SYMBIOSIS INSTITUTE OF TECHNOLOGY,HYDERABAD
Modallaguda (V), Nandigama (M), Rangareddy Dist, Hyderabad, Telangana, India, Pin Code : 509217

MetroConnect: Smart Metro Route Finder

Bachelor of Technology in

Computer Science and Engineering

Submitted by

Vinay Chakravarthy Addanki (PRN: 24070721043) &

Ancha Sreesainadh (PRN: 24070721043)

For the fulfillment

of

Data Structure and Algorithm Course

Symbiosis Institute of Technology, Hyderabad

Survey No. 292, Off Bangalore Highway, Modallaguda Village,

Nandigama Mandal, Ranga Reddy District, Telangana – 509217

CONTENTS

Abstract

Public transportation systems require cost-effective and time-efficient route planning. MetroConnect – Smart Metro Route Finder is a C-based console application developed as part of the Data Structures course. The objective of the system is to assist metro commuters in Hyderabad by identifying the shortest route, calculating the number of stations, determining whether a line interchange is required, and computing the estimated fare dynamically based on peak and non-peak travel timings.

The project uses fundamental Data Structure concepts such as arrays (for station storage), string manipulation (for case-insensitive search), and conditional logic (for interchange and fare calculation). The program is menu-driven, making it user-friendly. It accepts source and destination station names, searches them using linear string matching, and displays the route and fare. A peak hour surcharge is also implemented as per real Hyderabad Metro timings (9:00–10:30 AM and 4:30–6:00 PM).

The completed system demonstrates efficient use of Data Structures to solve real-world routing problems. It is lightweight, fast, and requires minimal system resources, making it ideal for academic and real-time applications.

Introduction

Urban cities face challenges like traffic congestion, pollution, and unpredictable travel time. Public transport systems like Hyderabad Metro help solve these problems — but route planning, station interchanges, and fare calculation are still confusing for new users.

The **goal** of MetroConnect is to simplify navigation for metro commuters by providing them with:

- Shortest route between stations
- Total travel distance in km
- Estimated fare including peak-hour surcharge

The application provides a real-world implementation of Data Structures in a transportation system.

Objectives

- **Accept source and destination station names.**
- **Identify shortest metro route using *Dijkstra's shortest path algorithm*.**
- **Calculate fare based on route distance and time of travel.**
- **Provide a menu-driven system for easy navigation.**

Scope

- **Works for the full Hyderabad Metro network (Red, Blue, Green Line).**
- **Station data stored internally — no internet dependency.**

Limitations

- **Does not show live train timings.**
- **Metro station data is static, requires update if new lines are added.**

Problem Definition

Problem:

To design a metro route computation system that finds the shortest route between any two stations, calculates distance, and computes fare dynamically.

Input Requirements

- Source Station Name
- Destination Station Name
- Travel time (HHMM format)

Output

- Shortest route (station by station path)
- Total distance (in kilometers)
- Fare including peak-hour surcharge (if applicable)

Literature Review / Existing Systems

Various metro navigation systems exist:

Platform	Type	Limitation
Hyderabad Metro Mobile App	Mobile app	Requires internet & mobile installation
Google Maps Transit Mode	Online navigation	Does not compute station-based fare dynamically
Route Maps (Printed at stations)	Manual reference	User must determine route and changes manually

Unlike these systems, **MetroConnect** operates fully offline and is designed to run in a programming environment, demonstrating Data Structure usage and enabling learners to understand how route planning logic works internally.

Methodology / System Design

Data Structures Used:

Data Structure	Purpose
Array of Strings	Store station names line-wise
2D Adjacency Matrix (Graph)	Represent connectivity between stations
Linear Search	To find station index
Dijkstra's Algorithm	To compute shortest path
Recursion	To print route in correct order

System Architecture:

User → Input (Source + Destination + Time)

↓

Station Search (Array + String Compare)

↓

Check if interchange needed

↓

Fare Calculation (Peak hour logic)

↓

Display Output (Route + Fare + Stops)

5. Implementation Details:

Programming Language Used

- C (ANSI standard)

Important Functions in the Program

Function Name	**Purpose**
`findStationIndex()`	Searches and returns station index
`dijkstra()`	Computes shortest path using graph
`printPath()`	Recursively prints computed path
`planTrip()`	Takes user input and performs routing

Fare Structure

Base Fare = Rs. 20

Fare per km = Rs. 5 × distance

Peak Hour Surcharge = Rs. 10

Peak Hours defined:

- Morning: **9:00 AM to 10:30 AM**
- Evening: **4:30 PM to 6:00 PM**

Results and Analysis

The program was tested for:

- Route with interchange (ex: Miyapur → MGBS)
- Route on same line (ex: Madhapur → Hitec City)
- Peak hour vs non-peak hour fare differences

Performance Evaluation:

Parameter	**Result**	**Parameter**	**Result**
Time Complexity	$O(V^2)$ (due to Dijkstra with matrix)	Time Complexity	$O(V^2)$ (due to Dijkstra with matrix)
Space Complexity	$O(V^2)$ (graph stored in adjacency matrix)	Space Complexity	$O(V^2)$ (graph stored in adjacency matrix)
Total Stations Supported	50 (expandable)	Total Stations Supported	50 (expandable)

Functional Validation

Condition	**Verified**
Accepts case-insensitive input	Yes
Detects interchange station	Yes
Fare changes during peak hours	Yes

Conclusion

The *MetroConnect* project successfully demonstrates how Data Structures and Algorithms can be applied to real-world problems.

Key learnings:

- Graphs can be used to model transportation networks.
- Dijkstra finds shortest paths efficiently.
- Arrays & string matching help manage metro station datasets

Future Enhancements

- Add mobile application/GUI
- Real-time train arrival API (IoT integration)
- Add QR ticket generation module

References

1. Hyderabad Metro Rail Official Website — https://www.ltmetro.com/
2. Reema Thareja, *Data Structures Using C*, Oxford University Press
3. E. Balagurusamy, *Programming in ANSI C*, McGraw Hill
4. GeeksforGeeks, "Arrays and Searching Algorithms" — https://www.geeksforgeeks.org/