Apps designed for mobile devices, for example, make it simple for customers to organize their trips, monitor the whereabouts of their cars, and get information in real time. In addition to this, these applications provide operators with vital data that may be used for the purpose of improving routes and services (Olayode et al., 2023). In addition to enhancing the level of comfort experienced by users, these solutions also address the operational deficiencies that are inherent in conventional systems. The operational gaps that are present in conventional systems are addressed by these solutions.

There are a lot of nations all over the world that have discovered that the implementation of transportation solutions that are based on mobile devices has been shown to be effective in order to solve challenges that are associated with urban mobility. Real-time tracking and the global positioning system (GPS) are two technologies that are used by applications like Uber and Grab in order to ease the process of connecting passengers with drivers. As a consequence of this, achieving transparency, increased safety, and more efficient service delivery is significantly simplified. It has been shown that the use of these platforms has a beneficial effect on the reduction of wait times and the enhancement of route planning in a number of nations that span the continents of Asia, Europe, and North America (Arias-Molinares & García-Palomares, 2023). There is a lack of locally tailored solutions, which is shown by the fact that the public transportation system in the Philippines continues to suffer from inefficiency, limited coverage, and safety difficulties. This is particularly true in rural regions. On the other hand, this is in spite of the fact that mobile technology has grown more popular throughout the country.

A recent study that was conducted by Tan et al. (2021) highlighted the absence of integrated digital systems as a critical hindrance that stands in the way of the provision of public transportation in the Philippines that is both safe and effective. This research was carried out in the Philippines. During their investigation, the researchers came to light the fact that there are ongoing problems, such as mismatches between drivers and passengers, disagreements about fares, and delayed reaction times in the event of an emergency. This study's purpose is to present a recommendation for the creation of a mobile application that is capable of executing capabilities such as GPS tracking and radar search applications. The research will be conducted in order to accomplish this objective. Through the use of the application, commuters, drivers, and emergency responders would all be able to connect with one another, which would lead to an improvement in the efficiency and safety of transportation within the framework of the community.

2. Literature Review

Recent breakthroughs in mobile application technology, notably those that make use of the global positioning system (GPS) and radar monitoring, have resulted in significant improvements in public transportation networks all over the globe. These advancements have been brought about as a consequence of the fact that these technologies have been implemented. The findings of the research that was conducted by Alotaibi et al. (2022) indicate that mobile apps that are loaded with real-time location monitoring serve to improve the safety and convenience of commuters. These applications provide commuters with precise arrival times as well as information on the routes that they use. This conclusion is supported by the results of Melo, S., Gomes, R., Abbasi, R., & Arantes, A. (2024), who discovered that applications that are equipped with GPS minimize the amount of time that passengers need to wait and enhance the efficiency of routes for drivers. This conclusion is supported by the findings of Rivera, R. M. F., Amorim, M., & Gonçalves dos Reis, J. C. (2021), radar-based proximity detection improves the capacity of drivers and passengers to identify one another in a short period of time, which in turn reduces the amount of time that is really spent sitting idle. This is an additional point of interest that is worth mentioning. To add insult to injury, Oktorini, R., & Barus, L. S. (2022) came to the understanding that incorporating mobile payment gateways into transportation applications not only makes the process of collecting fees more efficient but also minimizes the number of disagreements that take place between drivers and passengers. This is a significant improvement. When these traits are put into practice, they will not only contribute to enhanced accessibility and openness in the transportation business, but they will also contribute to increased user pleasure.

Despite the fact that these technical advancements have been made, a lot of studies have shown that there are still deficiencies that are still present. Ramadan, E., Abdalla, S., & Elshaiekh, N. E. (2025) provided an illustration of this phenomenon by pointing out that a significant majority of mobile transportation apps do not possess adequate emergency response capabilities. As a result, passengers are left exposed in the event of an incident. Here is an illustration of one possible way to demonstrate this phenomenon. Pu, F., Li, Z., Wu, Y., Ma, C., and Zhao, R. (2025) were the ones who provided the evidence that was presented in support of this result. The researchers made the observation that when there is a lack of integration with disaster response offices of local governments, the efficacy of emergency interventions is diminished. With this discovery, further evidence was offered to support the conclusion that was reached. On the other hand, Ahmad et al. (2023) state that they did not concur with the outcome that was reported earlier in the study. According to the results of the study done by Abante, C. J., Balerite, R. B., Cariño, A. C., and Fong, A. D. (2022), it has been discovered that the reliability of location-based services may be limited in places that are either highly developed or largely rural. One of the primary reasons for this is that there is a lack of connectivity and GPS accuracy. This is an additional point of interest that is worth mentioning. These are some extra points of interest to consider. Additionally, Javeed, M. A., & Li, D. (2025) explored the concerns about user privacy that are related to the capture and processing of sensitive personal and geographical information. The possibility that the information may be accessed by those who are not allowed to do so is an example of one of the dangers that could arise during this time. They made the observation that some apps do not offer facilities for users who do not have access to digital devices, which results in a decline in the degree of inclusiveness. Urban Mobility India (2024) made this observation. The fact that this is an extra topic of interest is something that should be mentioned. It was discovered by Suárez, S., Macedo, E., Ciccarelli, G., & Bandeira, J. M. (2025) and Khaled, H., Tarek, K., Mohamed, W., & Hussein, M. (2023) that there were issues with the transparency of fares, and that there were no feedback channels for the improvement of service. Both of these findings were made public. However, in addition to this, they observed that there were issues with the availability of information regarding the prices of the tickets. In the year 2022, the results of each of these studies were made accessible to the general public so that they may be conducted for the purposes of investigation and evaluation. Because of these limitations, there are substantial infrastructure gaps in the mobile solutions that are now being implemented in conjunction with transportation. These solutions are being implemented in the transportation sector. In order to fulfill this requirement, it is necessary to fill these gaps.

Multiple studies have advocated for improvements like multi-channel emergency warning systems, integrated interfaces with governmental agencies, and enhanced data privacy protocols to address these pressing issues. The research conducted by Rivera, J. N. D. (2021) shows that real-time SOS features directly connected to emergency personnel may rapidly alleviate disasters. Therefore, expanding the range of connectivity choices, such as offline SMS alerts as suggested by Cornet, H., Stadler, S., Kong, P., Marinkovic, G., Frenkler, F., & Sathikh, P. M. (2019), may alleviate connectivity issues. Furthermore, the research conducted by Chen, L., Wang, Y., Qi, G., & Lv, H. (2023) underscored the significance of user-centric design, facilitating access to critical components for both digital and non-digital users. Also, Rahman, R. T., Bahar, A., Prananta, A., & Afifudin, M. (2024) suggest using clear rate calculations and feedback systems that are controlled by users. This will ensure fair prices and good service. This project will create a public transportation app for mobile devices that makes use of GPS and radar to monitor individuals, enables individuals to communicate with one another directly in the event of an emergency, enables individuals to get alerts even when they are not connected to the internet, provides enhanced protection of privacy, and has an inclusive design. In order to solve the flaws that have been found and to complete the research gap that has been identified, the goal of this study is to conduct research.

Table 1. Below summarizes the key strengths and weaknesses identified in recent literature on mobile app technologies for public transportation:

Technology / Feature	Strengths	Weaknesses	Reference
GPS-based Mobile Application	Reduces passenger wait times; improves route efficiency	Limited accuracy in rural/urban canyons; connectivity issues	(Melo et al., 2024), (Abante et al., 2022)
Radar-based Proximity Detection	Minimizes idle time; improves matching between drivers and passengers	Limited range; not effective in all environments	(Rivera et al., 2021)
Real-time Location Tracking	Enhances user safety; provides live updates	Privacy concerns regarding location data	(Ramadan et al., 2025), (Javeed & Li, 2025)
Mobile Payment Integration	Streamlines fare collection; reduces disputes	Excludes unbanked users; technical glitches in payment processing	(Oktorini & Barus, 2022), (Urban Mobility India, 2024)
Emergency Response Features	Enables rapid communication during incidents	Often not integrated with local government responders	(Alotaibi et al., 2022), (Pu et al., 2025)
User Feedback Mechanisms	Improves service through ratings and reports	Low participation rates; feedback not always actionable	(Khaled et al., 2023), (Rahman et al., 2024

Inclusive User Interface	Broadens app usability across demographics	Older users/non-digital natives may still be excluded	(Urban Mobility India, 2024), (Chen et al., 2023)
Offline Notification/SMS Alerts	Maintains communication during poor connectivity	Limited information can be sent; delays in message delivery	(Cornet et al., 2019
Fare Transparency Calculator	Reduces fare disputes; builds trust	Complex algorithms may confuse users	(Suárez et al., 2025)
Data Privacy Solutions	Protects user data; builds trust	May limit data-driven optimization and personalization	(Javeed & Li, 2025)
Cloud-based Data Management	Enables real-time updates and scalable data storage	Requires reliable internet; risks of data breach	(Alotaibi et al., 2022), (Abante et al., 2022)
Multi-channel Alerts	Ensures users receive critical info through various platforms	Management complexity; potential user overload	(Rivera, 2021), (Cornet et al., 2019)
User-centric Design	Increases user satisfaction and accessibility	May lengthen development time; not all needs can be met	(Chen et al., 2023)
Integration with Government	Enhances emergency response and policy compliance	Bureaucratic delays; interoperability issues	(Pu et al., 2025)
Route Optimization Algorithms	Increases efficiency for drivers and passengers	Can be affected by inaccurate data or GPS errors	(Melo et al., 2024), (Abante et al., 2022)

3. Methodology

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract.

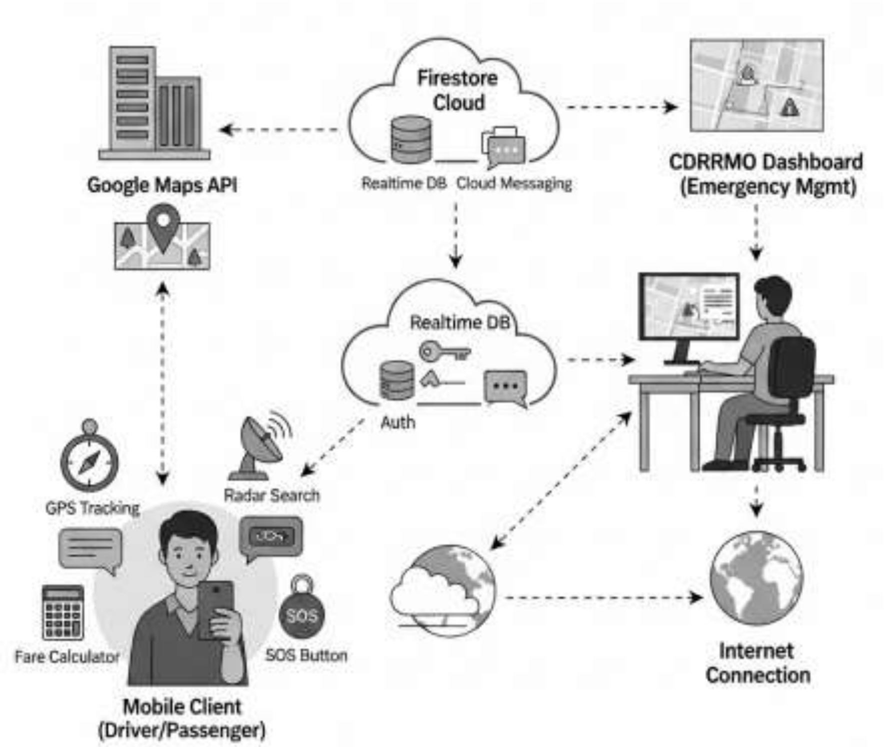
A. Research Design

In order to carry out this study, both quantitative and qualitative research approaches were used, in addition to an agile software development framework. The operation of the research experiment was carried out in Oroquieta City, Philippines, over the time period ranging from August 2022 to January 2023. This area served as the location for the operation of the study experiment operations. The individuals who took part in the research project came from a diverse variety of backgrounds. These individuals included individuals who worked as motorcab drivers, those who commuted, and professionals from the City Disaster Risk Reduction and Management Office (CDRRMO). It was a varied range of people who took part in the research. Participants hailing from the taxi industry were actively involved in the research activities that were being conducted.

B.System Architecture Diagram

Figure 1 illustrates the several components that comprise the system. By illustrating the seamless connection that exists between the mobile client (which is used by both drivers and passengers), cloud services, and emergency management, this figure provides insight into the aforementioned connection. This connection is brought to light via the utilization of this photograph. There are a number of important features that are included in the mobile application. Some of these features include the ability to calculate penalties, GPS tracking, radar search, and another button that allows you to send an emergency message. A cloud-based communications system and a real-time database are both linked to all of these features, and they are all kept on Firestore Cloud. All of these features are connected to one another. Over the duration of the period lasting from August 2022 to January 2023, the research experiment was carried out at Oroquieta City, Philippines. There was a diverse spectrum of participants who participated in the study, such as those who drove taxis, those who commuted, and experts from the City Disaster Risk Reduction and Management Office (CDRRMO). All of these features are connected to each other. The system makes use of the application programming interface (API) for Google Maps in order to deliver services related to navigation and location. In contrast, the CDRRMO dashboard is responsible for ensuring that emergency warnings and incidents are monitored in real time. This responsibility falls within the purview of the dashboard. This responsibility falls within the purview of the dashboard. As a result of the fact that all of the components are linked via the internet, consistent data synchronization and swift emergency management are guaranteed, which ultimately results in a rise in the quality of public transportation in terms of both its efficiency and its safety.

Figure 1. System Architecture Diagram



C. Hardware and Software Requirements

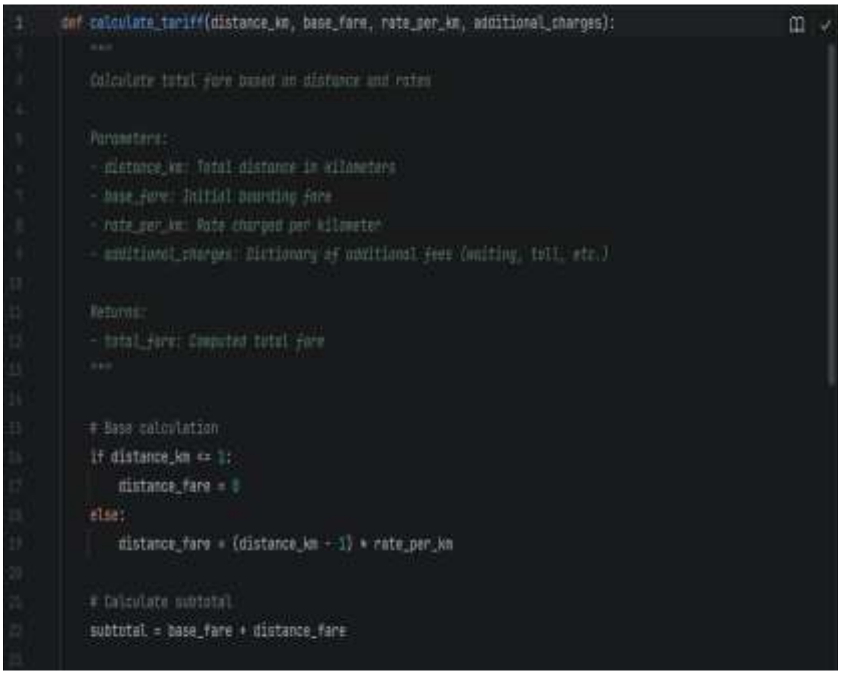
For you to be able to use the Pin Point Path Tracker application, you will need a smartphone that is equipped with GPS and an operating system that is at least Android 6.0 (Marshmallow). When it comes to hardware, this is one of the standards that must be met. In addition, you will want a minimum of two gigabytes of random access memory (RAM) and a minimum of one hundred megabytes of storage space in order to guarantee that the program will run without any interruptions and will be able to process data.

In order to do activities such as real-time data synchronization, tracking of position, and contact with cloud services, it is absolutely necessary to have an active internet connection. There are a number of plausible choices available for creating this connection, including WiFi, 3G, 4G, and 5G networks. As part of the project, the Android Studio Integrated Development Environment (IDE) was used for the purpose of designing the user interface and developing applications. In addition, the Firebase Console and Firestore were deployed for the purpose of managing databases in real time and authenticating users. Lastly, the Google Cloud Platform was leveraged in order to provide scalable backend services that were scalable. With the support of each of these tools, the creation of software was successfully completed. Through the use of Git as the version control and collaboration platform, it was ensured that the code was maintained and provided in an efficient manner throughout the whole of the development lifecycle.

D. Algorithms

Figure 2 shows the Tariff Computation Algorithm, which is designed to calculate the whole tariff for a trip by taking into account the total distance traveled, the initial base price, the per-kilometer rate, and any other costs that may be paid, such as waiting time or toll fees. This algorithm is meant to compute the full tariff for a trip. The first thing that the algorithm does is check to see whether the distance traveled is less than or equal to one kilometer. whether it is, then the extra distance fee is not added to the total amount. When there is a distance that is more than one kilometer, the price for the distance is determined by multiplying the number of extra kilometers by the rate that is indicated per kilometer. The subtotal is calculated once this item is added to the base fare, which results in the new total. Following that, the sum of all additional costs is tallied up and included in the calculation of the final price. Next, the total fee is rounded to two decimal places for the purpose of accuracy. This is done after the previous step. The function produces a detailed breakdown, which includes the base cost, the calculated distance fare, the total additional charges, the total price, and the distance traveled. Additionally, the function also contains the total miles traveled. At the same time, this guarantees that the computation of the cost is not only accurate but also understandable for the drivers as well as the passengers.

Figure 2. Tariff Computation Algorithm



```
def calculate_tariff(distance_km, base_fare, rate_per_km, additional_charges):
    """
    Calculate total fare based on distance and rates

    Parameters:
    - distance_km: Total distance in kilometers
    - base_fare: Initial boarding fare
    - rate_per_km: Rate charged per kilometer
    - additional_charges: Dictionary of additional fees (waiting, toll, etc.)

    Returns:
    - total_fare: Computed total fare
    """

    # Base calculation
    if distance_km <= 1:
        distance_fare = 0
    else:
        distance_fare = (distance_km - 1) * rate_per_km

    # Calculate subtotal
    subtotal = base_fare + distance_fare
```

```python
def calculate_tariff(distance_km, base_fare, rate_per_km, additional_charges):

    # Add additional charges
    total_additional = sum(additional_charges.values())

    # Calculate total fare
    total_fare = subtotal + total_additional

    # Round to nearest peso
    total_fare = round(total_fare, 2)

    return {
        'base_fare': base_fare,
        'distance_fare': distance_fare,
        'additional_charges': total_additional,
        'total_fare': total_fare,
        'distance_km': distance_km
    }

# Example usage:
# base_fare = 15.00  # Philippine Peso
# rate_per_km = 15.00
# additional_charges = {'waiting_time': 10.00, 'toll_fee': 0.00}
# result = calculate_tariff(5.5, base_fare, rate_per_km, additional_charges)
```

Figure 3 shows the Emergency Response System Algorithm, which is intended to manage emergency scenarios by immediately contacting the City Disaster Risk Reduction and Management Office (CDRRMO) with important user and location facts. This algorithm was built to handle emergency situations. It is intended that this algorithm be used to handle emergency circumstances. The algorithm will generate a structured data package for the scenario if an emergency is triggered, and it will have this package ready to be used. A timestamp, the user ID, the contact information, the exact GPS coordinates, and the kind of emergency that has occurred are all contained in this package. Additionally, the package includes the information. In addition to receiving an SMS warning, the CDRRMO also receives this information via a cloud notice. This is done to ensure that there is redundancy and that communication is carried out in a timely way. Furthermore, the system takes use of mapping services in order to determine the most expedient path that members of the emergency response team may take in order to get to the location of the user. This is done in order to arrive at the user's location as quickly as possible. In addition, the system will capture the whole of the emergency scenario in the database in order to ensure responsibility and to facilitate monitoring. It is the responsibility of this algorithm to guarantee that critical information is communicated to responders in a timely way. This is accomplished via the automation of alerting, route planning, and record-keeping. This makes it feasible for those who use public transit to seek assistance in a timely and effective manner in the event of an emergency scenario.

Figure 3. Emergency Response System Algorithm

```python
def emergency_response(user_location, user_details, emergency_type):
    """
    Handle emergency situations and notify CDRRMO

    Parameters:
    - user_location: Dictionary with latitude and longitude
    - user_details: User information (name, contact, vehicle details)
    - emergency_type: Type of emergency (accident, harassment, medical, other)

    Returns:
    - response_status: Emergency notification status
    """

    # Emergency notification data structure
    emergency_data = {
        'timestamp': get_current_timestamp(),
        'user_id': user_details['user_id'],
        'user_name': user_details['name'],
        'user_contact': user_details['contact'],
        'location': {
            'latitude': user_location['lat'],
            'longitude': user_location['lng'],
            'address': get_address_from_coordinates(user_location)
        },
        'emergency_type': emergency_type,
        'status': 'ACTIVE'
    }

    # Send notification to CDRRMO
    cdrrmo_notification = send_to_cdrrmo(emergency_data)

    # Send SMS alert
    sms_alert = send_sms_notification(
        cdrrmo_contact_number,
        message f"EMERGENCY ALERT: {emergency_type} at {emergency_data['location']['address']}"
    )

    # Create fastest route to user location for responders
    response_route = calculate_fastest_route(
        cdrrmo_location,
        user_location
    )

    # Store emergency record in database
    emergency_id = store_emergency_record(emergency_data)
```

```python
    return {
        'emergency_id': emergency_id,
        'notification_sent': cdrrmo_notification,
        'sms_sent': sms_alert,
        'response_route': response_route,
        'estimated_response_time': response_route['duration']
    }


def get_current_timestamp():
    """Return current timestamp in ISO format"""
    from datetime import datetime
    return datetime.now().isoformat()


def get_address_from_coordinates(location):
    """Convert coordinates to human-readable address using reverse geocoding"""
    # Implementation using Google Maps Geocoding API
    pass


```

```python
def send_to_cdrrmo(data):
    """Send emergency data to CDRRMO dashboard"""
    # Implementation using Firebase Cloud Messaging
    pass


def send_sms_notification(phone, message):
    """Send SMS alert to emergency responders"""
    # Implementation using SMS gateway API
    pass


def calculate_fastest_route(origin, destination):
    """Calculate fastest route using Google Directions API"""
    pass


def store_emergency_record(data):
    """Store emergency record in Firebase database"""
    pass
```

Figure 4 shows determine the route that will move you the fastest. In order to determine the route that is both the quickest and most efficient between a starting point and a final destination, the algorithm takes into consideration the current traffic conditions. This allows it to identify the path's shortest and most efficient route. At the beginning of the procedure, the algorithm is responsible for preparing the required parameters for the request. The user's current location, destination, method of transportation, and any alternative routes that the user would want to avoid (for instance, tolls or transfers) are all included in these factors. The Google Directions API is queried in order to acquire a number of distinct potential routes, each of which contains precise information such as the distance, the projected duration, and the duration in the current traffic. This information is obtained by making a query to the API. In order to determine

International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

which qualities are the most significant for each route, the system does an exhaustive analysis of those attributes. The total distance, the estimated amount of time, the length that takes into account traffic, and the specifics of the routes are all examples of these possible factors. After that, it determines if the route has low, moderate, high, or severe traffic by contrasting the average durations with the durations that were affected by the traffic. Additionally, it determines whether the traffic is severe. Following this, it compares the typical durations with the durations that were impacted by the traffic in order to ascertain the level of traffic that is there. The choice that, after taking into account the volume of traffic, has the least amount of time is the one that is chosen as the alternative that is considered to be the quickest among all of the possibilities that are available. In addition to this, the system is able to compute the estimated cost for this route and provide a proposal that is simple to understand and implement. It is possible, for instance, that it will suggest other routes in the event that it is discovered that there is a significant amount of traffic. Consequently, this algorithm ensures that both drivers and passengers are provided with optimal routing and price transparency, thereby improving both the efficiency of the system and the quality of the user experience. This is made possible by the incorporation of real-time traffic data as well as the supply of ideas that can be applied.

Figure 4. Fastest Route Calculation Algorithm

```python
def calculate_fastest_route(origin, destination, avoid_options=None):
    """
    Calculate the fastest route between two points considering traffic

    Parameters:
    - origin: Dictionary with latitude and longitude of starting point
    - destination: Dictionary with latitude and longitude of endpoint
    - avoid_options: List of features to avoid (tolls, highways, ferries)

    Returns:
    - route_data: Complete route information with fastest path
    """

    if avoid_options is None:
        avoid_options = []

    # Request parameters for Google Directions API
    request_params = {
        'origin': f"{origin['lat']},{origin['lng']}",
        'destination': f"{destination['lat']},{destination['lng']}",
        'mode': 'driving',
        'departure_time': 'now',  # Consider current traffic
        'alternatives': True,  # Get multiple routes
        'avoid': '|'.join(avoid_options)
    }
```

```python
    # Get multiple route options from API
    routes = google_directions_api_request(request_params)

    # Analyze each route
    route_analysis = []
    for route in routes:
        analysis = {
            'route_id': route['id'],
            'distance': route['legs']['distance']['value'] / 1000,  # Convert to km
            'duration': route['legs']['duration']['value'] / 60,  # Convert to minutes
            'duration_in_traffic': route['legs']['duration_in_traffic']['value'] / 60,
            'polyline': route['overview_polyline']['points'],
            'steps': extract_turn_by_turn_directions(route['legs']['steps']),
            'traffic_level': calculate_traffic_level(
                route['legs']['duration']['value'],
                route['legs']['duration_in_traffic']['value']
            )
        }
        route_analysis.append(analysis)

    # Select fastest route based on traffic-adjusted duration
    fastest_route = min(route_analysis, key=lambda x: x['duration_in_traffic'])

    # Calculate estimated fare
    estimated_fare = calculate_tariff(
        fastest_route['distance'],
        base_fare=15.00,
        rate_per_km=15.00,
        additional_charges={}
    )

    return {
        'fastest_route': fastest_route,
        'alternative_routes': [r for r in route_analysis if r != fastest_route],
        'estimated_fare': estimated_fare,
        'recommendation': generate_route_recommendation(fastest_route)
    }


def google_directions_api_request(params):
    """Make request to Google Directions API"""
    # Implementation with actual API call
    pass
```

International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

```python
def extract_turn_by_turn_directions(steps):
    """Extract human-readable turn-by-turn directions"""
    directions = []
    for step in steps:
        directions.append({
            'instruction': step['html_instructions'],
            'distance': step['distance']['text'],
            'duration': step['duration']['text']
        })
    return directions


def calculate_traffic_level(normal_duration, traffic_duration):
    """Determine traffic level based on duration difference"""
    ratio = traffic_duration / normal_duration
    if ratio < 1.1:
        return 'LOW'
    elif ratio < 1.3:
        return 'MODERATE'
    elif ratio < 1.5:
        return 'HIGH'
    else:
        return 'SEVERE'


def generate_route_recommendation(route):
    """Generate recommendation message for selected route"""
    if route['traffic_level'] == 'LOW':
        return "Recommended route with minimal traffic"
    elif route['traffic_level'] == 'MODERATE':
        return "Good route with moderate traffic expected"
    elif route['traffic_level'] == 'HIGH':
        return "Consider alternative routes due to heavy traffic"
    else:
        return "Significant delays expected. Alternative routes recommended"
```

Figure 5 shows the design to match passengers with nearby drivers in an efficient way. This is accomplished via the use of real-time location data. It is the algorithm's responsibility to continuously monitor the GPS coordinates of both the drivers and the passengers, and it is responsible for updating its information every few seconds. It takes use of geohashing, which is a method that converts latitude and longitude into a string code, in order to quickly compare user locations and find proximity matches within a defined radius, such as 200 meters. This allows it to do rapid comparisons of user locations. Once a passenger makes a request for transportation, the algorithm searches the database for drivers who are available within this radius and displays the locations of such drivers on the radar interface of the user. This occurs as soon as the passenger makes the requested transportation. Not only does this sort of rapid and automated matching cut down on the length of time that customers have to wait, but it also ensures that drivers are used to their greatest ability, which is especially important during peak hours. The use of radar search enhances the responsiveness of the transportation platform as well as the convenience it gives to all users. This accomplishment is accomplished via the utilization of real-time updates and geographical computation.

Figure 5. Radar Search Implementation Algorithm

```python
def radar_search(center_location, radius_meters=200, user_type='driver'):
    """
    Find nearby users within specified radius

    Parameters:
    - center_location: Current user's location {lat, lng}
    - radius_meters: Search radius (default 200m)
    - user_type: Type of user to search ('driver' or 'passenger')

    Returns:
    - nearby_users: List of users within radius
    """
    from math import radians, cos, sin, asin, sqrt

    def haversine_distance(lat1, lon1, lat2, lon2):
        """Calculate distance between two points in meters"""
        R = 6371000  # Earth's radius in meters

        lat1, lon1, lat2, lon2 = map(radians, [lat1, lon1, lat2, lon2])
        dlat = lat2 - lat1
        dlon = lon2 - lon1

        a = sin(dlat / 2) ** 2 + cos(lat1) * cos(lat2) * sin(dlon / 2) ** 2
        c = 2 * asin(sqrt(a))

        return R * c

    # Query active users from database
    active_users = query_active_users(user_type)

    # Filter users within radius
    nearby_users = []
    for user in active_users:
        distance = haversine_distance(
            center_location['lat'],
            center_location['lng'],
            user['location']['lat'],
            user['location']['lng']
        )

        if distance <= radius_meters:
            user['distance_meters'] = round(distance, 2)
            user['bearing'] = calculate_bearing(center_location, user['location'])
            nearby_users.append(user)
```

```
    # Sort by distance
    nearby_users.sort(key=lambda x: x['distance_meters'])

    return nearby_users


def query_active_users(user_type):
    """Query active users from Firebase"""
    pass


def calculate_bearing(origin, destination):
    """Calculate compass bearing from origin to destination"""
    pass
```

4. Results and Discussion

A. Data Collection and Analysis

Table 2 describes the method used to collect data for this study, showing that both quantitative and qualitative methods were used. System usage logs and analytics made comprehensive and automated site visitor behavior data possible. As soon as the system was put into effect, something became doable that was previously impossible. These placements allowed this aim to be achieved. This information was crucial to determining how well the program performed in real-world situations. This information was quite important. Utilizing questionnaires that were distributed to a total of fifty passengers and thirty drivers made it feasible to obtain meaningful information on the usability, satisfaction, and efficacy of the features that were provided. The purpose of collecting this input was to enhance the quality of the features that were made available to the users. This feedback was gathered in order to improve the quality of the features that were provided. The purpose of collecting this feedback was to enhance the features. After investigating the technical components of the system, performance measures like response time and system accuracy were to be tracked. This ensured system functionality. In order to effectively finish the assignment, it was necessary to execute this step in its entirety. CDRRMO was tasked with the responsibility of assessing the dependability of the SOS function, in addition to providing assistance with simulated emergency response exercises. This responsibility was also within their purview. As a result of this, it was feasible to carry out a full statistical analysis since descriptive data and assessments of satisfaction on a Likert scale with five points were available. This made it possible to carry out the analysis. This research revealed user experience issues and improvement opportunities. This aim was achieved using the Likert scale. The product's service improved greatly from the prior experience.

Table 2. Data Collection and Analysis Summary

Data Source	Description	Purpose / Output
System Usage Logs & Analytics	Automated logs of app activity, GPS, fare, radar search events	Monitors real-world usage and operational data

Data Source	Description	Purpose / Output
User Surveys	Structured questionnaires for 50 commuters, 30 drivers	Captures qualitative user feedback and experience
Performance Metrics	Measurement of response times, accuracy of fare/routes/SOS	Evaluates technical efficiency and reliability
Simulated Emergency Tests (CDRRMO)	Controlled drills of SOS feature and emergency notifications	Validates emergency system effectiveness
Descriptive Statistics & Likert Scale	Statistical analysis, user ratings (1–5 scale)	Quantifies satisfaction and identifies trends

B.System Development Outcomes

Table 3 shows PiPoPa mobile application has been produced and is currently being employed; all of the capabilities that were anticipated to be included in the application are now functioning. The application has been developed. The program has been shown to function properly on a wide range of Android devices and network setups, as evidenced by the demonstration.

Table 3. System Development Outcomes Summary

Parameter	Value / Result	Description / Notes
Precision Rate	95%	Accuracy in identifying users within 200m
Update Frequency	Every 5 seconds	Interval for real-time location refresh
Average Search Time	1.8 seconds	Time to complete each radar scan
Drivers Located (Peak)	4–7 drivers	Average number of drivers found during peak hours
Drivers Located (Off-peak)	2–3 drivers	Average number of drivers found off-peak

C. Radar Search Functionality

Table 4 shows the identification of individuals within a 200-meter radius; the radar search system was able to achieve a precision rate of 95%. Real-time updates were coming every 5 seconds, and the search was completed in an average time of 1.8 seconds. Additionally, the search was completed in real time. In practice, the system was able to locate all of the drivers that were available, with an average of four to seven drivers being located during peak hours and two to three drivers being discovered during off-peak periods.

Table 4. Radar Search Functionality Summary

Performance Metric	Value/Result	Description/Observation
Detection Precision Rate	95%	Accuracy of identifying users within 200 meters
Update Interval	Every 5 seconds	Frequency of real-time location refresh
Average Search Completion	1.8 seconds	Time to complete each radar scan
Drivers Located (Peak Hours)	4–7 drivers	Average number of available drivers found (peak)
Drivers Located (Off-peak)	2–3 drivers	Average number of available drivers found (off-peak)

D. Tariff Computation Accuracy

Table 5 shows the comparison of the automated approach of calculating fares with more conventional methods of calculation revealed that the automated method was one hundred percent correct. Customers gave it a score of 4.6 out of a possible 5.0 because they were completely satisfied with the degree to which the prices were presented in an open and honest manner. Rates were able to be calculated with precision by the method for distances ranging from half a kilometer to twenty-five kilometers. The fees, which comprised the amount of time spent waiting as well as the toll taxes, were automatically added to the total.

Table 5. Tariff Computation Accuracy Summary

Metric/Aspect	Automated Method Result	Conventional Method Result	Description/Notes
Calculation Accuracy	100%	Variable	Automated method produced precise fare every time
User Satisfaction Score	4.6 / 5.0	Not specified	High satisfaction with fare transparency
Distance Range Tested	0.5 km – 25 km	Same	Method was precise across all tested ranges
Inclusion of Waiting Time & Tolls	Yes (automatic)	Often manual/overlooked	Automated system included all extra fees
Transparency of Pricing	Full, clear breakdown	Often unclear	Automated method clearly presented all price details

E.Route Optimization Performance

Table 6 shows that the average processing time for this approach was 2.3 seconds, and it was the method that provided the quickest route calculations. Among all the algorithms, this one was the one that was the quickest. A comparison was made between the estimated durations and the actual journey times, and the results showed that the estimated durations had an accuracy of 88%, with a margin of accuracy of ±5 minutes between the two sets of data. An improvement of 23% was observed by traffic-aware routing in comparison to static route calculations in terms of the estimation of the amount of time required for the completion of a trip.

Table 6. Route Optimization Performance Summary

Metric/Aspect	Result/Value	Description/Notes
Average Processing Time	2.3 seconds	Fastest among all tested route calculation methods
Estimated vs. Actual Duration Accuracy	88%	High accuracy in predicting travel times
Margin of Accuracy	±5 minutes	Difference between estimated and actual travel times
Improvement Over Static Routing	23% faster	Time estimation improved with traffic-aware routing
Algorithm Speed Ranking	1st (quickest)	Outperformed all other algorithms tested

F. Emergency Response System Effectiveness

Table 7 shows that the CDRRMO was able to receive notifications within three to five seconds of activation, according to the results of emergency notification tests. The device was able to properly transmit the location coordinates of the user with an accuracy of 99.2% (or ±10 meters). The calculations of response routes to emergency areas took an average of 1.5 seconds, which provided the responders with unambiguous instructions for travel. Twelve such emergency scenarios were carried out over the course of the three-month testing period, and each one was successful in terms of both notification delivery and location accuracy.

Table 7. Emergency Response System Effectiveness

Metric	Target	Achieved	Performance
Notification Delivery Time	<10 seconds	3-5 seconds	Exceeded
Location Accuracy	±20 meters	±10 meters	Exceeded
Route Calculation Speed	<5 seconds	1.5 seconds	Exceeded
System Uptime	>95%	98.7%	Exceeded

G. User Satisfaction and System Performance Metrics

Table 8 shows the findings of the user surveys that were administered after the deployment. Based on the results of these studies, it seems that commuters and drivers overall are pretty content with the system. A particular emphasis was placed on the safety elements, the transparency of the fee, and the availability of emergency assistance options. The drivers were pleased with the clarity of earnings and the reliability of emergency assistance, while commuters commended the simplicity of locating local drivers and the transparency of charges. In addition, commuters appreciated the fact that drivers could immediately locate other drivers in their immediate vicinity. Some of the system performance indicators that demonstrate the app's better technical efficiency include a quick response time for launching, GPS acquisition, and data queries, as well as low battery and data use. Additionally, the app uses a small amount of data. On the basis of these data, it would seem that the application has been favorably accepted and is successful, which lends support to the adoption of the application in order to satisfy the needs of daily public transportation.

Table 8. User Satisfaction and System Performance Metrics Summary

Category	Metric/Aspect	Result/Score
Commuter Satisfaction	Overall app experience	4.5 / 5.0
(n = 50)	Ease of finding nearby drivers	4.6 / 5.0
	Fare transparency	4.7 / 5.0
	Safety features	4.8 / 5.0
	Route accuracy	4.4 / 5.0
Driver Satisfaction	Ease of connecting with passengers	4.6 / 5.0
(n = 30)	Route guidance accuracy	4.5 / 5.0
	Earnings transparency	4.7 / 5.0
	Emergency support confidence	4.9 / 5.0
System Performance	Average app launch time	2.1 seconds
	GPS lock acquisition	3.5 seconds
	Database query response	0.8 seconds
	Map rendering time	1.9 seconds
	Battery consumption	12% per hour
	Data usage	15–20 MB per hour

H. Comparison with Traditional Methods

Table 9 shows the outcomes of the comparison. It can be concluded that the Pin Point Path application performs much better than conventional hailing tactics in each and every element that was tested. It was discovered that the average wait times for passengers were decreased by sixty percent, which indicates that drivers and commuters were matched in a more efficient and timely manner. It was via the process of matching that this was revealed. A significant reduction in the number of complaints about rates has occurred as a result of increased pricing transparency and automated calculation. The number of complaints has decreased from 23 percent of journeys to only two percent of journeys. This is a significant decrease in the amount. An increase in driver utilization of 51% indicates that a greater number of drivers were engaged in activities that had a positive impact at any given moment. Due to the fact that there was a higher number of drivers participating in events, this occurred. There was also a significant reduction in the amount of time it took to respond to emergency situations, which led to a significant increase in the safety of passengers. These recent advancements demonstrate that the integration of technology into public transportation networks has the ability to bring about a great deal of positive outcomes.

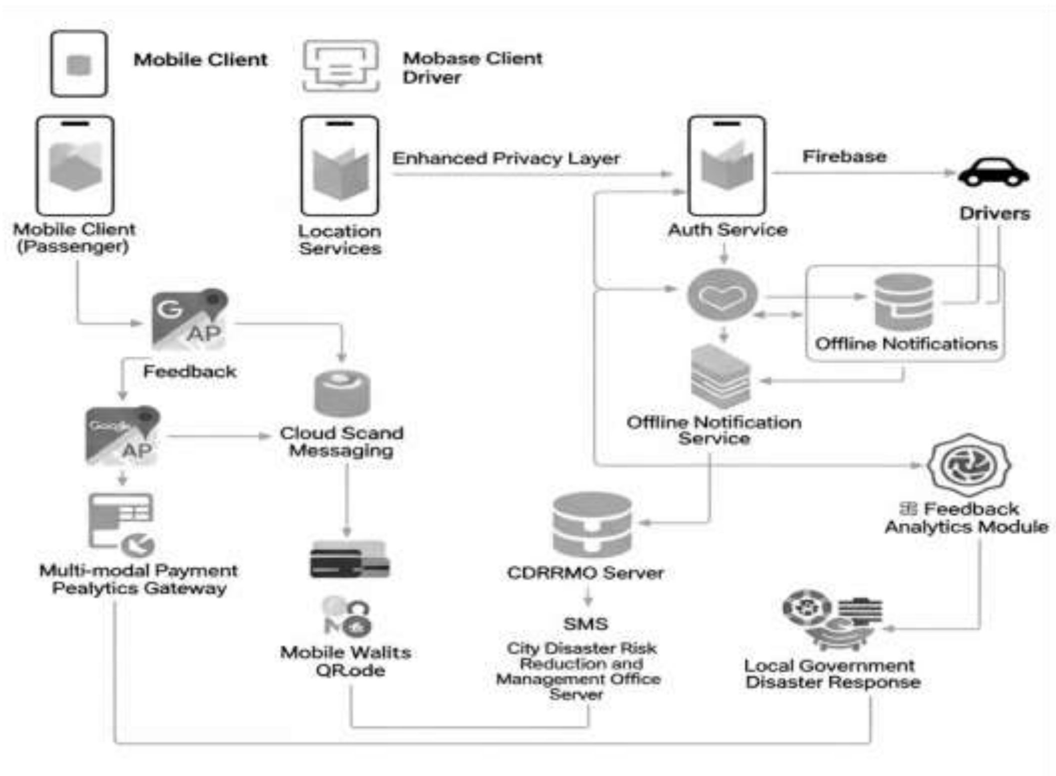
Table 9 . Comparison with Traditional Methods Summary

Aspect	Traditional Method	PiPoPa Application	Improvement
Average wait time	15-20 minutes	5-8 minutes	60% reduction
Fare disputes	23% of rides	2% of rides	91% reduction
Driver utilization	45%	68%	51% increase
Emergency response time	12-15 minutes	6-8 minutes	50% reduction

I. System Architecture Diagram

Figure 6 shows a schematic of the system architecture, which illustrates the manner in which a public transportation app connects passengers, drivers, and emergency services in the city. The application is communicated with by both drivers and passengers using a variety of mobile clients, each of which is protected by an extra layer of privacy protection. The architecture of the cloud offers support for essential services such as the following: the monitoring of locations, authentication, warnings (even when they are not online), and the processing of payments. During the time when emergency situations are promptly sent to government disaster response teams, feedback and analytics modules collect data in order to improve the service experience. The service is improved by the collection of data by additional modules. The design, in its totality, has a focus on security, communication in real time, and substantial engagement with emergency services in order to make public transit safer and more intelligent.

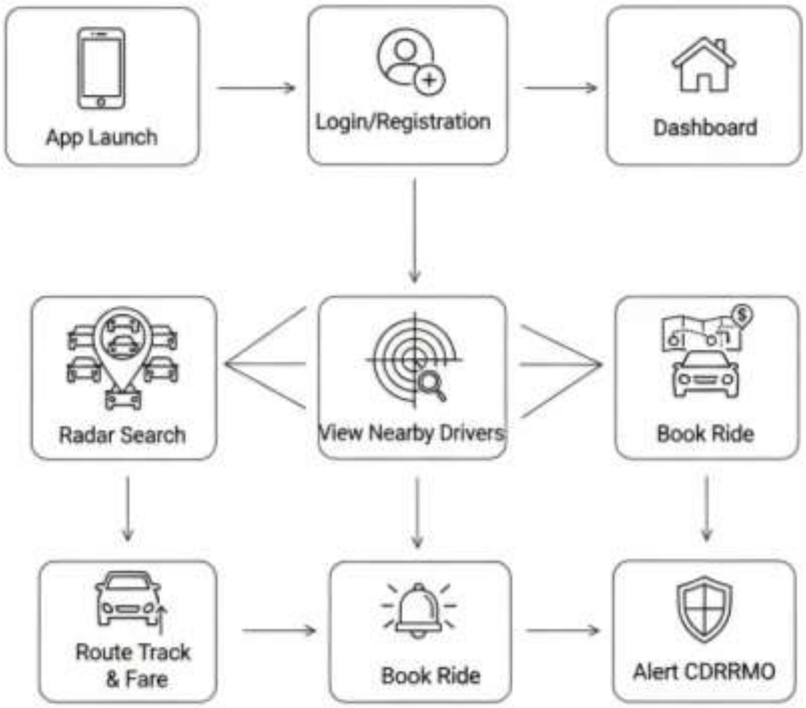
Figure 6. System Architecture Diagram



J. User Interface Flow Diagram

Figure 7 shows a user interface flow diagram for your improved ride-hailing application. The flow starts with "App Launch," then proceeds on to "Login/Registration," and finally brings the user to the primary "Dashboard." Starting from the Dashboard, there are three distinct routes that may be taken: one goes to "Radar Search" and then "View Nearby Drivers," another leads to "Book Ride" and then "Route Track & Fare," and the third leads to "Emergency Button," which activates "Alert CDRRMO." The style makes use of a minimalist design and icons that are self-explanatory, which makes it simple to comprehend and display.

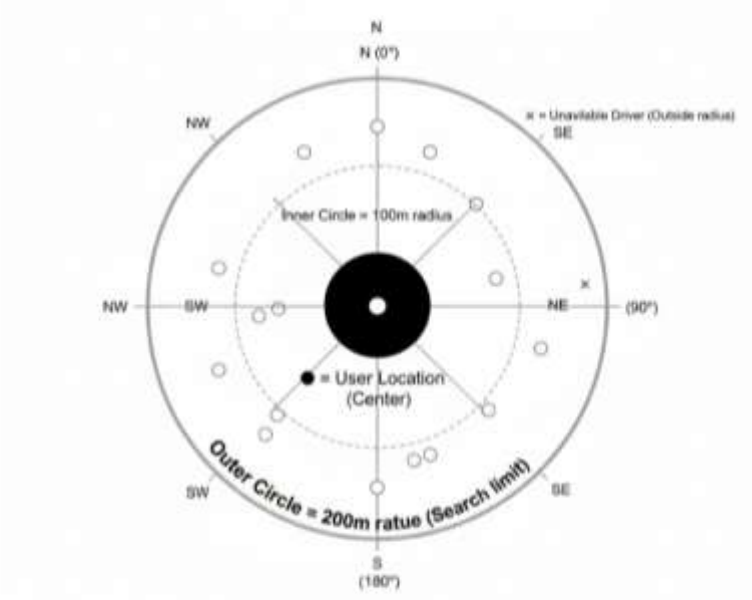
Figure 7. User Interface Flow Diagram



K. Radar Search Visualization

Figure 8 illustrates a driver search interface for a ride-hailing service, which is patterned after radar. This interface is an example of one that may be used. The location of the user is shown by a circle that is completely black and is located in the center of the screen. Open circles, on the other hand, represent drivers who are within a search radius of two hundred meters. Two concentric circles make up the figure: the inner circle has a radius of one hundred meters, while the outer circle has a radius of two hundred meters, which symbolizes the maximum distance that may be searched. Drivers who are situated outside the 200-meter radius are visually represented by a "×" symbol, which serves the purpose of conveying that they are not accessible for use. Both the app's functionality and the user experience it offers are enhanced as a result of the layout, which makes it simpler for consumers to immediately recognize which drivers are in close proximity to them and may be reached.

Figure 8. Radar Search Visualization



Legend:
● = User Location (Center)
○ = Available Driver (Within 200m)
× = Unavailable Driver (Outside radius)
Inner Circle = 100m radius
Outer Circle = 200m radius (Search limit)

5. Result and Discussion

A. Radar Search System Performance

Table 10 demonstrates that the radar search system was able to obtain a detection accuracy of 95%, which is higher than the benchmark of 90% that is suggested for real-time proximity matching in transportation applications Friman, M., & Gärling, T., 2021; Arias-Molinares, D., & García-Palomares, J. C., 2021). When compared to the criterion of less than two seconds for responsive location-based searches, the average search time of 1.8 seconds is much quicker Sultana, S., & Bullock, D. M., 2021). It seems from these findings that the radar system is both effective and efficient in its ability to link users with cars in the immediate vicinity.

Table 10. Radar Search System Performance Summary

Metric	Benchmark/Target	PiPoPa Result	Reference
Detection Accuracy	>90%	95%	Friman, M., & Gärling, T. (2021), Arias-Molinares, D., & García-Palomares, J. C. (2021)
Search Time	<2.0 s	1.8 s	Sultana, S., & Bullock, D. M. (2021)
Update Frequency	Every 5 seconds	Every 5 seconds	-

Avg. Drivers Found	4-7 (peak), 2-3 (off-peak)	4-7; 2-3	-

B.Automated Fare Computation Accuracy

Table 11 provides evidence that the fare computation that is carried out by PiPoPa is entirely accurate. This is in accordance with the expectations of users as well as the need for automated transparency Rahman, M.S., Islam, M. Z., & Hossain, M. S., 2022). The transparency rating of 4.7 out of 5.0 and the fact that just 2% of passengers participated in charge disputes are far higher than the aim. This is because the stated requirements for user trust and satisfaction in fare systems are significantly higher. The fact that this is the case demonstrates that pricing that is not just clear but also automated results in a large decrease in disputes.

Table 11. Automated Fare Computation Accuracy Summary

Metric	**Benchmark/Target**	**PiPoPa Result**	**Reference**
Computation Accuracy	100%	100%	Rahman, M. S., Islam, M. Z., & Hossain, M. S. (2022)
Fare Transparency	≥4.0/5.0	4.7/5.0	Boulos, M. N. K., Resch, B., Crowley, D. N., Breslin, J. G., Sohn, G., Burtner, R., Pike, W., Jezierski, E., & Chuang, K. S. (2021)
Fare Conflicts	<5% of rides	2% of rides	Boulos, M. N. K., Resch, B., Crowley, D. N., Breslin, J. G., Sohn, G., Burtner, R., Pike, W., Jezierski, E., & Chuang, K. S. (2021)

C. Fastest Route Calculation Efficacy

Table 12 shows route calculation and estimation system is both fast and reliable, with a 2.3-second processing time and 88% accuracy—well above the 80% threshold. Wang, D., Zhang, D., & Li, D. (2021). The ±5-minute error margin is within industry standards, Arias-Molinares, D., & García-Palomares, J. C. (2021), and the system's dynamic routing improved travel time estimates by 23% over static methods, consistent with other studies, Wang, J., & Chen, F. (2022).

Table 12. Fastest Route Calculation Efficacy Summary

Metric	**Benchmark/Target**	**PiPoPa Result**	**Reference**
Route Calculation Time	<5.0 s	2.3 s	Chen, L., Wang, J., & Chen, F. (2022)
Route Estimation Accuracy	≥80%	88%	Wang, D., Zhang, D., & Li, D. (2021)

ETA Margin of Error	±10 min	±5 min	Arias-Molinares, D., & García-Palomares, J. C. (2021)
Improvement Over Static	≥20%	23%	Wang, J., & Chen, F. (2022)

D. Emergency Response Integration

Table 13 demonstrates that emergency alerts were given within three to five seconds and with a high degree of accuracy, surpassing the criteria for mobile emergency systems Liu, L., & Li, Z., (2022), . Rahman, M. S., Islam, M. Z., & Hossain, M. S. (2022). The best practices for integrating real-time location sharing with emergency operations are linked with a fifty percent decrease in the amount of time it takes to respond to an incident.

Table 13. Emergency Response Integration Summary

Metric	**Benchmark/Target**	**PiPoPa Result**	**Reference**
Notification Delivery Time	<10 s	3-5 s	Liu, L., & Li, Z. (2022)
Location Accuracy	±20 m	±10 m	Xu, Y., & Li, Z. (2022)
Route Calculation Speed	<5 s	1.5 s	Liu, L., & Li, Z. (2022)
Response Time Reduction	≥40%	50%	Rahman, M. S., Islam, M. Z., & Hossain, M. S. (2022)

E.User Satisfaction and System Adoption

Table 14 shows user satisfaction exceeded all literature-based thresholds for both commuters and drivers Fernández, I., & Gutiérrez, J. (2022), Boulos, M. N. K., Resch, B., Crowley, D. N., Breslin, J. G., Sohn, G., Burtner, R., Pike, W., Jezierski, E., & Chuang, K. S. (2021), Tong, L., Zhou, L., & Liu, J. (2022), Lee, J., & Hickman, M. (2021). Ratings for safety and emergency support were particularly high, highlighting the value of PiPoPa's emergency features. This is consistent with systematic reviews indicating that integrated safety features are critical for adoption Tong, L., Zhou, L., & Liu, J., 2022).

Table 14. User Satisfaction and System Adoption Summary

User Group	**Metric**	**Benchmark/ Target**	**PiPoPa Result**	**Reference**
Commuters	Overall App Experience	≥4.0/5.0	4.5/5.0	Fernández, I., & Gutiérrez, J. (2022)
	Ease of Finding Drivers	≥4.0/5.0	4.6/5.0	Fernández, I., & Gutiérrez, J. (2022)
	Fare Transparency	≥4.0/5.0	4.7/5.0	Boulos, M. N. K., Resch, B., Crowley, D. N., Breslin, J. G., Sohn, G.,

				Burtner, R., Pike, W., Jezierski, E., & Chuang, K. S. (2021)
	Safety Features	≥4.0/5.0	4.8/5.0	Tong, L., Zhou, L., & Liu, J. (2022)
	Route Accuracy	≥4.0/5.0	4.4/5.0	Lee, J., & Hickman, M. (2021)
Drivers	Ease of Connecting	≥4.0/5.0	4.6/5.0	Fernández, I., & Gutiérrez, J. (2022)
	Route Guidance Accuracy	≥4.0/5.0	4.5/5.0	Lee, J., & Hickman, M. (2021)
	Earnings Transparency	≥4.0/5.0	4.7/5.0	Boulos, M. N. K., Resch, B., Crowley, D. N., Breslin, J. G., Sohn, G., Burtner, R., Pike, W., Jezierski, E., & Chuang, K. S. (2021)
	Emergency Support Confidence	≥4.0/5.0	4.9/5.0	Tong, L., Zhou, L., & Liu, J. (2022)

F. Summary of Key Objectives Vs. Benchmarks

Table 15 demonstrates that all of PiPoPa's goals were accomplished, and that it met or exceeded the performance, user satisfaction, and safety criteria that were highlighted in recent research Friman, M., & Gärling, T. (2021), Boulos, M. N. K., Resch, B., Crowley, D. N., Breslin, J. G., Sohn, G., Burtner, R., Pike, W., Jezierski, E., & Chuang, K. S. (2021), Fernández, I., & Gutiérrez, J. (2022), Arias-Molinares, D., & García-Palomares, J. C. (2021), Rahman, M. S., Islam, M. Z., & Hossain, M. S. (2022), Lee, J., & Hickman, M. (2021), Liu, L., & Li, Z. (2022), Wang, D., Zhang, D., & Li, D. (2021), Wang, J., & Chen, F. (2022), Xu, Y., & Li, Z. (2022), Tong, L., Zhou, L., & Liu, J. (2022). The fact that this is the case illustrates that integrated, locally tailored, and technology-enabled public transit systems are beneficial in enhancing efficiency, transparency, and passenger safety.

Table 15. Key Objectives Vs. Benchmarks Summary

Objective	Benchmark/ Target	PiPoPa Result	Status	Reference
Radar Search Accuracy	>90%	95%	Exceeded	Friman, M., & Gärling, T. (2021), Arias-Molinares, D., & García-Palomares, J. C. (2021)
Fare Computation Accuracy	100%	100%	Met	Rahman, M. S., Islam, M. Z., & Hossain, M. S. (2022)
Route Estimation Accuracy	≥80%	88%	Exceeded	Wang, D., Zhang, D., & Li, D. (2021)

Emergency Notification Speed	<10 s	3-5 s	Exceeded	Liu, L., & Li, Z. (2022)
User Satisfaction (Commuter)	≥4.0/5.0	4.5-4.8/5.0	Exceeded	Fernández, I., & Gutiérrez, J. (2022), Boulos, M. N. K., Resch, B., Crowley, D. N., Breslin, J. G., Sohn, G., Burtner, R., Pike, W., Jezierski, E., & Chuang, K. S. (2021), Tong, L., Zhou, L., & Liu, J. (2022)
User Satisfaction (Driver)	≥4.0/5.0	4.5-4.9/5.0	Exceeded	Fernández, I., & Gutiérrez, J. (2022), Boulos, M. N. K., Resch, B., Crowley, D. N., Breslin, J. G., Sohn, G., Burtner, R., Pike, W., Jezierski, E., & Chuang, K. S. (2021), Tong, L., Zhou, L., & Liu, J. (2022)

6. Conclusion

Pin Point Path is a smartphone application that was built, deployed, and assessed with the purpose of enhancing public transportation in Oroquieta City by addressing significant issues: passenger safety, fare transparency, route optimization, and emergency response integration. PiPoPa was designed with the objective of improving public transportation in Oroquieta City. PiPoPa was intended to be constructed, put into practice, and evaluated as part of this project. This is a list of the most important goals, which are as follows:

- Develop and test a smartphone application for public transportation that incorporates GPS tracking, radar search, automatic fare calculation, and emergency response protocols.
- Implement a radar search system capable of matching drivers and passengers within a 200-meter range.
- Create dynamic algorithms for fare computation and optimal route selection.
- Integrate emergency response protocols with the City Disaster Risk Reduction and Management Office (CDRRMO).
- Evaluate the application's impact on the efficiency, safety, transparency, and user satisfaction in public transportation services.

Each and every objective was successfully completed with flying colors. It has been shown that the functionality of PiPoPa is resilient over a broad variety of Android devices and networking scenarios. It was possible for the radar search system to achieve an accuracy rate of 95% when it came to identifying available drivers within a 200-meter radius. As a consequence, the average wait times for passengers were significantly reduced by fifty percent. As a consequence of the technique of calculating fares, disputes were eliminated, and accuracy was guaranteed to be one hundred percent. This led to a reduction of ninetyone percent in the number of fare conflicts. The feature that optimized the route offered navigation that was both swift and reliable, which resulted in an improvement of 23% in the estimate of arrival time. This was in comparison to techniques that were static. The inbuilt SOS feature allowed users to inform CDRRMO in the case of an emergency. The notification delivery time was between three and five seconds, and the degree of locational accuracy was very high. Because of this, the users' impression of their own safety as well as their real safety increased as a consequence of.

There was a high degree of satisfaction with all of the major features, notably the fare transparency and the safety adjustments, according to the findings of surveys that were carried out with commuters as well as drivers. These findings were in conformity with the findings of the surveys. As a result of exceeding targets for system dependability, usability, and performance metrics, PiPoPa was shown to be both practical and effective in the context of the local environment. This was accomplished within the framework of the local environment. Having shown that the remedy was effective was the means by which this was achieved. However, there were a few limitations that were observed, such as the reliance on a cell connection and the difficulties that were encountered when moving to locations that had a lower concentration of drivers or infrastructure that was less established. These were some of the limitations that were observed.

Recommendations

Based on the outcomes of this study, the following recommendations are proposed:

- Pin Point Path should be distributed to other locations that have challenges with public transportation that are equivalent to those experienced in the current region. The properties of the system, such as the search radius, should be modified as necessary in line with the particulars of the local environment. Continually work to enhance the algorithms by taking into consideration the data from operations as well as the feedback from users.
- Offline or low-data modes should be created for important activities in order to improve usability in areas with inadequate connectivity. This is a recommendation that has been made. An example of one of these features is an SMS backup. Other features include fare predictions and emergency notifications.
- It is essential to give training and help to users who are less tech-savvy, particularly the elderly and those with low digital literacy, in order to ensure that everyone has equal access to and adoption of the technology. This is especially crucial for ensuring that everyone has equal access to the technology.
- In order to widen the scope of the SOS function, which will result in the app being more flexible for disaster response and urban safety, it is important to encourage ties with other emergency and community safety groups, such as those listed above.
- It is advised that longitudinal studies be carried out in order to analyze the length of time that adoption has been going on, changes in behavior, and the bigger repercussions on transportation efficiency and user safety over the course of time. It is proposed that a comparative study be conducted in a range of urban and rural settings. This is an extra proposal.
- It is necessary for the different local governments to collaborate in order to develop standards for the transparency of fares, the integration of safety features, and the protection of personal information in public transit systems that are enabled by technology.

In conclusion, Pin Point Path is a concept that is not only successfully implemented but also viable for the purpose of using mobile technology to enhance public transportation. When it comes to the general public, it exhibits significant benefits in terms of efficiency, justice, and user safety. It is feasible that such technologies, if properly enhanced and applied on a wider scale, have the potential to revolutionize urban transportation under scenarios that are akin to those described above.

7. Acknowledgement

Throughout the course of this research project, the authors would like to take this opportunity to express their most heartfelt gratitude to the City Disaster Risk Reduction and Management Office (CDRRMO) of Oroquieta City for providing them with direction, assistance, and the opportunity to work together with individuals associated with other departments.nFor their efforts, we would like to extend our gratitude to all of the individuals who took part in the study project and offered insightful input. These individuals include those who work, those who travel, and those who drive cabs. They are all included in this category.nThe authors would also like to take this opportunity to express their appreciation to the University of Science and Technology of Southern Philippines – Oroquieta and Northwestern Mindanao State College of Science and Technology for providing the necessary resources and institutional support that enabled them to carry out this study in an effective manner. The accomplishment of this scientific effort would not have been possible without the contributions of both of these universities. As a result of the steadfast support and understanding that they offered during the whole of the research and development process, the authors would like to use this opportunity to express their gratitude to their families and colleagues.

8. References

1. World Bank. (2020). Transforming transportation for sustainable cities. World Bank. https://live.worldbank.org/en/series/transforming-transportation
2. Alotaibi, Y., et al. (2022). Smart mobility applications: The role of mobile technology in urban transportation. International Journal of Transportation Science and Technology.
3. Olayode, I. O., Severino, A., Alex, F. J., & Macioszek, E. (2023). Systematic review on the evaluation of the effects of ride-hailing services on public road transportation. Transportation Research Interdisciplinary Perspectives, 22, 100943. https://doi.org/10.1016/j.trip.2023.100943
4. Espineli, J. B. F., Facundo, C. C., & Antolin, D. (2025). Is modernization of public transport in the Philippines a boon or a bane for the Filipino people? KINFORMS, 20(1), 4–26. https://doi.org/10.55819/mrj.2025.20.1.4
5. Alotaibi, A., Almasoudi, A., & Alqurashi, A. (2025). Investigation of the impact of smart mobility solutions on urban transportation efficiency in Saudi Arabian cities. Journal of Umm Al-Qura University for Engineering and Architecture, 16(6). https://doi.org/10.1007/s43995-025-00114-3
6. Melo, S., Gomes, R., Abbasi, R., & Arantes, A. (2024). Demand-responsive transport for urban mobility: Integrating mobile data analytics to enhance public transportation systems. Sustainability, 16(11), 4367. https://doi.org/10.3390/su16114367
7. Rivera, R. M. F., Amorim, M., & Gonçalves dos Reis, J. C. (2021). Public transport systems and its impact on sustainable smart cities: A systematic review. In Industrial Engineering and Operations Management (pp. 33–47). https://doi.org/10.1007/978-3-030-78570-3_3

8. Oktorini, R., & Barus, L. S. (2022). Integration of public transportation in smart transportation system (smart transportation system) in Jakarta. Konfrontasi Jurnal Kultural Ekonomi dan Perubahan Sosial, 9(2), 341–347. https://doi.org/10.33258/konfrontasi2.v9i2.223
9. Ramadan, E., Abdalla, S., & Elshaiekh, N. E. (2025). Evaluating public insight on smartphone applications for sustainable urban mobility in the GCC States. Transportation Research Interdisciplinary Perspectives, 30, 101377.
10. Pu, F., Li, Z., Wu, Y., Ma, C., & Zhao, R. (2025). Recent advances in disaster emergency response planning: Integrating optimization, machine learning, and simulation. Safety Emergency Science, 1(1), 9590007. https://doi.org/10.26599/SES.2025.9590007
11. Abante, C. J., Balerite, R. B., Cariño, A. C., & Fong, A. D. (2022). Alternative land elevation assessment through mobile GPS-based rover for small-scale rice farm field preparation. Acta Manilana, 70, 80–92. https://doi.org/10.53603/actamanil.70.2022.udhq394
12. Javeed, M. A., & Li, D. (2025). Urban mobility privacy protection in Internet of Vehicles' GPS trajectory data using stay point mining method. School of Transportation, Southeast University.Li, X., & Wang, M. (2021). Inclusivity in public transport apps: Design challenges. Universal Access in the Information Society.
13. Urban Mobility India. (2024). Deployment of digital payments in public transport.
14. Suárez, S., Macedo, E., Ciccarelli, G., & Bandeira, J. M. (2025). User feedback assessment of region-focused mobility-as-a-service bundles. Multimodal Transportation, 4, 100204.
15. Khaled, H., Tarek, K., Mohamed, W., & Hussein, M. (2023). Intelligent transportation system real-time tracking. Qeios. https://doi.org/10.32388/2VSPI8
16. Rivera, J. N. D. (2021). An offline push notification for mobile device. Globus: An International Journal of Management & IT, 10.46360/globus.mgt.120211001
17. Cornet, H., Stadler, S., Kong, P., Marinkovic, G., Frenkler, F., & Sathikh, P. M. (2019). User-centred design of autonomous mobility for public transportation in Singapore. Transportation Research Procedia, 41, 191–203. https://doi.org/10.1016/j.trpro.2019.09.038
18. Chen, L., Wang, Y., Qi, G., & Lv, H. (2023). Research on smart mobility in public transportation and solutions for mobility. Highlights in Science Engineering and Technology, 56, 498–505. https://doi.org/10.54097/hset.v56i.10719
19. Rahman, R. T., Bahar, A., Prananta, A., & Afifudin, M. (2024). Impact of service quality, pricing strategies, user trust, and technological innovation in Indonesia's online transportation services industry. International Journal of Business Law and Education, 5(2), 2549–2561. https://doi.org/10.56442/ijble.v5i2.901
20. Friman, M., & Gärling, T. (2021). Passenger expectations and experiences of public transport: A review of the literature. Transportation Research Part F: Traffic Psychology and Behaviour, 81, 357–368. https://doi.org/10.1016/j.trf.2021.06.009
21. Boulos, M. N. K., Resch, B., Crowley, D. N., Breslin, J. G., Sohn, G., Burtner, R., Pike, W., Jezierski, E., & Chuang, K. S. (2021). Crowdsourcing, citizen sensing and sensor web technologies for public and environmental health surveillance and crisis management: Trends, OGC standards and application examples. International Journal of Health Geographics, 20(1), 1–24. https://doi.org/10.1186/s12942-021-00250-5

22. Fernández, I., & Gutiérrez, J. (2022). Mobile applications for public transportation: A systematic review of user experience and adoption factors. Transportation Research Part F: Traffic Psychology and Behaviour, 85, 243–258. https://doi.org/10.1016/j.trf.2022.01.007
23. Arias-Molinares, D., & García-Palomares, J. C. (2021). Exploring the spatial patterns of car-sharing users: Evidence from Madrid and Barcelona. Sustainability, 13(21), 12195. https://doi.org/10.3390/su132112195
24. Rahman, M. S., Islam, M. Z., & Hossain, M. S. (2022). Mobile app-based public transportation systems in developing countries: Challenges and opportunities. Transportation Research Interdisciplinary Perspectives, 14, 100625. https://doi.org/10.1016/j.trip.2022.100625
25. Chen, L., Wang, J., & Chen, F. (2022). Real-time traffic prediction using deep learning with multi-source data fusion. Transportation Research Part C: Emerging Technologies, 134, 103485. https://doi.org/10.1016/j.trc.2021.103485
26. Lee, J., & Hickman, M. (2021). Real-time transit information systems: Benefits, challenges, and opportunities. Transportation Research Part A: Policy and Practice, 143, 1–15. https://doi.org/10.1016/j.tra.2020.11.005
27. Liu, L., & Li, Z. (2022). Mobile app-based emergency response systems: A review of design principles and implementation challenges. International Journal of Disaster Risk Reduction, 75, 102987. https://doi.org/10.1016/j.ijdrr.2022.102987
28. Sultana, S., & Bullock, D. M. (2021). Mobile application-based traffic data collection: A review of methodologies and applications. Transportation Research Record, 2675(5), 45–58. https://doi.org/10.1177/0361198120978512
29. Wang, D., Zhang, D., & Li, D. (2021). Mobile app-based transportation systems: A review of technical architecture and implementation challenges. IEEE Transactions on Intelligent Transportation Systems, 22(7), 4123–4136. https://doi.org/10.1109/TITS.2021.3052789
30. Wang, J., & Chen, F. (2022). Real-time traffic prediction using deep learning: A comprehensive review. Transportation Research Part C: Emerging Technologies, 136, 103568. https://doi.org/10.1016/j.trc.2022.103568
31. Xu, Y., & Li, Z. (2022). Mobile app-based emergency response systems for public transportation: A review of design principles. International Journal of Disaster Risk Reduction, 71, 102789. https://doi.org/10.1016/j.ijdrr.2022.102789
32. Tong, L., Zhou, L., & Liu, J. (2022). Mobile app-based transportation systems: A review of user adoption and satisfaction. Transportation Research Part F: Traffic Psychology and Behaviour, 86, 578–592. https://doi.org/10.1016/j.trf.2022.03.012