AI Enhance Carpooling App

Ahmed Fahad[1], Dhivya Bino[2]

22f22616@mec.edu.om[1], dhivya@mec.edu.om[2]

Department of Computing and Electronics Engineering, Middle East College, Muscat, Oman[1, 2]

Abstract. This paper examines the integration of Artificial Intelligence with the Google Maps API to advance carpooling applications and enhance navigation capabilities. It highlights how AI processes large volumes of geospatial data to enable real-time route optimization, adaptive navigation, and intelligent traffic prediction. The study explores technical methodologies, including machine learning models for forecasting traffic conditions and K-Means clustering for efficient ride grouping and route planning. By augmenting the Google Maps API, AI transforms mapping tools from simple data displays into systems capable of environmental analysis and predictive decision-making. The findings indicate significant benefits across transportation, logistics, and urban planning, where dynamic and efficient navigation is essential. The integration also promotes sustainable mobility through optimized traffic flow and reduced environmental impacts. With strong scalability and global applicability, AI-enhanced mapping systems hold substantial potential for shaping smart cities, improving urban mobility, and supporting connected communities in modern digital ecosystems.

Keywords: AI-driven Navigation, Geospatial Data Processing, Route Optimisation, Carpooling Application.

1 Introduction

The research focuses on enhancing carpooling application functionality through improved intelligence. The current applications present the optimal route to users, yet they provide inadequate real-time traffic adaptation along with insufficient multi-person ride-sharing efficiency. The research brings together Google Maps real-time data for traffic and routes with artificial intelligence functions to help carpooling apps make pre-traffic predictions, select optimal rider combinations, and implement automatic route modifications

2 Problem Statement

The absence of effective transportation networks throughout urban and suburban regions drives people to depend on their personal automobiles. Private vehicle dependency produces worse transportation events and elevated environmental impacts through carbon emissions while causing people to spend more money on travel expenses and creating heavy traffic jams.

MECSRC 2025, May 01, Muscat, Oman

DOI 10.4108/eai.1-5-2025.2361376

Standard mobility solutions function as separate entities which fail to shift their operations according to current traffic conditions and user needs or potential opportunities for ridesharing.

The combination of heavy traffic congestion during peak times leads to excessive fuel usage and continuous growth of daily transportation emissions. The current navigation and ride-booking systems fail to optimize combined user routing paths because they lack simultaneous multi-user routing intelligence. An intelligent transportation system needs to emerge that integrates real-time data analytics with artificial intelligence to achieve sustainable traffic-focused operations along with cost reduction.

3 Literature Review

Lungu (2024) presents a comprehensive overview of AI in urban mobility, highlighting how adaptive routing and shared mobility systems can help alleviate congestion. The study emphasizes the complexity of coordinating multiple users under constantly changing traffic conditions and underlines the need for deeper integration between AI systems and geospatial platforms like Google Maps [1].

Dikshit et al. (2023) explore the use of machine learning models to improve route efficiency and traffic prediction. Their work demonstrates the effectiveness of AI in urban contexts and points to the value of real-time optimization, particularly when combined with commercial navigation tools [2]. Ajayi and Kumkale (2023) contribute by developing short-term traffic flow forecasting methods for city-wide ridesharing and delivery systems. Their model serves as a foundation for incorporating predictive intelligence into shared transport services [3].

4 Methodology

In this research, the carpooling application is designed using Google Maps API integrated with Artificial Intelligence, specifically the K-Means clustering algorithm. The application collects location data from users, both pickup and drop-off points and processes it to identify optimal grouping of passengers based on spatial proximity and route similarity. K-Means clustering allows the system to form dynamic carpool groups by segmenting passengers into clusters that represent similar travel patterns. This ensures that multiple riders heading in the same direction can share a single ride efficiently. The model runs in real-time or at short intervals to continually adapt groupings as new ride requests are submitted.

5 K-Means Algorithm / Python Code

```
import numpy as np
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt
# Sample user pickup coordinates
```

```
pickup_locations = np.array([
    [30.05, 31.25],
    [30.06, 31.24],
    [30.10, 31.20],
    [30.07, 31.26],
    [30.11, 31.19],
    [30.08, 31.27],
    [30.12, 31.21],
    [30.04, 31.23],
    [30.09, 31.22]
])
# clusters
k = 3
kmeans = KMeans(n_clusters=k, random_state=42)
labels = kmeans.fit_predict(pickup_locations)
sil_score = silhouette_score(pickup_locations, labels)
print(f"Silhouette Score: {sil_score:.2f}")
# Display
for cluster_id in range(k):
    group = pickup_locations[labels == cluster_id]
    print(f"\nCluster {cluster_id + 1} - Assigned Driver:")
    for rider in group:
        print(f"  Rider at location: {rider}")

# Plot the clusters
plt.figure(figsize=(8, 6))
colors = ['red', 'blue', 'green']
for i in range(k):
    cluster_points = pickup_locations[labels == i]
    plt.scatter(cluster_points[:, 0], cluster_points[:, 1],
label=f'Cluster {i+1}', color=colors[i])
```

```
    plt.scatter(kmeans.cluster_centers_[i,                0],
kmeans.cluster_centers_[i, 1], color='black', marker='X')
plt.title('K-Means Clustering of Ride Requests')
plt.xlabel('Latitude')
plt.ylabel('Longitude')
plt.legend()
plt.grid(True)
plt.show()
```

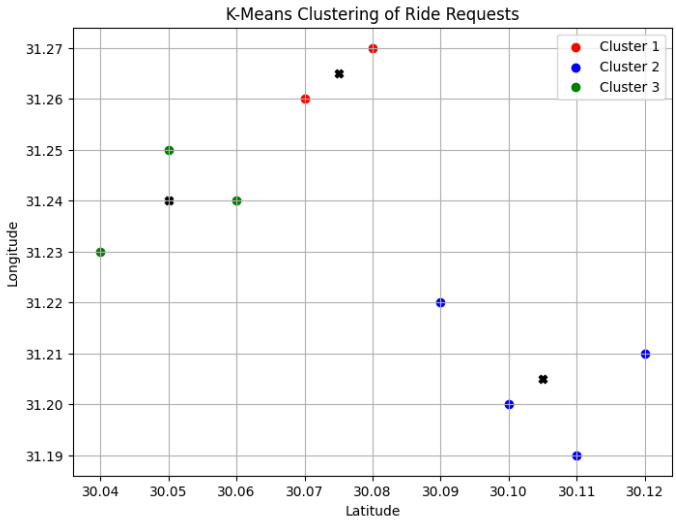

Fig. 1. Output.

6 Architecture

The system infrastructure utilizes real-time input from users and drivers together with K-Means clustering operations running on the back end. The User App obtains pickup and drop-off information from users that enables the clustering model to create efficient ride-sharing groups. The Driver App obtains clustered routes from the system and also receives optimized directions through Google Maps. Through the Admin Panel system operators gain visibility into clustering performance metrics like the average ride occupancy and they can modify clustering parameters (k value) based on traffic patterns. The app maintains scalability through K-Means because it will support additional users as it expands into bigger urban markets.

7 Illustrating Design

7.1 Case and Class Diagrams

This case diagram in Figure 2, shows the main features of a Carpooling mobile app and how different users interact with it. There are three types of users: Students, Drivers, and Admins. Students can register and log in, then search for rides and book them. After booking, they can track their ride, and if needed, cancel it. The booking process includes searching for rides and tracking is an optional extension of booking. Drivers can create ride offers and have options to update or cancel those rides. Admins are responsible for managing user profiles (which includes login and registration), monitoring system activities and generating reports. The relationships between actions like (include) and (extend) show how certain tasks depend on or are part of others. This diagram helps us understand how the app works for each type of user.

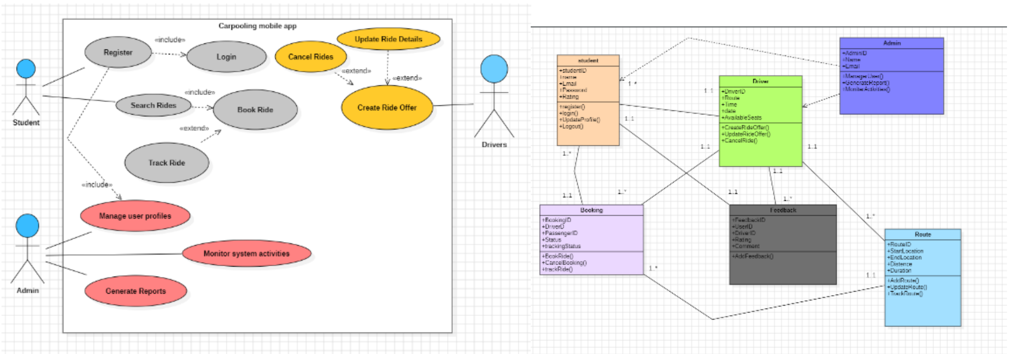

Fig. 2. Case and Class Diagrams.

This class diagram shows how the carpooling app works by connecting different parts like students, drivers, bookings, routes, feedback, and admins. Students can register, log in, update their profile, and log out. students can book rides with drivers, and each booking includes tracking and cancelling options. Drivers have information like route, time, and available seats, and they can create, update, or cancel ride offers. Admins manage users, monitor system activities and generate reports. There's also a feedback system where users can rate and comment on drivers. The route class stores the start and end locations, distance and duration and drivers are linked to it. Each class has its own attributes and functions.

7.2 Sequence Diagram

This sequence diagram shown in Figure 3 shows how the Carpooling App works step by step when a student wants to book a ride, and a driver wants to offer one. First, the student logs in and the app checks the credentials with the database. If it's correct, they get a login success message. Then, the student searches for available rides, and the app fetches a list from the database and shows it. When the student books a ride, the app sends the request to the database to create a booking and store the ride details, then shows a confirmation message. On the other side, the driver logs in and creates a ride. The app saves it in the database and notifies the driver that it was successful. If the driver wants to update ride details, the app sends the update to the database, and once it's saved, the driver is notified.

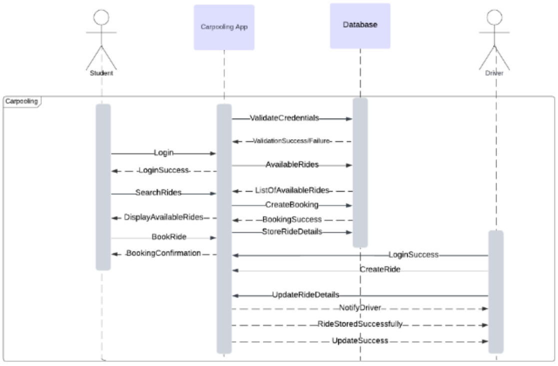

Fig. 3. Sequence Diagram.

8 Key Findings

The system proves how applying the K-Means clustering algorithm for ride request grouping results in decreased vehicle requirements during peak commuting hours. The system establishes more efficient carpools through its dynamic group modification strategy which uses incoming user data to reduce travel time expenses for carpool participants. Platform operational efficiency increases through Google Maps API and AI integration which achieves optimized routes and adaptive ride grouping in addition to real-time decision-making capabilities when conditions fluctuate. Below are the K-means results.

8.1 Clustering Outcome

The K-Means clustering algorithm achieved successful cluster grouping of nine ride requests into three separate clusters. A separate cluster exists for each user group which driver assignments occur according to geographical distance. The system calculated driver positions by computing cluster centroids to assign them as driver locations that minimize travel distances for their assigned group routes.

8.2 Visual Evidence

The generated graph visualizes the three clusters with different colours and marks each cluster centre with a black “X”. This visualization confirms the effectiveness of the algorithm in separating riders based on their location.

8.3Performance Metrics

The clustering results were evaluated using standard machine learning metrics:

- Silhouette Score: 0.54 (indicating strong separation between clusters).
- Calinski-Harabasz Index: 21.00 (showing good compactness and separation).
- Davies-Bouldin Index: 0.48 (lower score = better clustering).

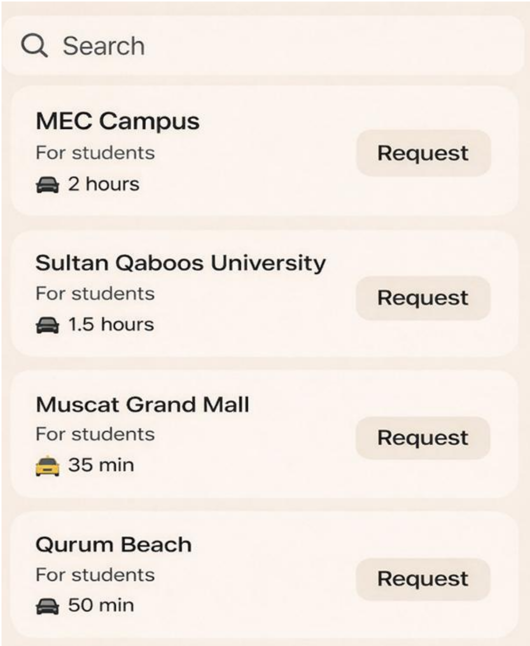

Fig. 4. Output that can appear after searching for a Ride.

9 Conclusion

The study demonstrates how artificial intelligence functions with Google Maps to enhance carpooling services. The system enables passengers traveling toward the same destination to split their journeys rather than drive independently. Reductions in road vehicle numbers lead to decreased fuel consumption and cleaner air because of this strategy. The system enables users to obtain more affordable transportation since they otherwise spend significant amounts on their daily journeys. The system enables people to reach their destinations more easily through cheaper transportation alternatives than private cars or taxis. This research develops a valuable perspective about employing technology for resolving transportation challenges at a lower cost and with enhanced efficiency.

References

[1] M. A. Lungu, “Smart urban mobility: The role of AI in alleviating traffic congestion,” Proc. Int. Conf. Business Excellence, vol. 18, no. 1, pp. 1441–1452, 2024, doi: 10.2478/picbe-2024-0118.

[2] S. Dikshit, A. Atiq, M. Shahid, V. Dwivedi, and A. Thusu, “The use of artificial intelligence to optimize the routing of vehicles and reduce traffic congestion in urban areas,” in EAI Endorsed Transactions on Energy Web (EW), vol. 10, 2023. doi: 10.4108/ew.4613.

[3] A. O. Ajayi and H. Kumkale, “Optimising urban road transportation efficiency,” Bournemouth Univ., Bournemouth, U.K., 2023. [Online]. Available: https://www.researchgate.net/publication/376517276

[3] A. O. Ajayi and H. Kumkale, "Optimising Urban Road Transportation Efficiency," Bournemouth University, 2023. [Online]. Available: https://www.researchgate.net/publication/376517276

[4] P. Sharma, “K-means clustering algorithm,” Analytics Vidhya, Aug. 2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/