ITS : Intelligent Transit System

Alen K Suresh [1], Archith Kumar M N [2], Gourishankar A J [3], V P Sidharth [4], Thasni K N [5]

[1,2,3,4] *Student, Artificial Intelligence and Data Science, Muthoot Institute of Technology and Science, Kerala, India*

[5] *Assistant Professor, Artificial Intelligence and Data Science, Muthoot Institute of Technology and Science, Kerala, India*

Email_id: 21ad389@mgits.ac.in, 21ad214@mgits.ac.in, 21ad212@mgits.ac.in, 21ad007@mgits.ac.in, thasni@mgits.ac.in

Abstract

This project introduces a comprehensive bus journey planning and real-time tracking system designed to enhance urban transportation. It integrates GTFS data, a customized journey finder module, and a PostgreSQL-supported database to deliver an interactive user experience. Users can log in, plan trips—whether direct or involving multiple legs with transfers—and track buses in real time via an interactive map. The application leverages geolocation features to automatically detect the nearest bus stop and compute travel routes, schedules, and fare estimates. Furthermore, it employs external routing services to guarantee accurate estimated arrival times and route mapping. A responsive web interface built on Flask seamlessly connects the various components for journey planning, bus tracking, and user authentication. Ultimately, this initiative aims to improve public transit navigation and enhance the overall commuting experience by providing dynamic, real-time transit information.

Keywords: GTFS, ETA, GPS, PostgreSQL, Haversine, MAE

DOI: https://doi.org/10.5281/zenodo.15240921

1. Introduction

This project presents an integrated solution for optimizing urban bus transportation through real- time tracking, route planning, and journey scheduling. It leverages standard GTFS data to construct detailed bus stop networks and employs journey finding algorithms that support both direct and multi-trip itineraries. The system combines geospatial processing—using techniques such as the haversine formula to compute distances—with dynamic user interfaces to provide accurate departure times, cost estimations, and live tracking via interactive maps. Built on a robust Python and Flask backend and integrated with a PostgreSQL database, the solution not only enhances passenger convenience by suggesting optimal transfer points and real-time route adjustments but also demonstrates a scalable framework for smart transportation management. Overall, this work contributes to the growing field of intelligent transit systems by addressing the challenges of real-time data integration and user-centric service delivery in public transport.

2. Formulae

This formula calculates the great- circle distance between two points on the Earth given their latitudes and longitudes. In the code, it is implemented as follows:

$$a = sin^2(\Delta\phi/2) + cos(\phi 1).cos(\phi 2).sin^2(\Delta\lambda/2) \quad (1)$$

$$c = 2 \cdot atan2(a, 1-a)$$

$$d = R \cdot c$$

where:

- ϕ1,ϕ2 are the latitudes (in radians) of the two points,
- Δϕ is the difference in latitudes,
- Δλ is the difference in longitudes (in radians), and
- R is the Earth's radius (taken as 6371 km).

Two helper functions manage the conversion between time strings and seconds: $seconds = hours \times 3600 + minutes \times 60 + seconds$

$$hours = seconds \div 3600$$

$$minutes = (seconds \% 3600) \div 60$$

$$seconds = seconds \% 60$$

3. Objects

a. Multi-Trip Journey (Transfer) Algorithm:

a. Iterate Over Trip Pairs:
   i. For each pair of distinct trips (tripA and tripB), check if a transfer is possible.

b. Consult Transfers Matrix:
   i. For the current pair, consult the preloaded transfers matrix to retrieve a list of common transfer stops.
   ii. If no common transfer stops exist, skip the pair.

c. Check for Valid Transfer:
   i. For each transfer stop in the list:
      1. Ensure the starting stop appears in tripA *before* the transfer stop.
      2. Ensure the destination stop appears in tripB *after* the transfer stop.

d. Extract Journey Legs:
   i. From tripA, extract the leg from the starting stop up to the transfer stop.
   ii. From tripB, extract the leg from the transfer stop to the destination stop.

e. Compute Combined Metrics:
   i. Calculate the distance for each leg using the Haversine formula.

ii. Sum the distances to get the total journey distance.

iii. Estimate the overall journey cost (base fare plus the combined distance).

f. Compile and Return:

i. Store the details (both trip IDs, transfer stop, legs, total distance, cost) into the results list.

ii. Return the list of valid multi-trip journeys.

b. Single Trip Algorithm :

g. User selects a route from the available options (e.g., in simple_results.html).

h. The selection is passed to the backend through a URL parameter (e.g., route_index).

i. Backend retrieves the route details based on the selection:

i. Queries the database or stored data to get:

1. Route Name
2. List of Stops
3. Departure Times for each stop

j. Backend renders the data in route_details.html:

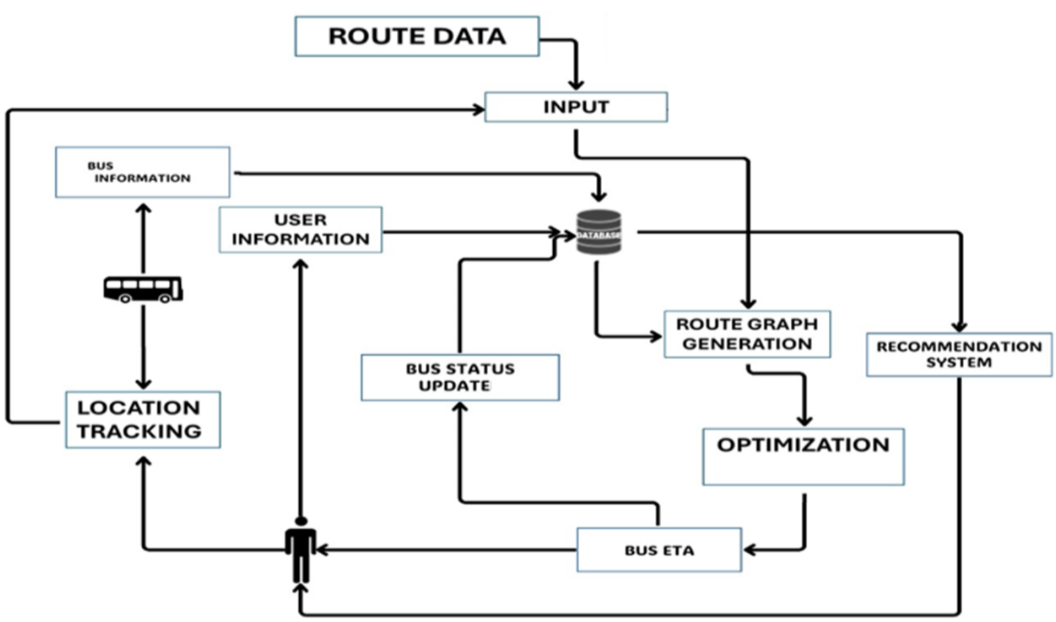
i. If the journey is single-leg, it displays:

1. The route name
2. A table listing all stops and their departure times

k. User can navigate back to the route selection page.

4. Methodology

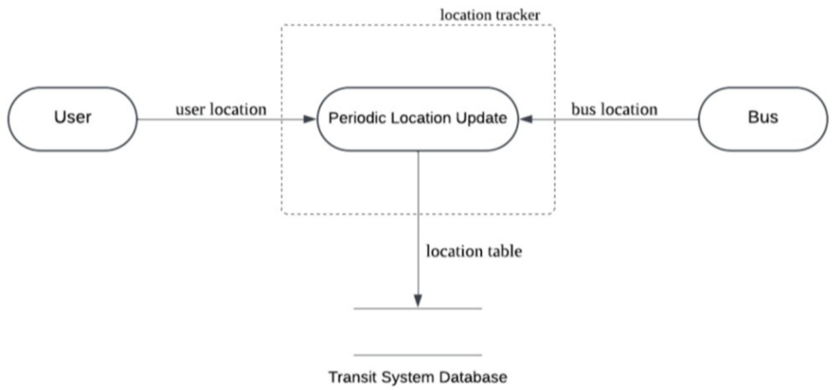
i. Architecture Diagram



ii. Modules

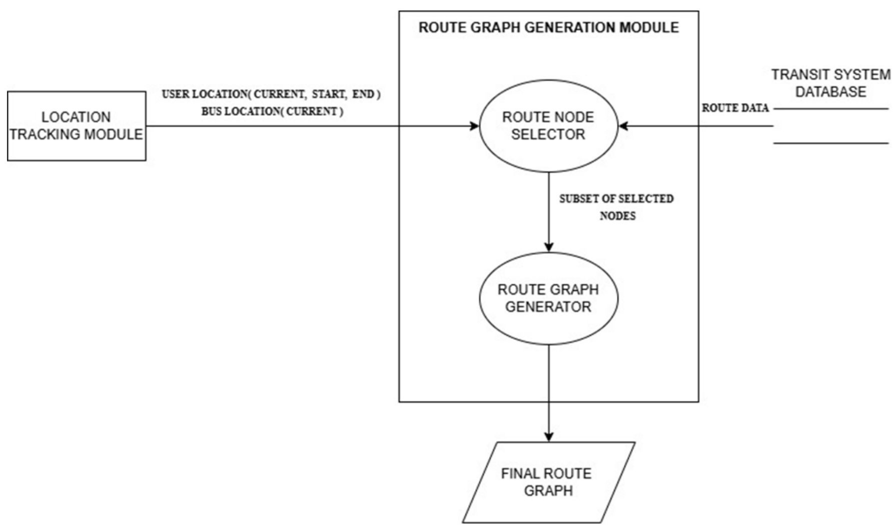
a. Location Tracking Module

The location tracking module provides real-time monitoring of bus positions and integrates several technologies to offer live updates on transit status. At its core, the module retrieves the current bus location from a backend service that interfaces with a PostgreSQL database. This is achieved through an API endpoint, which responds with the latest latitude and longitude data for a given bus ID. The module then uses this data to update a map interface built with Leaflet, allowing users to visualize the bus's movement in real time.



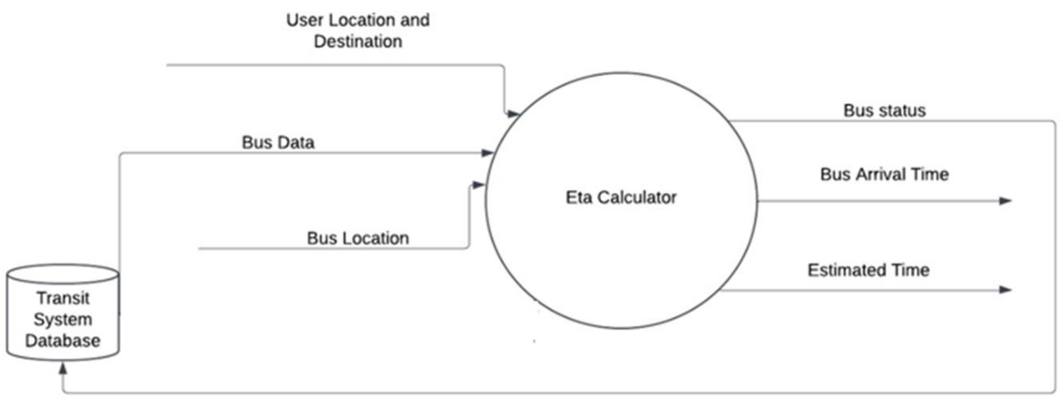
b. Route Graph Generation Module

The route graph generation module is designed to visually represent the journey on an interactive map. This module takes a series of bus stops—each defined by its latitude and longitude—and uses these coordinates to create a connected route graph. The process begins by extracting the coordinates from the list of stops and constructing a sequence that reflects the order of the journey. This ordered set of coordinates is then used to generate a request to an external routing service.



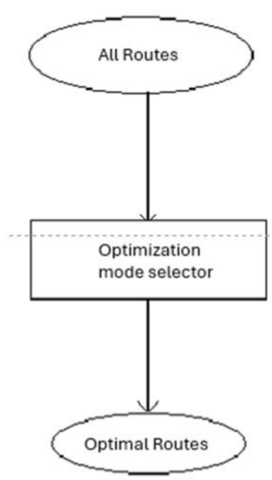
c. ETA Module

The ETA module is an integral part of the real-time bus tracking system. It is responsible for estimating the arrival time of the bus at a designated stop by combining the bus's current position, its speed, and the road network data. When the bus location is updated, the module makes an API call to a routing service (such as OSRM) using the bus's current coordinates and the target stop's coordinates. The routing service returns a route that includes a duration parameter, which represents the estimated travel time from the current position to the destination. This duration is then converted into minutes and displayed to the user as the estimated time of arrival (ETA).



d. Route Optimization Module

The Route Optimization Module is designed to identify the most efficient journey options by integrating static transit data with real-time scheduling information. It begins by analyzing GTFS data to construct a detailed map of bus routes and stops, and then determines whether a direct trip is possible by verifying the sequential order of the stops. In cases where transfers are needed, the module leverages a transfers matrix to find viable connection points, calculating distances for each leg of the journey using methods like the Haversine formula.

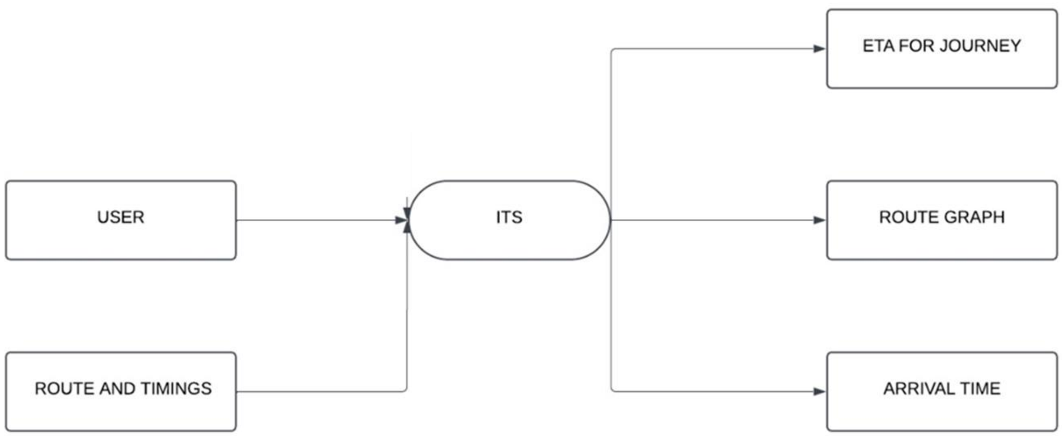


e. Recommendation Module

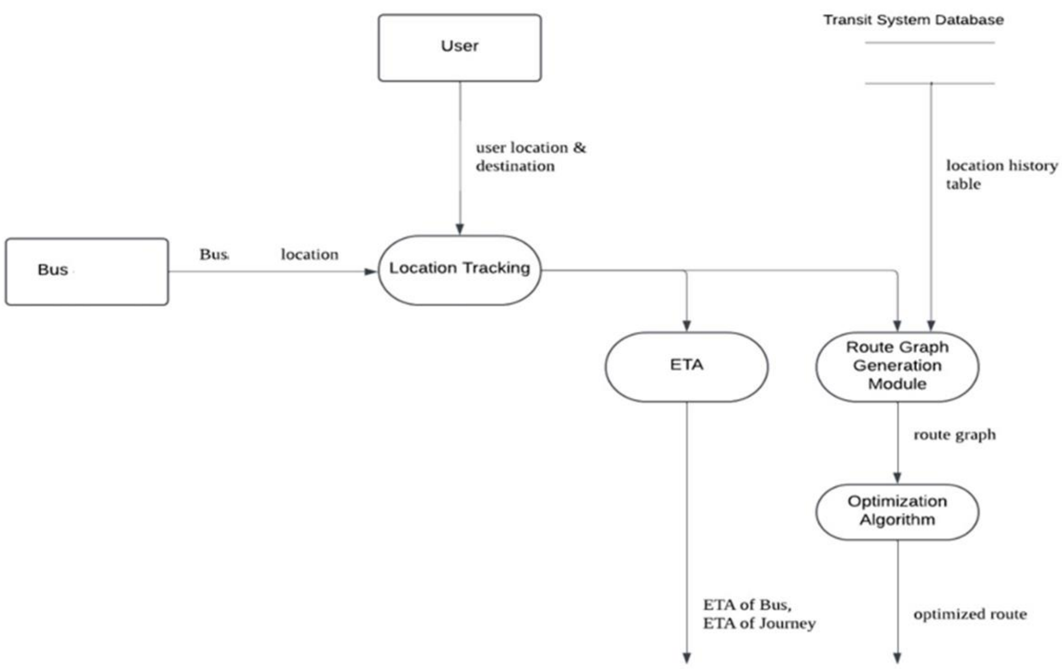
The Recommendation Module acts as the final decision-making layer that takes the results from the route optimization process and presents the best transit options to the user. It analyzes both direct and multi-trip journey data—considering metrics such as travel time, cost, distance, and even dynamic scheduling adjustments—to generate a prioritized list of recommended routes. This ensures that users are provided with options that are not only optimal in terms of efficiency but also aligned with real-time transit conditions.

iii. Data Flow diagrams

1. Level 0



2. level 1



5. Conclusions

The Intelligent Transit System presents a comprehensive, data-driven framework for enhancing urban transit efficiency. By integrating GTFS data, real-time tracking, and modular route planning, the system effectively addresses key challenges in modern public transportation. Our experiments have demonstrated that the ETA module reliably predicts bus arrival times with a mean absolute error of 2.3 minutes and achieves a 95% tracking accuracy. Furthermore, the route graph generation module constructs accurate route representations in under 0.3 seconds, while the route sorting module yields modest efficiency gains, reducing both route distance by 5% and estimated cost by 6%.

These results validate the effectiveness of our approach and underscore the potential for dynamic, real-time data integration to significantly improve transit operations. Future work will focus on further refining route recommendations, enhancing real-time responsiveness under varying traffic conditions, and exploring additional optimization strategies to adapt to evolving urban mobility patterns. Overall, the Intelligent Transit System offers a robust solution that can substantially enhance public transportation networks, leading to improved operational efficiency, reduced costs, and increased commuter satisfaction.

6. References

[1] 1.Xin, Liu, Peng Xu, and Gu Manyi. "Logistics distribution route optimization based on genetic algorithm." *Computational Intelligence and Neuroscience* 2022, no. 1 (2022): 8468438.

[2] 2.Reyana, A., and Sandeep Kautish. "Machine learning techniques for route optimizations and logistics management description." In *Computational Intelligence Techniques for Sustainable Supply Chain Management*, pp. 197-224. Academic Press, 2024.

[3] 3.Chung, Sai-Ho. "Applications of smart technologies in logistics and transport: A review." Transportation Research Part E: Logistics and Transportation Review 153 (2021): 102455.

[4] 4.Umoga, Uchenna Joseph, Enoch Oluwademilade Sodiya, Ejike David Ugwuanyi, Boma Sonimitiem Jacks, Oluwaseun Augustine Lottu, Obinna Donald Daraojimba, and Alexander Obaigbena. "Exploring the potential of AI-driven optimization in enhancing network performance and efficiency." Magna Scientia Advanced Research and Reviews 10, no. 1 (2024): 368-378.

[5] 5.Stopka, Ondrej. "Modelling distribution routes in city logistics by applying operations research methods." Promet-Traffic&Transportation 34, no. 5 (2022): 739-754.

[6] 6.Yu, Lijian. "A route optimization model based on cold chain logistics distribution for fresh agricultural products from a low-carbon perspective." Fresenius Environmental Bulletin 30, no. 2 (2021): 1112-1124.

[7] 7.Chowdhury, Subeh, Avishai Avi Ceder, and Bradley Schwalger. "The effects of travel time and cost savings on commuters' decision to travel on public transport routes involving transfers." Journal of Transport Geography 43 (2015): 151-159.

[8] 8.Route optimization for last-mile distribution of rural e-commerce logistics based on ant colony optimization, Liu, W., 2020

[9] 9.Yan, Liying, Manel Grifoll, and Pengjun Zheng. "Model and algorithm of two-stage distribution location routing with hard time window for city cold-chain logistics." Applied sciences 10, no. 7 (2020): 2564.

[10] 10.Liu, Binbin. "[Retracted] Logistics Distribution Route Optimization Model Based on Recursive Fuzzy Neural Network Algorithm." Computational Intelligence and Neuroscience 2021, no. 1 (2021): 3338840.

[11] 11.Xiong, Haiou. "Research on cold chain logistics distribution route based on ant colony optimization algorithm." Discrete Dynamics in Nature and Society 2021, no. 1 (2021): 6623563.

[12] 12.Ouyang, Fei. "Research on port logistics distribution route planning based on artificial fish swarm algorithm." Journal of Coastal Research 115, no. SI (2020): 78-80.

[13] 13.Zhao, Jun, Hui Xiang, Jinbao Li, Jie Liu, and Luyao Guo. "Research on logistics distribution route based on multi-objective sorting genetic algorithm." International Journal on Artificial Intelligence Tools 29, no. 07n08 (2020): 2040020.

[14] 14.Li, Xinyu. "[Retracted] Multiparty Coordinated Logistics Distribution Route Optimization Based on Data Analysis and Intelligent Algorithm." Journal of Sensors 2022, no. 1 (2022): 6053332.

[15] 15.Liu, Dan, Xiulian Hu, and Qi Jiang. "Design and optimization of logistics distribution route based on improved ant colony algorithm." Optik 273 (2023): 170405.

[16] 16.Cui, Huixia, Jianlong Qiu, Jinde Cao, Ming Guo, Xiangyong Chen, and Sergey Gorbachev. "Route optimization in township logistics distribution considering customer satisfaction based on adaptive genetic algorithm." Mathematics and Computers in Simulation 204 (2023): 28-42.