Advancing Urban Transportation Efficiency: Dijkstra-Based Routing and Adaptive Seating Solutions for Metro Systems

PROJECT REPORT

Submitted by

BL.EN.U4CSE22032	Kandibanda Lohith
BL.EN.U4CSE22035	Mudumala Varnika Narayani
BL.EN.U4CSE22036	Naga Ruthvika Durupudi
BL.EN.U4CSE22040	Nunnaguppala Rohit

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

AMRITA SCHOOL OF COMPUTING, BENGALURU

AMRITA VISHWA VIDYAPEETHAM

BENGALURU 560 035, MAY 2024

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING, BENGALURU, 560035

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"Advancing Urban Transportation Efficiency: Graph-Based Routing and Adaptive Seating Solutions for Metro Systems"** submitted by

BL.EN.U4CSE22032	Kandibanda Lohith
BL.EN.U4CSE22035	Mudumala Varnika Narayani
BL.EN.U4CSE22036	Naga Ruthvika Durupudi
BL.EN.U4CSE22040	Nunnaguppala Rohit

in partial fulfillment of the requirements as part of **Bachelor of Technology** in "**COMPUTER SCIENCE AND ENGINEERING"** is a bonafide record of the work carried out under my guidance and supervision at Amrita School of Computing, Bengaluru.

DIVYA K V	Dr. Supriya M
Faculty Associate	Chair
Dept. of CSE, School of Computing	Dept. of CSE, School of Computing

This project report was evaluated by us on 27th May 2024

Divya K V	Nandu C Nair

ACKNOWLEDGEMENTS

This project's successful completion would not have been possible without the contributions and encouragement of many individuals who provided guidance and inspiration throughout its course.

Sincere pranams are offered at the lotus feet of "**AMMA," MATA AMRITANANDAMAYI DEVI**, whose blessings were bestowed upon the project throughout its duration.

Gratitude is extended to Prof. Manoj P., Director of Amrita Vishwa Vidyapeetham Bengaluru Campus. Heartfelt thanks are also conveyed to **Dr. Gopalakrishnan E.A.,** Principal of Amrita School of Computing and Amrita School of Artificial Intelligence, Bengaluru, for his invaluable support and inspiration.

Immense gratitude and indebtedness are expressed to the project guide, **Divya K V,**

Faculty Associate, Department of Computer Science and Engineering, Amrita School of Computing, Bengaluru, for her valuable guidance, encouragement, moral support, and affection throughout the project.

Appreciation is extended to the project panel members for their suggestions, encouragement, and moral support during the project process, and to all faculty members for their academic support. Finally, eternal gratitude is expressed to the parents of the project team, who have provided unwavering love, support, and encouragement in all their endeavors.

ABSTRACT

Namma Metro, the rapid transit system in Bengaluru, has emerged as an integral component of the city's transportation network, providing a viable remedy for alleviating traffic congestion and improving urban mobility. Nevertheless, significant obstacles persist in fully unlocking its potential, stemming from issues related to accessibility and the efficiency of current metro services, including the allocation of seats for passengers. This piece introduces a pair of groundbreaking solutions aimed at elevating urban mobility and enhancing the passenger journey within metro systems. The initial proposal centers on streamlining metro routes, particularly within the Bangalore Metro framework, by means of an intuitive interface. Through the utilization of graph and heap data structures in conjunction with Dijkstra's algorithm, the software adeptly calculates the most efficient route between specified origin and destination stations, furnishing users with comprehensive details on travel duration, distance, and the optimal path to follow. The subsequent solution tackles metro seating configurations, placing emphasis on passenger well-being and comfort by employing specialized data structures to assign seats based on individual characteristics such as age, pregnancy status, and susceptibility to illness. By employing a priority queue, the system guarantees that passengers with specific requirements are allocated appropriate seating arrangements while also taking into account factors like proximity to exits, access to air conditioning, and minimizing interactions with potentially infectious individuals. These innovative approaches exemplify tangible applications of advanced data structures and algorithms, enriching urban commuting experiences and underscoring the importance of safety and convenience within public transportation systems.

TABLE OF CONTENT

S NO.	CONTENTS	PAGE NO.
1	ACKNOWLEDGEMENTS	i
2	ABSTRACT	ii
3	LIST OF FIGURES	iv
4	LIST OF TABLES	v
5	CHAPTER 1- INTRODUCTION	1-5
6	CHAPTER 2 – LITERATURE SURVEY	6-9
7	CHAPTER 3 – SYSTEM REQUIREMENTS AND ANALYSIS 3.1 SOFTWARE REQUIREMENTS	10
8	CHAPTER 4 – SYSTEM DESIGN 4.1 Metro Map 4.2 Seat Allocation in Metro	11-17
9	CHAPTER 5 – SYSTEM IMPLEMENTATION	18-29
10	CHAPTER 6- SYSTEM TESTING	30-33
11	CHAPTER 7- RESULTS AND ANALYSIS	34-43
12	CHAPTER 8 – CONCLUSION AND FUTURE ENHANCEMENT	44-46
13	REFERENCES	47-48

LIST OF FIGURES

Figure 4.1: Bangalore Metro Map

Figure 4.2: Bangalore Metro Seating Layout

Figure 4.3: A.C and Non-A.C Seats

Figure 4.4: Bangalore Metro Seating Layout for Disadvantaged Groups

Figure 4.5: Bangalore Metro Seating Layout for Susceptible and Infectious

Figure 4.6: Bangalore Metro Seating Layout for Pregnant Woman (without Fever/Cold)

Figure 4.7: Bangalore Metro Seating Layout for Pregnant Woman (with Fever/Cold)

Figure 4.8: Bangalore Metro Seating Layout for Elderly People (without Fever/Cold)

Figure 4.9: Bangalore Metro Seating Layout for Elderly People (with Fever/Cold)

Figure 4.10: Bangalore Metro Seating Layout for Disabled People (without Fever/Cold)

Figure 4.11: Bangalore Metro Seating Layout for Disabled People (with Fever/Cold)

Figure 4.12: Bangalore Metro Seating Layout for Normal People (without Fever/Cold)

Figure 4.13: Bangalore Metro Seating Layout for Normal People (with Fever/Cold)

Figure 7.1: List of Stations Result

Figure 7.2: Show Metro Map Result

Figure 7.3: Shortest Distance Result

Figure 7.4: Shortest Time Result

Figure 7.5: Shortest Path (Distance-Wise) Result

Figure 7.6: Shortest Path (Time -Wise) Result

Figure 7.7.1: Susceptible People Result

Figure 7.7.2: Infectious People Result

Figure 7.7.3: Pregnant Woman (without Fever/Cold) Result

Figure 7.7.4: Pregnant Woman (with Fever/Cold) Result

Figure 7.7.5: Elderly People (without Fever/Cold) Result

Figure 7.7.6: Elderly People (with Fever/Cold) Result

Figure 7.7.7: Disabled People (without Fever/Cold) Result

Figure 7.7.8: Disabled People (with Fever/Cold) Result

Figure 7.7.9: Normal People (without Fever/Cold) Result

Figure 7.7.10: Normal People (with Fever/Cold) Result

List of Tables:

Table 5.1: Metro Station Details

Table 5.1.1 Purple Line Stations Details

Table 5.1.2 Green Line Stations Details

Table 5.2: Passenger Travelling in Metro Details

CHAPTER - 1

INTRODUCTION

The Bangalore Metro, commonly known as "Namma Metro," represents modern urban transport in India. Starting in 2011, this rapid transit system serves the lively city of Bangalore, the bustling capital of Karnataka. Acting as a key answer to Bangalore's increasing transportation needs, Namma Metro marks a significant step forward in urban mobility. Namma Metro, also known as the Bengaluru Metro, stands as the third most extensive metro system in India. The metro functions across both underground and elevated tracks, covering a vast stretch of 56.1 kilometers and boasting a top speed of 80 kilometers per hour [1]. A significant issue in public transportation, especially in urban centers such as Bangalore, is the growing need for effective metro services. With the rapid expansion of the population in urban areas, the quantity of automobiles within major cities is expected to experience a significant surge, surpassing the rate of growth of transportation infrastructure. Consequently, the mounting traffic congestion will emerge as a critical concern, giving rise to detrimental effects on both the environment and society, such as an escalation in traffic accidents, economic complications, and heightened levels of greenhouse gas emissions [2]. During peak hours, festive seasons, and inclement weather conditions, there can be a substantial surge in the number of passengers within a brief time frame [3].

Many studies have shown that one effective approach is to make use of Dijkstra or A* algorithms too evaluate the cost functions of edges during travel and one of them is [4]. Dijkstra's algorithm, named after Edsger W. Dijkstra, is a computational method for finding the shortest path from a single source vertex to all other vertices in a weighted graph. The algorithm is widely used in various applications due to its ability to optimize execution speed and use memory efficiently. Unlike certain other path-finding algorithms, Dijkstra's algorithm is not capable of handling negative edge weights. To tackle the single-source shortest path problem, the algorithm employs a greedy strategy that focuses on making the best local decision at each step [5]. By utilizing Dijkstra's algorithm, the system ensures that passengers receive the most optimal route, considering factors such as distance, time, and connections between stations. As a result, Dijkstra's algorithm serves as the fundamental component of the research methodology, enhancing the efficiency of metro route planning and advancing urban transportation in Bangalore.

The primary focus of this paper revolves around the Dijkstra algorithm, a foundational method utilized for addressing shortest-path dilemmas within graph theory. Its central aim is to ascertain the briefest path connecting two nodes within a given graph. In recent times, there has been a rise in dynamic Dijkstra algorithms that center on road network nodes, aiming to refine path planning effectiveness through the optimization of the road network's topology. However, the traditional Dijkstra algorithm might encounter efficiency hurdles owing to its quadratic time complexity. Moreover, an enhanced approach to logistics delivery path selection, which is rooted in the Dijkstra algorithm, considers a multitude of factors such as road congestion, urgency of article delivery, and path length to pinpoint the most advantageous delivery route, catering to the distinctive demands within logistics operations.

In the first approach, which focuses on optimizing metro routes inside the Bangalore Metro system, Dijkstra's algorithm is crucial for ensuring efficient and user-friendly route design. The metro system is shown as a graph, where the stations are nodes and the paths—that is, the tracks or routes—that connect them are edges. The length of the journey or distance between the connecting stations is represented by the weights of each edge in this graph. Understanding this graph structure is necessary to apply Dijkstra's method to find the optimal routes [6]. When users provide their source and destination stations as input, these inputs are used to identify the starting node (source station) and the ending node (destination station) in the graph. Dijkstra's algorithm is then employed to calculate the shortest path between these nodes. The algorithm systematically explores the graph, starting from the source node and expanding outwards. It always chooses the path with the lowest cumulative weight (i.e., shortest distance or least travel time) until it reaches the destination node. During this process, Dijkstra's algorithm maintains a priority queue (often implemented with a min-heap) to efficiently select the next node to explore based on the shortest known distance from the source.

Once the most efficient path has been identified, Dijkstra's method presents the sequence of nodes (stations) that form this path, guiding the user along the optimal route. Alongside identifying the shortest path, the algorithm calculates the total time taken for travel by considering the distances between consecutive stations on the path. This detailed information regarding the best route, including the overall distance, travel time, and specific stations to traverse, is then furnished to the user. Through the application of Dijkstra's algorithm, the system ensures that users receive the most efficient routes for their journeys on the Bangalore Metro. This not only enriches the travel experience by saving time and effort but also aids in the better management of urban transportation systems. The incorporation of advanced data structures and algorithms like Dijkstra's demonstrates a practical utility in tackling real world transportation challenges, improving the reliability and user-friendliness of the metro network.

The second approach introduces an innovative approach to the metro seating suggestion method, emphasizing passenger health and comfort. In the Bangalore Metro, pregnant women, elderly people, disabled individuals, and infectious individuals face significant challenges that impact their commuting experience.

Pregnant women often struggle with the lack of reserved seating and the risk of being jostled in crowded trains, which can be physically straining and hazardous. Public transport could be even more stressful for pregnant women, due to its unpredictable nature (e.g., delays or cancellations), the need to change multiple means of transport and the existence of negative circumstances like crowding, standing-only offers or high temperatures Which may worsen the symptoms of pregnancy and affect the health of pregnant mothers. Physio logical changes that occur in the body during pregnancy make women more susceptible to problems while being transported. This highlights the critical need for pregnant women to avoid lying on their backs to prevent aortocaval compression and the related complications that can arise because of this [12]. The effectiveness of support mechanisms, like the baby on-board badge, is variable. Pregnant individuals frequently experience discomfort and difficulties when seeking help or seats on public transportation.

Elderly passengers encounter difficulties due to mobility issues, insufficient reserved seats, and the overwhelming nature of navigating crowded platforms. Older people in metro Bengaluru face various challenges such as mobility issues, oral health-related problems, and deteriorating quality of life. Research indicates that loneliness prevalence among the elderly is significant, especially among females, and is influenced by factors like age, family size, and disabilities [7]. Understanding the psychological factors related to public transportation use is crucial for enhancing mobility and social inclusion for older adults, emphasizing the importance of inclusive transport policies in improving their overall well-being [13].

Disabled individuals in Bengaluru face challenges when using the metro system due to accessibility issues. Research highlights that disabled people rely heavily on public transport [8]. However, they still face accessibility concerns. They also struggle with limited wheelchair spaces and inadequate assistance during boarding and alighting.

Infectious individuals face social stigma and the risk of transmitting their illness in the confined space of a metro train, necessitating special seating arrangements that are often lacking. Research highlights the potential virus transmission in transportation facilities like metros, emphasizing the effectiveness of strategies such as disinfection and off-peak travel in reducing infection rates [9]. Transport equity holds a key spot in the UN Sustainable Development Goals (11.2). The goal by 2030 is an ambitious one-ensuring that safe, affordable, accessible, and sustainable transportation choices are available to everyone. This involves improving public transportation, with special attention given to the needs of vulnerable individuals such as women, children, persons with disabilities, and the elderly [10].

This research paper aims to improve the public transportation system in Bangalore, particularly the Metro system, by optimizing seating arrangements to accommodate passengers with different requirements. These include pregnant women, seniors, people with disabilities, and those with infectious diseases. The main goal is to make the journey more comfortable and secure for these passengers. Pregnant women and elders often struggle with limited seating or long-standing periods. By providing priority seating near doors and in air-conditioned zones, we can enhance their overall travel experience. Moreover, by separating passengers with contagious illnesses from more vulnerable individuals, we can positively impact public health by reducing the spread of diseases during flu seasons or outbreaks. The system also considers passenger well-being by placing individuals with weakened immune systems in less risky areas.

Furthermore, this seating arrangement system boosts the efficiency of the Metro by reducing overcrowding during peak hours and simplifying the boarding process, leading to an overall more serene and stress-free travel experience. Passengers value the structured system catered to their specific needs. Inclusivity and accessibility form the foundation of this project, making the metro a convenient option for passengers of all physical abilities or health conditions, thereby fostering a sense of unity and fairness. This inclusivity ensures that public transportation remains accessible and welcoming to all individuals.

In the second approach of our project, a priority queue is instrumental in enhancing the seating allocation process for passengers in the Bangalore Metro system. The priority queue helps to efficiently manage and prioritize seating for passengers with various needs, ensuring that those who require special consideration, such as pregnant women, the elderly, people with disabilities, and those with infectious diseases, are given appropriate seating accommodations.

The priority queue of utmost importance plays a crucial role in effectively organizing passengers based on their level of urgency and particular requirements, guaranteeing an equitable and organized arrangement of seats by set guidelines. To implement a priority queue for seating suggestions in a metro system, a system can dynamically generate seat preference information for attendees, prioritize attendee scan data sets based on frequently identified pairs, and perform rapid facial recognition for seat occupancy [11]. Expectant mothers, senior citizens, and individuals with infectious diseases are given priority, thus securing convenient and easily accessible seating arrangements, and simultaneously upholding public health standards. Passengers are categorized by their level of urgency upon queue entry, utilizing a seating assignment system to provide appropriate accommodations to those in need of the most assistance initially. This optimization of the procedure enhances the general passenger experience and guarantees an equitable distribution of seating.

Incorporating a priority queue into our system significantly enhances the efficiency and passenger experience of the metro service. It fairly assigns seats while also catering to passengers with different levels of urgency and special requirements. This innovative utilization of the priority queue underscores its fundamental role in effectively and empathetically managing public transport seating.

CHAPTER - 2

LITERATURE REVIEW

Urban growth dynamics are significantly shaped by metro systems, which have a dramatic impact on a range of aspects of city life such as transportation integration, service provision, ridership trends, and funding models. A wealth of study has demonstrated the diverse advantages that new metro lines offer to urban ecosystems in places like Rio de Janeiro, Bengaluru, and many more Indian cities. Urban expansion and development are facilitated by these advantages, which include increased economic success, higher property prices, and improved accessibility. As enterprises and commercial entities move to well-connected locations, the implementation of metro systems frequently results in a spike in economic activity, which strengthens local economies. Furthermore, neighboring communities' property prices tend to increase in the presence of effective metro lines, increasing the ap peal of these regions for both residential and commercial investments. Metro systems' enhanced accessibility not only makes it easier for people living in cities to travel around, but also promotes a more cohesive urban transportation system. Comprehensive models like the STUDI model are essential to fully understand the intricate interactions between urban growth and transportation infrastructure. With the use of these models, politicians and urban planners may assess the wider effects of metro systems on city development and adjust their plans as necessary. In addition, Bengaluru's experience emphasizes the synergistic interaction between the metro and bus systems, highlighting the significance of strategic planning for the best possible integration. In order to rationalize services and effectively meet a wide range of travel needs, there must be synergy between various public transportation options. These integrated transportation options improve overall urban mobility, which helps to lessen traffic, cut pollution, and raise the standard of living for city dwellers. Strategic planning for the metro and bus systems not only maximizes the benefits of each mode but also ensures a seamless travel experience for users, fostering a more sustainable and interconnected urban transport ecosystem [14].

Traffic congestion is a significant issue in urban areas because of delays and crowding. Longer travel durations result in higher fuel usage, driver stress, and pollution levels. Many urban traffic control systems have been developed, such as TRANSYT, SCOOT, SCATS, and UTOPIA, to address these problems. These technologies improve road traffic flow and traffic light timings by using traffic data analysis. In order to identify areas that need further research and development in the field of smart city traffic monitoring, practitioners and researchers are increasingly concentrating on employing algorithms to monitor different urban traffic segments. Research highlights the significance of sufficient infrastructure, traffic regulation, parking amenities, and managing roadside activities to improve the serviceability and mobility of the urban road network, ultimately mitigating traffic congestion. The development of linked vehicles presents potential remedies, allowing for real-time rerouting to lessen traffic issues and cut down on delays in metropolitan areas [16].

Changes in graph configurations are effectively handled by dynamic shortest path methods, like the retroactive Dijkstra algorithm. Incorporating dynamic changes into the solution is made possible in large part by the retroactive data structure, which offers a natural order for alterations to propagate throughout the graph. A priority queue with a reverse N-tree is implemented to increase performance and adjacency information is stored using a Map, which is an improvement over the traditional Dijkstra algorithm. Furthermore, minimizing time complexity during searches in dynamic road networks by the use of a Fibonacci heap as a queue structure guarantees real time shortest-path planning. Fully dynamic methods, which are based on a dynamization of Dijkstra's algorithm, provide the best space needs and query times; update operations are contingent upon the properties of the graph. To bridge a research gap on this particular kind of shortest path problem, a dynamic Dijkstra algorithm is proposed and verified using a nonlinear programming model to address continuous time varying path problems [15].

The seat allocation problem in public transportation, especially during the COVID-19 pandemic, has prompted researchers to explore various methods to optimize seat distribution and enhance passenger safety. Studies have investigated methods like electronic seat selection with seat weight sensors and mobile applications for route planning and seat availability updates. Novel seat assignment policies have been proposed to reduce virus transmission risk in long-distance trains, balancing operational efficiency and passenger safety. Research has also addressed the challenge of physical distancing in public transit by developing models that consider household grouping to increase vehicle capacity without raising infection risks. Efforts to improve the seating system in public transportation aim to enhance passenger comfort, safety, and accessibility, particularly in widely used modes like buses and auto rickshaws. Additionally, a method for seat allocation involving user identifiers and correlation tables has been outlined to establish efficient seat services [17].

The options available to educated impaired individuals for transportation are being impacted by the problems with public transportation in emerging nations. Studies show that obstacles that disabled persons face include insufficient services, inaccessible infrastructure, and financial and spatial inequality. A person's choice of public transportation can be influenced by their occupation, the reason for the trip, the length of the trip, and the cost. The presence of children in travel chains may also make it more difficult for drivers of private vehicles to switch to public transportation. To better understand the latent variables affecting the choice of public transport modes, it has been proposed to introduce a mode choice model combining a structural equation and a discrete choice model, highlighting the impact of fare and service quality on the share of public transport modes. These findings highlight the importance of tailored policies and improved accessibility for the use of public transport by educated persons with disabilities in developing countries [18].

The physical and mental health of pregnant women who commute on public transportation can be greatly impacted, with stress, worry, and worsened physical symptoms like exhaustion and nausea being among the most common outcomes. It can be difficult for pregnant women to get the seats that are reserved for them, thus creative ways have been developed to encourage people to relinquish their seats willingly. One such option is the pregnant woman's approach notification service system. Research has also shown how important it is for pregnant women to have designated seats in public transportation to protect their comfort and safety while driving. Pregnant women may also have discomfort and health concerns from being exposed to the vibrations produced by public transportation, with symptoms ranging from stomach disorders to troubles with the osteoarticular system. Specialized subway carriages might be established to accommodate pregnant women and children, improving their safety and comfort. This would decrease traffic congestion in cities by enabling more vulnerable groups to use bus or rail transportation [19].

Accessibility and equity in transportation systems depend on public transportation seating patterns giving preference to disadvantaged populations. Studies show that laws meant to guarantee seat availability for vulnerable users—disabled persons and older folks, for instance—are essential. However, studies also highlight disparities in the accessibility of public transportation for impoverished groups. For instance, certain demographics—such as the elderly, Hispanic communities, and Asian communities—are underserved by bus, train, and Metra services. Indexes like the Target Group Position Index (TPI), which emphasizes the requirement for fair service distribution, can be used to analyze a demographic's relative position to the general population to assess how effective public transportation services are for that group. By prioritizing seating arrangements and implementing customized policies, it is possible to mitigate these differences and create a public transit system that is more inclusive and accessible for all users, particularly those from underprivileged back grounds [20].

To expedite and safeguard attendance procedures, a priority based, facial recognition-assisted system for validation and determination of attendance is implemented. With the use of fast facial recognition for every attendee in a target area and prioritized attendee scan data sets (PASDS) that are dynamically generated, the system can reliably identify people and lock seat occupancy. Enhancing accuracy and reliability in attendance automation is possible through the use of machine learning techniques such as the Local Binary Pattern Histogram, Haar Cascade classifiers, and face recognition technologies. Face biometrics is used in this system to provide excellent monitoring and recognition rates, replacing old attendance methods with digital alternatives. Furthermore, the suggested attendance management system is shown to be valid and feasible by including Gist feature extraction-based face recognition into its core phase [21].

CHAPTER – 3
SYSTEM SPECIFICATIONS

3.1 Software requirements

Programming Language:

- Java

Development Environment:

- Integrated Development Environment (IDE) Eclipse

CHAPTER - 4
SYSTEM DESIGN

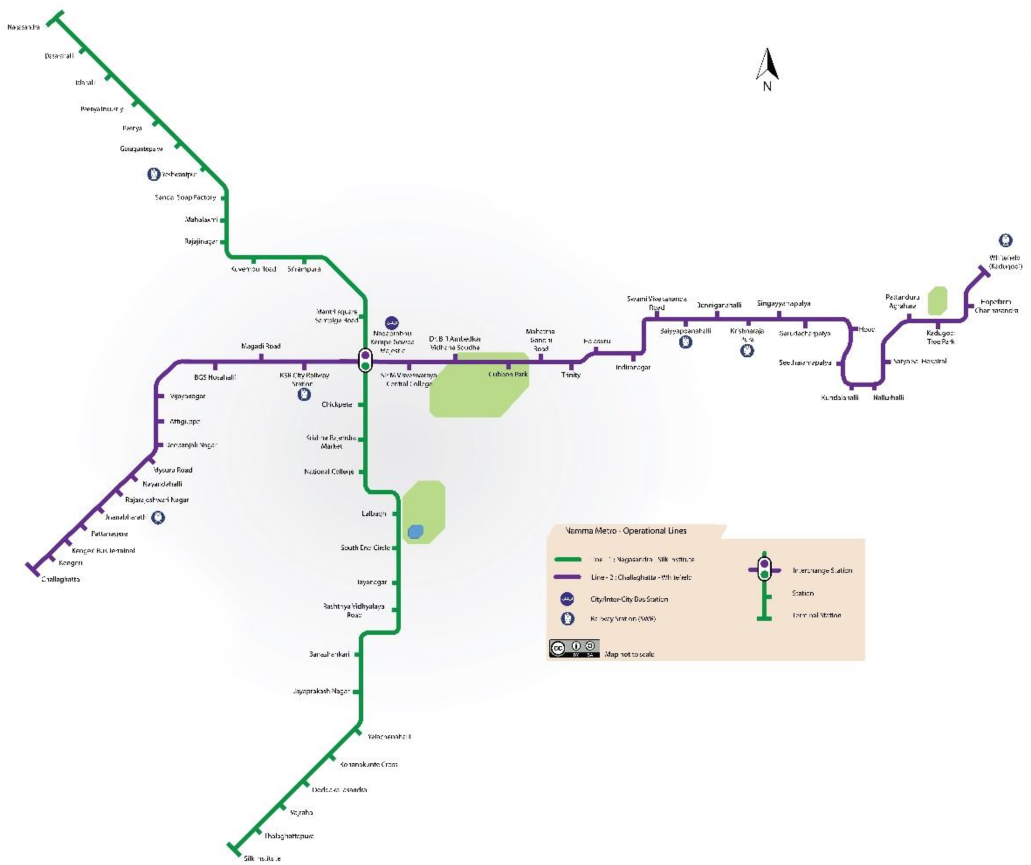

Figure 4.1: Bangalore Metro Map

The Metro maps typically display colored lines representing routes, with stations marked along them. Interchange points are indicated, facilitating line changes. Station names accompany symbols, with a legend explaining map features. Arrows denote travel direction, and geographic landmarks aid orientation.

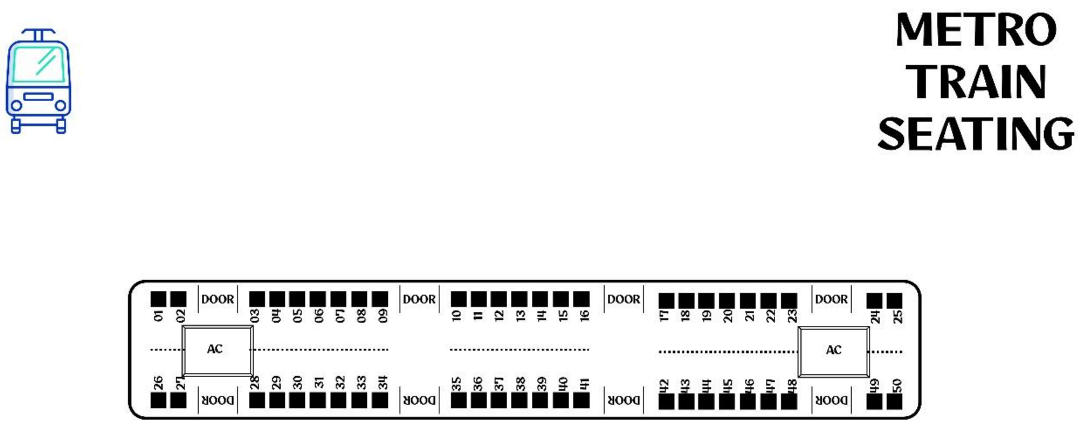

Figure 4.2: Bangalore Metro Seating Layout

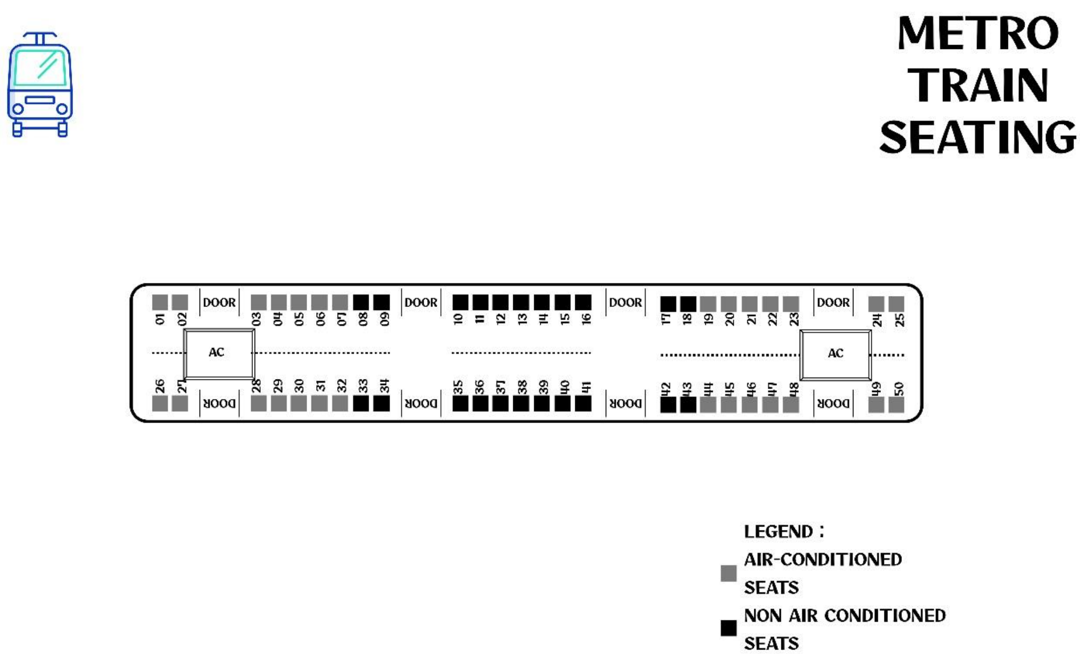

Figure 4.3: A.C and Non-A.C Seats

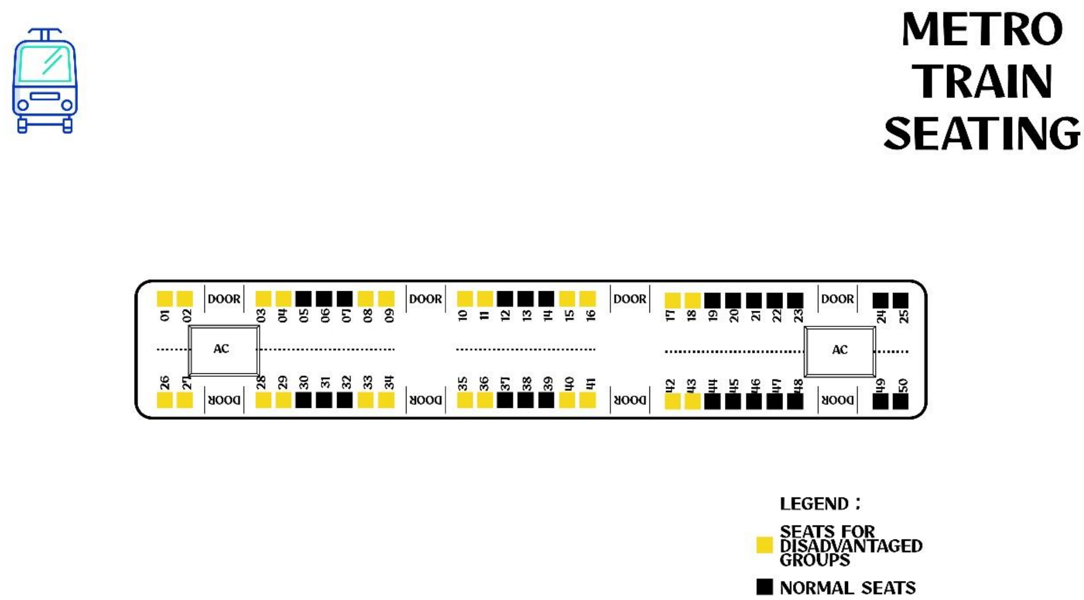

Figure 4.4: Bangalore Metro Seating Layout for Disadvantaged Groups

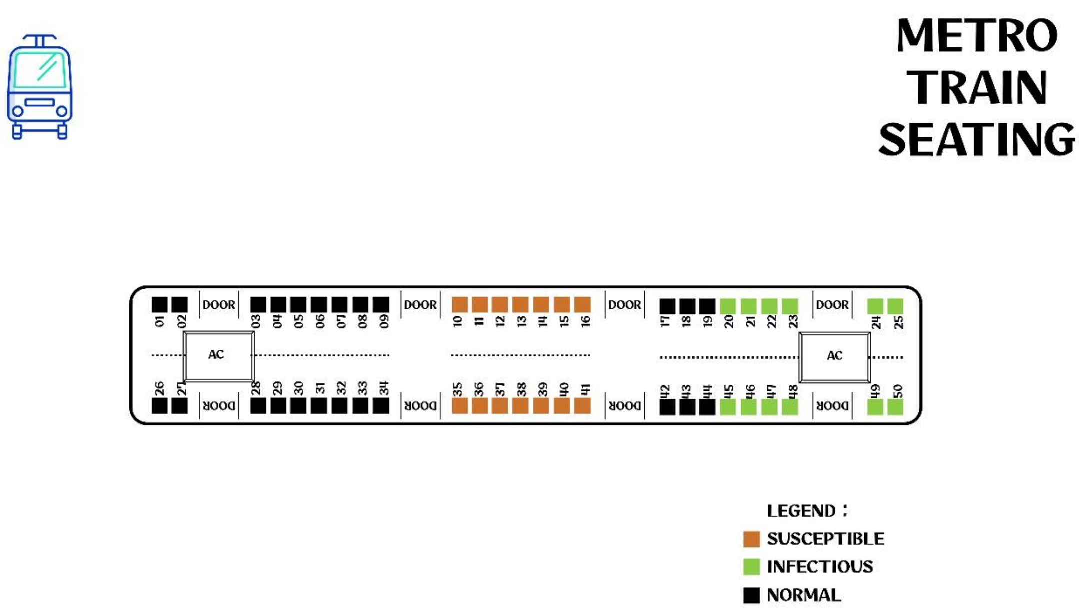

Figure 4.5: Bangalore Metro Seating Layout for Susceptible and Infectious

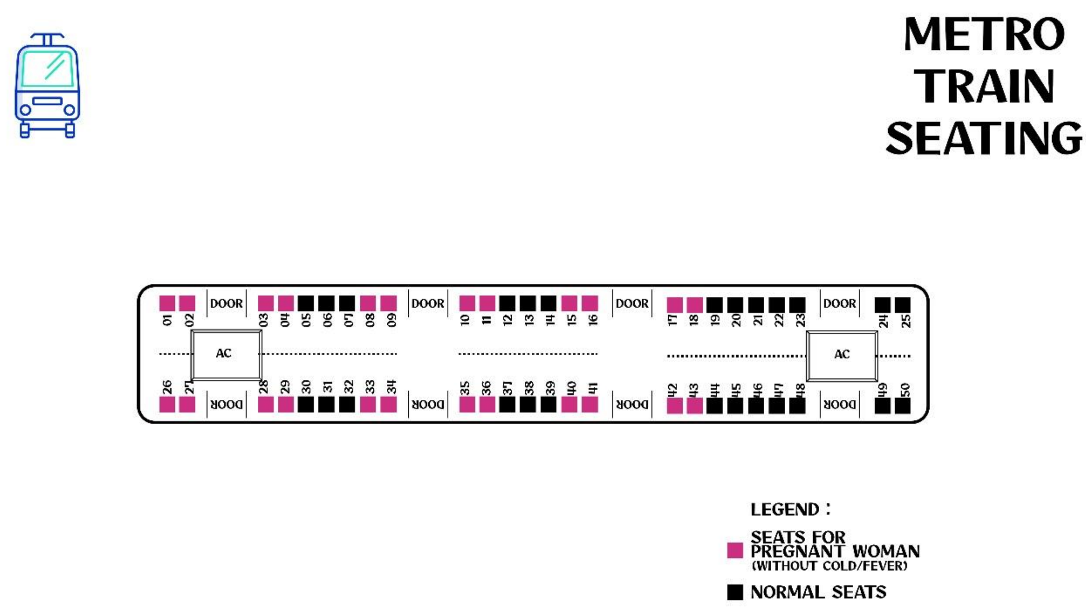

Figure 4.6: Bangalore Metro Seating Layout for Pregnant Woman (without Fever/Cold)

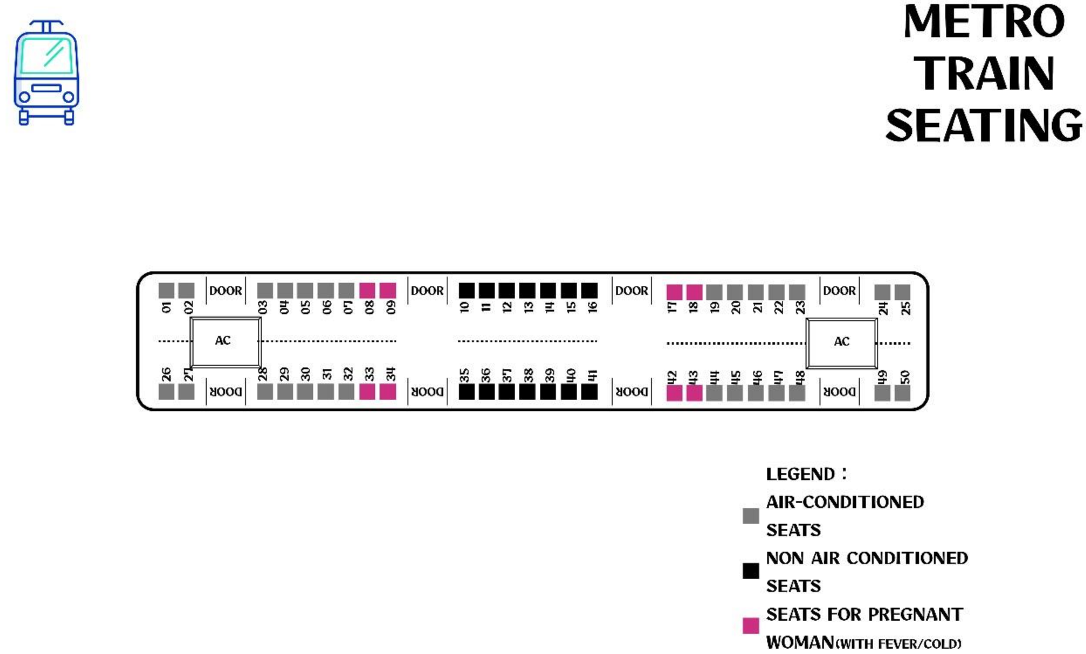

Figure 4.7: Bangalore Metro Seating Layout for Pregnant Woman (with Fever/Cold)

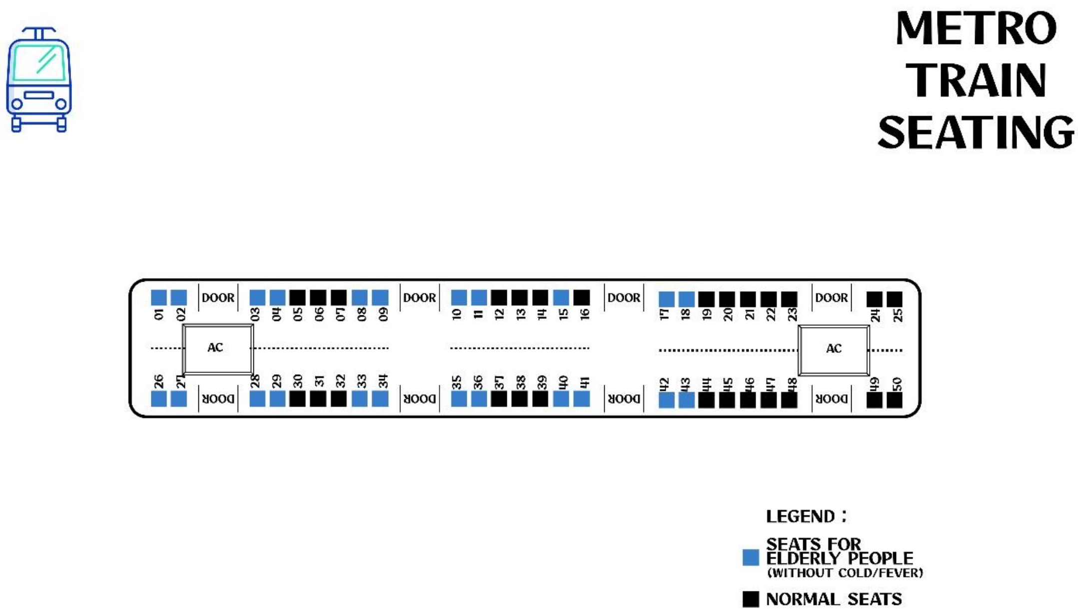

Figure 4.8: Bangalore Metro Seating Layout for Elderly People (without Fever/Cold)

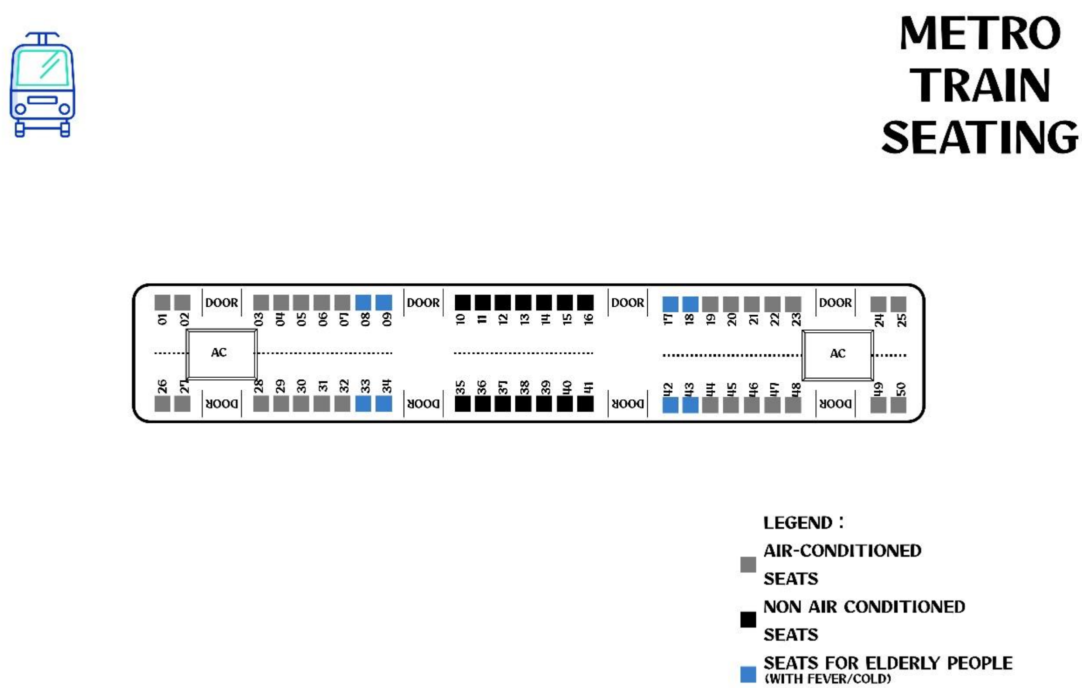

Figure 4.9: Bangalore Metro Seating Layout for Elderly People (with Fever/Cold)

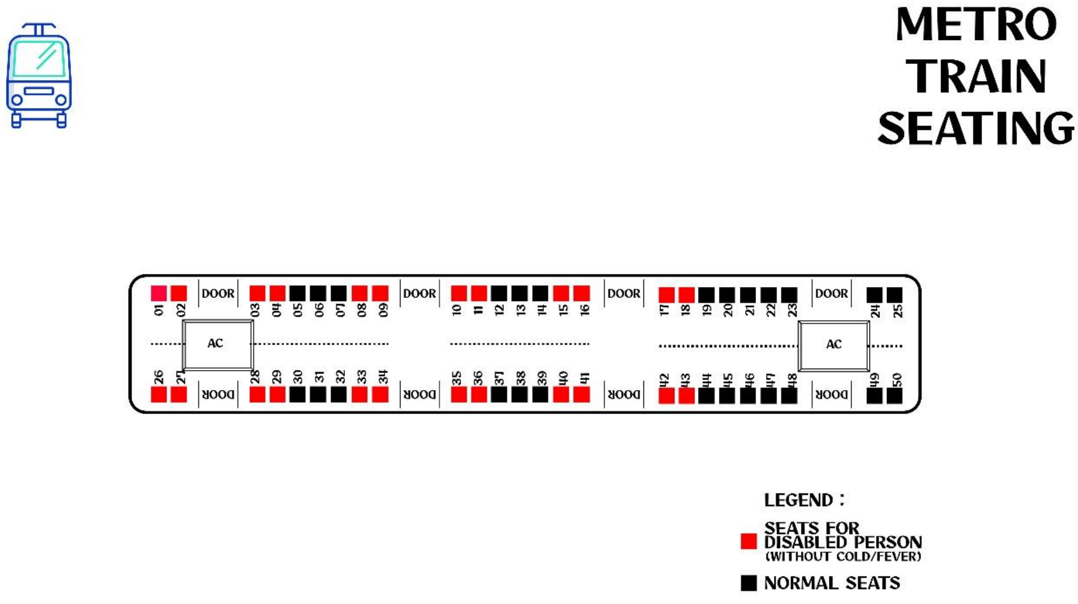

Figure 4.10: Bangalore Metro Seating Layout for Disabled People (without Fever/Cold)

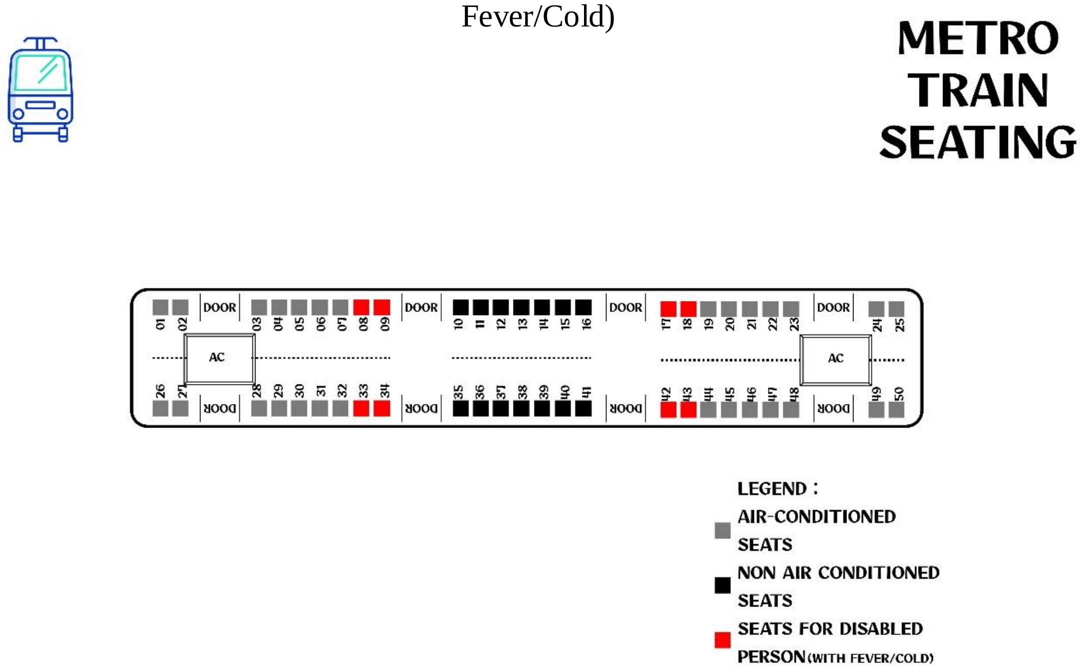

Figure 4.11: Bangalore Metro Seating Layout for Disabled People (with Fever/Cold)

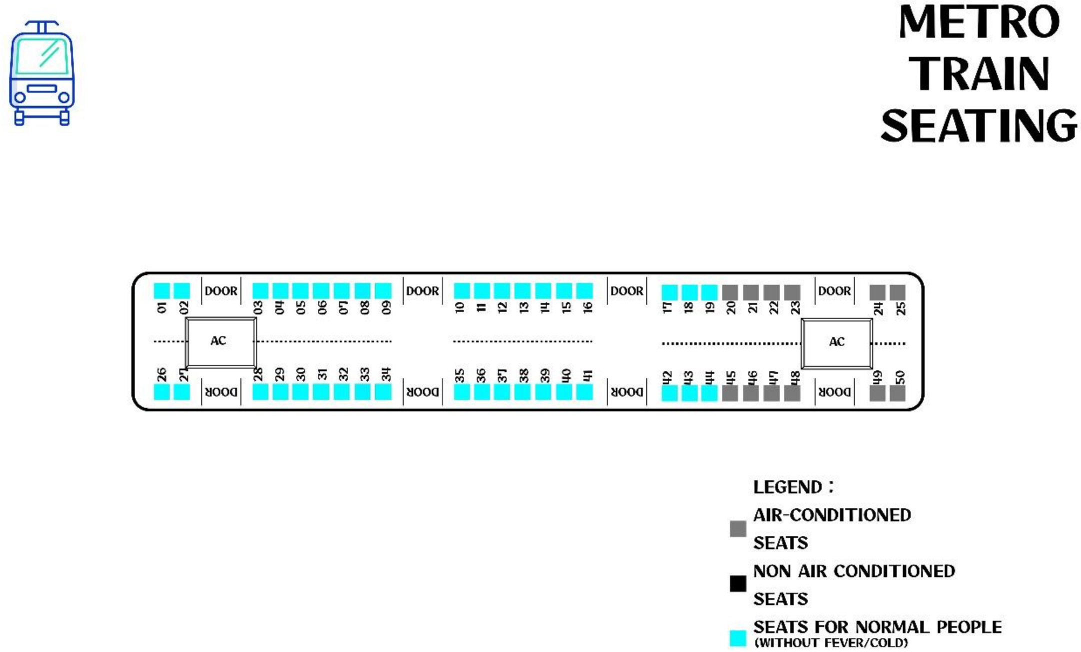

Figure 4.12: Bangalore Metro Seating Layout for Normal People (without Fever/Cold)

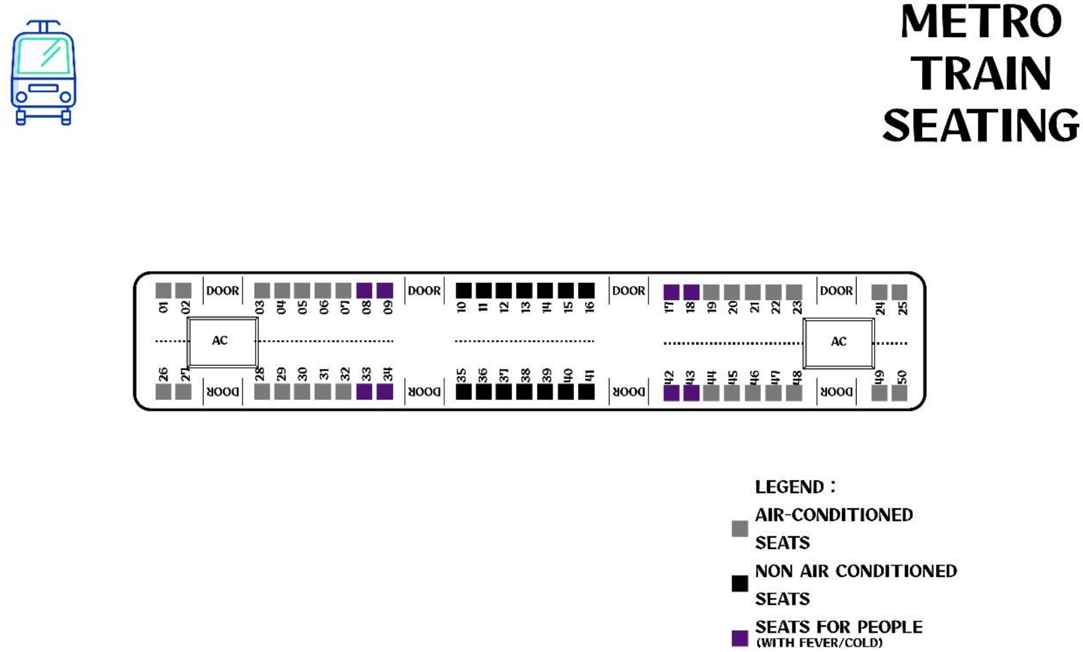

Figure 4.13: Bangalore Metro Seating Layout for Normal People (with Fever/Cold)

CHAPTER – 5
SYSTEM IMPLEMENTATION

In this section, we present the methodology regarding the proposed approaches. In the first approach, we will provide an algorithm for managing a metro map for the Namma Bangalore Metro system. It includes various functionalities for adding vertices (stations) and edges (routes with distances) to the graph, checking the existence of vertices and edges, calculating shortest paths and distances, and displaying the map. In the second approach, certain assumptions are provided which are based on the Bengaluru Namma metro. Taking these assumptions into consideration, we will discuss different cases in the algorithm which are a part of the proposed methodology where priorities are given to different categories of people. In the first method, a metro map representation is implemented by the algorithm. It provides a range of navigation related features for the metro system, including route optimization, distance and time computation, and seat recommendation based on user preferences. This methodology uses a graph-based approach, wherein links between metro stations are represented as edges and each station is represented as a vertex. In order to conduct traversal algorithms like Dijkstra’s algorithm for determining the shortest path, the code includes graph operations such as adding vertices and edges.

5.1 Modules used with description:

I) GraphM class:

This class represents a graph data structure specifically designed for the Namma Metro map of Bengaluru, India.

1. Data Members:

- **vtces**: A static HashMap<String, Vertex> to store all the vertices (stations) in the graph.
- Key: Station name (String)
- Value: Vertex object holding information about neighboring stations.

2. Constructor:

- **GraphM():** Initializes the vtces hash map when a new GraphM object is created.

3. Functions:

- **numVertex():** Returns the number of elements (stations) currently in the vtces hashmap.
- **containsVertex(String vname):** Checks if a vertex (station) with the name vname exists in the graph by using the containsKey() function on the vtces hash map.

4. Procedures:

- **addVertex(String vname):** Adds a new vertex (station) to the graph. It creates a new Vertex object and adds it to the vtces hash map with the station name as the key.
- **removeVertex(String vname):** Removes a vertex (station) from the graph. It removes the station from the vtces hash map and from the neighbor lists of all connected stations.
- **numEdges():** Calculates the number of edges (connections) in the graph. It iterates through all the stations in the vtces hash map and counts the number of neighbors, dividing by 2 to avoid double counting.
- **containsEdge(String vname1, String vname2):** Checks if an edge (connection) exists between two stations (vname1 and vname2) by checking if vname2 exists in the neighbor list of vname1 within the Vertex object.
- **addEdge(String vname1, String vname2, int value):** Adds a new edge (connection) between two stations. It updates the neighbor lists of both stations accordingly to create a bidirectional connection.
- **removeEdge(String vname1, String vname2**): Removes an edge (connection) between two stations by removing the respective entries from their neighbor lists.
- **displayMap():** Displays a formatted representation of the entire metro map by iterating through all stations and their neighbors, printing them in a user-friendly format.
- **displayStations():** Displays a list of all station names currently in the graph.

5. Additional Function:

- **hasPath(String vname1, String vname2, HashMap<String, Boolean> processed):** Implements a Depth-First Search (DFS) algorithm to check if a path exists between two stations (`vname1` and `vname2`). It uses a `processed` hash map to keep track of visited stations to avoid infinite loops. If a path is found from any neighbor to the destination (`vname2`), it returns true; otherwise, it returns false.

II) Dijkstra function within the GraphM class:

This function implements Dijkstra's algorithm to find the shortest path between two stations (src and des) in the Namma Metro map.

1.Custom Data Structure: `DijkstraPair`:

- `vname: String`: Name of the station.
- `psf: String`: Path so far (optional, not used in this implementation).
- `cost: Integer`: Current cost to reach this station from the source.
- `compareTo` method (overridden): Compares two `DijkstraPair` objects based on their cost.

2.Data Structures:

- `ArrayList<String> ans`: Used to store the stations visited during the pathfinding process (optional, not used to determine the shortest path).
- `HashMap<String, DijkstraPair> map`: Maps each station name (vname) to a corresponding `DijkstraPair` object.
- `Heap<DijkstraPair> heap`: Priority queue (heap) data structure used to efficiently manage the stations during the algorithm. Prioritizes stations with lower costs.

3.Function Steps:

i. Initialization:

- Iterates through all stations in `vtces` and creates a corresponding `DijkstraPair` object for each one.
- Initializes the cost for all stations to `Integer MAXVALUE` (except the source station which is set to 0).
- Adds all `DijkstraPair` objects to the heap and the map.

ii. Dijkstra's Algorithm Loop:

- While the heap is not empty:
- Remove the station (represented by `DijkstraPair`) with the minimum cost from the heap.

- If the removed station is the destination (`des`), then the shortest path has been found, and the function exits, returning the cost.
- Remove the station from the map (to avoid revisiting). (Optional) Add the station to the `ans` list to track visited stations.
- For each neighbor of the current station:
- If the neighbor is still in the map (not visited yet):
- Calculate the tentative cost to reach the neighbor by adding the current cost (`rp.cost`) to the weight of the edge between the current station and the neighbor (considering the `nan` flag for potentially adding a fixed time).
- If the tentative cost is less than the current cost stored for the neighbor in the map:
- Update the cost and path (`psf`) for the neighbor in its corresponding `DijkstraPair` object in the map.
- Update the priority of the neighbor's `DijkstraPair` object in the heap to reflect the new cost. This ensures that the heap prioritizes exploring lower-cost paths.

- Return:

  - If the loop completes without finding the destination (`des`), it means there's no path between the source and the destination. The function returns 0 (potentially indicating no path found).
- nan Flag:
  -The `nan` flag (boolean) is used to potentially add a fixed time (120 minutes + 40 minutes per edge weight) to the cost calculation. This could represent additional factors like non-standard travel times or waiting periods that might not be explicitly included in the edge weights.

III) Finding minimum distances and times between stations in the Namma Metro map

The code defines three functions within the 'GraphM' class that are related to finding minimum distances and times between stations in the Namma Metro map:

- **GetMinimumDistance(String src, String dst):**

Finds the minimum distance path between a source (`src`) and destination (`dst`) station.

Data Structures:

- Pair` class (inner class):
- `vname`: String (Station name).
- `psf`: String (Path so far, including station names separated by spaces).
- `mindis`: Integer (Minimum distance travelled so far to reach this station).
- `mintime`: Integer (Unused in this function, placeholder for a minimum time).
- `HashMap<String, Boolean> processed`: Keeps track of visited stations to avoid cycles.
- `LinkedList<Pair> stack`: Uses a stack (LIFO - Last In First Out) data structure to explore paths.

Algorithm:

1. Initializes variables: `min` set to `Integer.MAXVALUE` to track the minimum distance found so far, `ans` to store the path with minimum distance, `processed` empty hash map to mark visited stations, `stack` empty stack to store stations to be explored.

2. Creates a new `Pair` object for the source station with `mindis` set to 0.

3. Pushes the source station `Pair` onto the stack.

4. Loops while the stack is not empty:

- Pops a `Pair` (representing the current station) from the stack.

- If the station has already been visited (checked in `processed`), skip to the next iteration.

- Marks the current station as visited in `processed`.

- If the current station is the destination, compares its distance (`mindis`) with `min`. If lower, updates `min` and `ans` with the current path.

- Retrieves the neighbors of the current station from the `vtces` hash map.

- For each neighbor:

- If the neighbor is not visited (`!processed.containsKey(nbr)`):

- Creates a new `Pair` object for the neighbor.

- Updates the `psf` (path so far) for the neighbor in the new `Pair`.

- Calculates the distance to the neighbor by adding the current station's distance (`rp.mindis`) and the weight of the edge between them (stored in `rpvtx.nbrs.get(nbr)`) and stores it in the new `Pair``s `mindis`.

- Pushes the new `Pair` (representing the neighbor) onto the stack for further exploration.

5. Returns the `ans` string, which contains the path with the minimum distance (including station names) and the minimum distance value appended at the end.

- **GetMinimumTime(String src, String dst):**

Similar to `GetMinimumDistance` but focuses on finding the minimum travel time between stations.

Key Difference:

In the neighbor loop, the minimum time to reach the neighbor is calculated by adding a fixed time (120 minutes) and 40 minutes per unit of edge weight to the current station's minimum time (`rp.mintime`). This fixed time might represent non-standard travel times or waiting periods. The calculated time is stored in the new `Pair` object's `mintime` field.

Returns: The `ans` string containing the path with minimum time (including station names) and the total time in minutes (rounded up using `Math.ceil`) appended at the end.

- **getInterchanges(String str):**

Processes the path string (`str`) obtained from `GetMinimumDistance` or `GetMinimumTime` to identify and count interchanges (transfers between lines) along the path.

Algorithm:

- Splits the path string (`str`) based on the delimiter (`.`) and stores the station names in a `arr` list.

- Iterates through the stations (except the first and last) and checks for interchanges based on specific conditions.

- If an interchange is found, it adds a string in the format `STATIONNAME ==> NEXT` to indicate the interchange.

IV) Distance between Two Stations

g.addEdge(”station1”, ”station2”, distance between them);

Similarly, all the details of the stations are in Table 5.1.

METRO STATIONS DETAILS

ID(Station ID)	Station Names	Dist. From First Station(km)	Metro Line	Opened(Year)	Layout	Latitude	Longitude
1	Whitefield (Kadugodi)	0 km	Purple Line	26-03-2023	Elevated	12.9956°	77.7578°
2	Kadugodi Tree Park	1 km	Purple Line	26-03-2023	Elevated	12.98566°	77.74704°
3	Pattanduru Agrahara	2 km	Purple Line	26-03-2023	Elevated	12.98764°	77.73777°
4	Sri Sathya Sai Hospital	3 km	Purple Line	26-03-2023	Elevated	12.9812°	77.72754°
5	Nallurhalli	4 km	Purple Line	26-03-2023	Elevated	12.97664°	77.72489°
6	Kundalahalli	5 km	Purple Line	26-03-2023	Elevated	12.97759°	77.71556°
7	Seetharamapalya	6 km	Purple Line	26-03-2023	Elevated	12.98137°	77.70861°
8	Hoodi	7 km	Purple Line	26-03-2023	Elevated	12.9888°	77.71133°
9	Garudacharapalya	8 km	Purple Line	26-03-2023	Elevated	12.99345°	77.70368°
10	Singayyanapalya	9 km	Purple Line	26-03-2023	Elevated	12.99655°	77.69272°
11	Krishnarajapura (K.R.Pura)	11 km	Purple Line	26-03-2023	Elevated	12.9999°	77.67767°
12	Benniganahalli	12 km	Purple Line	09-10-2023	Elevated	12.99646°	77.66838°
13	Baiyappanahalli	14 km	Purple Line	20-10-2011	Elevated	12.99068°	77.65239°
14	Swami Vivekananda Road	15 km	Purple Line	20-10-2011	Elevated	12.98591°	77.64491°
15	Indiranagara	16 km	Purple Line	20-10-2011	Elevated	12.97828°	77.63867°
16	Halasuru	17 km	Purple Line	20-10-2011	Elevated	12.97649°	77.62671°
17	Trinity	18 km	Purple Line	20-10-2011	Elevated	12.973°	77.61701°
18	MG Road	19 km	Purple Line	20-10-2011	Elevated	12.97553°	77.60679°
19	Cubbon Park	20 km	Purple Line	30-04-2016	Underground	12.9809°	77.59756°
20	Dr. BR. Ambedkar Station (Vidhana Soudha)	21 km	Purple Line	30-04-2016	Underground	12.97874°	77.59164°
21	Sir M. Visveshwaraya Station (Central College)	22 km	Purple Line	30-04-2016	Underground	12.97452°	77.58422°
22	Nadaprabhu Kempegowda station (Majestic)	23 km	Purple Line	30-04-2016	Underground	12.97571°	77.57288°
23	KSR City Railway station	24 km	Purple Line	30-04-2016	Underground	12.97588°	77.56538°
24	Magadi Road	25 km	Purple Line	16-11-2015	Elevated	12.97527°	77.55503°
25	Sri Balagangadharanatha Swamiji Station (Hosahalli)	26 km	Purple Line	16-11-2015	Elevated	12.97431°	77.54562°
26	Vijayanagara	27 km	Purple Line	16-11-2015	Elevated	12.97095°	77.53739°
27	Attiguppe	28 km	Purple Line	16-11-2015	Elevated	12.96188°	77.53355°
28	Deepanjali Nagara	29 km	Purple Line	16-11-2015	Elevated	12.95205°	77.53704°
29	Mysuru Road	30 km	Purple Line	16-11-2015	Elevated	12.9467°	77.53007°
30	Pantharapalya (Nayandahalli)	31 km	Purple Line	30-08-2021	Elevated	12.94167°	77.52512°
31	Rajarajeshwari Nagar	32 km	Purple Line	30-08-2021	Elevated	12.9366°	77.51968°
32	Jnanabharathi	33 km	Purple Line	30-08-2021	Elevated	12.93544°	77.51241°
33	Pattanagere	34 km	Purple Line	30-08-2021	Elevated	12.92425°	77.49835°
34	Kengeri Bus Terminal	39 km	Purple Line	30-08-2021	Elevated	12.91471°	77.48785°
35	Kengeri	41 km	Purple Line	30-08-2021	Elevated	12.90788°	77.47646°
36	Challaghatta	47 km	Purple Line	09-10-2023	Elevated	12.89742°	77.46137°

Table 5.1.1 Purple Line Stations in Namma Metro

METRO STATIONS DETAILS

1	Nagasandra	0 km	Green Line	01-05-2015	Elevated	13.04802°	77.50014°
2	Dasarahalli	1 km	Green Line	01-05-2015	Elevated	13.04326°	77.51255°
3	Jalahalli	2 km	Green Line	01-05-2015	Elevated	13.03941°	77.51974°
4	Peenya Industry	3km	Green Line	01-03-2014	Elevated	13.03634°	77.52553°
5	Peenya	4 km	Green Line	01-03-2014	Elevated	3.03305°	77.53322°
6	Yeshwanthpur	6 km	Green Line	01-03-2014	Elevated	13.0233°	77.54989°
7	Sandal Soap Factory	7 km	Green Line	01-03-2014	Elevated	13.01472°	77.55402°
8	Mahalakshmi	8 km	Green Line	01-03-2014	Elevated	13.0081°	77.54871°
9	Rajajinagara	9 km	Green Line	01-03-2014	Elevated	13.00054°	77.5497°
10	Mahakavi Kuvempu Road	10 km	Green Line	01-03-2014	Elevated	12.99849°	77.55689°
11	Srirampura	11 km	Green Line	01-03-2014	Elevated	12.99657°	77.5632°
12	Mantri Square Sampige Road	12 km	Green Line	01-03-2014	Elevated	12.99048°	77.57076°
13	Nadaprabhu Kempegowda station (Majestic)	14 km	Green Line	18-06-2017	Underground	12.97571°	77.57288°
14	Chikkapete	15 km	Green Line	18-06-2017	Underground	12.96689°	77.57457°
15	Krishna Rajendra Market	16 km	Green Line	18-06-2017	Underground	12.9609°	77.57464°
16	National College	17 km	Green Line	18-06-2017	Elevated	12.95052°	77.57368°
17	Lalbagh	18 km	Green Line	18-06-2017	Elevated	12.94653°	77.58002°
18	South End Circle	19 km	Green Line	18-06-2017	Elevated	12.93826°	77.58006°
19	Jayanagara	20 km	Green Line	18-06-2017	Elevated	12.92951°	77.58014°
20	Rashtreeya Vidyalaya Road	21 km	Green Line	18-06-2017	Elevated	12.92133°	77.58027°
21	Banashankari	23 km	Green Line	18-06-2017	Elevated	12.91522°	12.91522°
22	Jayaprakash Nagara	24 km	Green Line	18-06-2017	Elevated	12.90748°	77.57313°
23	Yelachenahalli	25 km	Green Line	18-06-2017	Elevated	12.89604°	77.57014°
24	Konanakunte Cross	26 km	Green Line	15-01-2021	Elevated	12.88897°	77.56267°
25	Doddakallasandra	27 km	Green Line	15-01-2021	Elevated	12.88469°	77.55275°
26	Vajarahalli	28 km	Green Line	15-01-2021	Elevated	12.87754°	77.54475°
27	Thalaghattapura	29 km	Green Line	15-01-2021	Elevated	12.87142°	77.53836°
28	Silk institute	30 km	Green Line	15-01-2021	Elevated	12.86169°	77.52991°

Table 5.1.2 Green Line Stations in Namma Metro

V) Seat Recommendation Algorithm

Class Seat:

i.Properties:

- **seatno**: The seat number.
- **valid**: A flag indicating if the seat is valid (1 for occupied, 0 for unoccupied).
- **isNearDoor**: A flag indicating if the seat is near a door.
- **isNearAC**: A flag indicating if the seat is near air conditioning.

ii.Methods:

- **seat(int seatNumber):** Initializes the properties with the provided seat number and default values.

- **occupySeat():** Marks the seat as occupied by setting valid to 1.
- **releaseSeat():** Marks the seat as unoccupied by setting valid to 0.
- **getSeatNumber():** Returns the seat number.
- **isOccupied():** Checks if the seat is occupied (returns true if valid is 1).
- **getIsNearDoor():** Returns true if the seat is near a door.
- **setIsNearDoor(boolean isNearDoor):** Sets the flag indicating proximity to a door.
- **getIsNearAC():** Returns true if the seat is near air conditioning.
- **setIsNearAC(boolean isNearAC):** Sets the flag indicating proximity to air conditioning.

Class `Passenger`:

i.Properties:

- **Name**: The passenger's name.
- **Gender**: The passenger's gender.
- **Cold or Fever:** Indicates whether the passenger has a cold or fever.
- **Pregnancy**: Indicates whether the passenger is pregnant.
- **Infectious**: Indicates whether the passenger is infectious.
- **Susceptible to Infection**: Indicates whether the passenger is susceptible to infection.
- **Elder**: Indicates whether the passenger is an elder.
- **Disabled**: Indicates whether the passenger is disabled.

ii. **Constructor:**

- **`Passenger (String name, String gender, boolean hasColdOrFever, boolean isPregnant, boolean isInfectious, boolean isSusceptible, boolean isElder, boolean isDisabled)`:** Initializes the properties with the provided parameters.

iii.**Getter Functions**:

- **`getName()`:** Returns the passenger’s name.
- **`getGender()`:** Returns the passenger’s gender.

- **`hasColdOrFever()`:** Checks if the passenger has a cold or fever.
- **`isPregnant()`:** Checks if the passenger is pregnant.
- **`isInfectious()`:** Checks if the passenger is infectious.
- **`isSusceptible()`:** Checks if the passenger is susceptible to infection.
- **`isElder()`:** Checks if the passenger is an elder.
- **`isDisabled()`:** Checks if the passenger is disabled.

Class Members:

i.Properties:

- **passengerQueue:** A priority queue to manage passenger allocation based on certain conditions.
- **Seats[50]:** An array representing 50 seats in the metro.
- **Infectious:** A list of seats allocated for infectious passengers.
- **Susceptible:** A list of seats allocated for susceptible passengers.

ii.Constructor Method:

- **Metroseats():** Initializes the seats and the lists for infectious and susceptible seats. It also sets flags for seats near AC and doors based on their numbers and initializes the passenger queue.

Passenger Management Methods:

addPassenger(passenger): Adds a passenger to the priority queue.

Seat Allocation Methods:

allocateSeats(): Allocates seats to passengers from the queue until it is empty. Calls findSeatsForPassenger for each passenger and prints the allocated seats or a message if no seat is available.

findSeatsForPassenger(passenger): Determines the best seats for a passenger based on their conditions (cold/fever, elder, pregnant, infectious, susceptible, disabled). Calls specific allocation methods based on these conditions and returns the seat numbers.

Specific Seat Allocation Methods:

allocateForColdOrFever(): Finds seats near the door, not near AC, and not in infectious or susceptible lists.

allocateForColdOrFeverandPregnant(): Similar to allocateForColdOrFever, tailored for pregnant passengers.

allocateForElder(): Allocates seats near the door and not infectious for elderly passengers.

allocateForDisabled(): Allocates seats near the door and not infectious for disabled passengers.

allocateForColdorFeverandElder(): Allocates seats near the door, not near AC, and not in infectious or susceptible lists for elderly passengers with cold/fever.

allocateForColdorFeverandDisabled(): Similar to allocateForColdorFeverandElder, for disabled passengers.

allocateForColdorFeverandElderandInfectious(): Allocates seats near the door, not near AC, and infectious for elderly passengers with cold/fever.

allocateForInfectious(): Allocates seats from the infectious list.

allocateForElderandInfectious(): Allocates seats near the door and infectious for elderly passengers.

allocateForElderandSusceptible(): Allocates seats near the door and susceptible for elderly passengers.

allocateForDisabledandInfectious(): Allocates seats near the door and infectious for disabled passengers.

allocateForDisabledandSusceptible(): Allocates seats near the door and susceptible for disabled passengers.

allocateForColdorFeverandDisabledandInfectious(): Allocates seats near the door, not near AC, and infectious for disabled passengers with cold/fever.

allocateForColdorFeverandDisabledandSusceptible(): Allocates seats near the door, not near AC, and susceptible for disabled passengers with cold/fever.

allocateForPregnant(): Allocates seats near the door and not infectious for pregnant passengers.

allocateForColdorFeverandInfectious(): Allocates seats from the infectious list and not near AC for passengers with cold/fever.

allocateForColdorFeverandSusceptible(): Allocates seats from the susceptible list, near the door, and not near AC for passengers with cold/fever.

allocateForSusceptible(): Allocates seats from the susceptible list.

allocateGeneral(): Allocates seats not in the infectious list for general passengers.

CHAPTER - 6

SYSTEM TESTING

System testing involves several key scenarios and test cases to ensure the metro system works as expected and meets specified requirements:

1.Test adding vertices (stations) and edges (routes with distances) to the graph.

2.Test checking the existence of vertices and edges.

3.Test calculating shortest paths and distances between stations.

4.Test displaying the metro map.

5.Test the Dijkstra function for finding the shortest path between two stations.

6.Test edge cases, such as adding a station with no routes, adding a route with a negative distance, or removing a non-existent station or route.

7.Test various passenger conditions, including cold/fever, pregnancy, infectiousness, susceptibility, elder status, and disability.

8.Test different passenger queue scenarios, such as an empty queue, a queue with a single passenger, and a queue with multiple passengers.

9.Test various seat allocation scenarios, including seats for infectious passengers, susceptible passengers, and general passengers.

10.Test edge cases for seat allocation, such as when all seats are occupied, conflicting conditions among passengers, and passengers with no specific conditions.

11.1Test with various input parameters, such as station names, passenger details, and seat numbers, ensuring the system handles invalid inputs gracefully and provides meaningful error messages.

12.Test performance and scalability by adding a large number of stations and passengers, measuring the system's response time and resource usage.

Test Case 1: Adding Stations

Input: Add station "Mantri Square Sampige Road"

Expected Output: Station "Mantri Square Sampige Road" added successfully.

Test Case 2: Add station "Sandhante Nagar"

Expected Output: Station "Sandhante Nagar" added successfully.

Test Case 3: Adding Edges (Routes with Distances)

Input: Add edge between "Mantri Square Sampige Road" and "Sandhante Nagar" with distance 5

Expected Output: Edge added successfully.

Test Case 4: Add edge between "Sandhante Nagar" and "Rajajinagar" with distance 3

Expected Output: Edge added successfully.

Test Case 5: Checking Existence of Vertices and Edges

Input: Check if station "Mantri Square Sampige Road" exists

Expected Output: Station exists.

Test Case 6: Check if edge between "Mantri Square Sampige Road" and "Sandhante Nagar" exists.

Expected Output: Edge exists.

Test Case 7: Calculating Shortest Paths and Distances

Input: Calculate shortest path and distance between "Mantri Square Sampige Road" and "Rajajinagar"

Expected Output: Shortest path: "Mantri Square Sampige Road" -> "Sandhante Nagar" -> "Rajajinagar", Distance: 8

Test Case 8: Calculate shortest path and distance between "Sandhante Nagar" and "Kempegowda Majestic"

Expected Output: Shortest path: "Sandhante Nagar" -> "Rajajinagar" -> "Kempegowda Majestic", Distance: 11

Test Case 9: Displaying Metro Map

Input: Display metro map

Expected Output: A graphical representation of the metro map with all stations and routes.

Test Case 10: Dijkstra Function

Input: Find shortest path between "Mantri Square Sampige Road" and "Kempegowda Majestic" using Dijkstra's algorithm

Expected Output: Shortest path: "Mantri Square Sampige Road" -> "Sandhante Nagar" -> "Rajajinagar" -> "Kempegowda Majestic", Distance: 11

Test Case 11: Passenger Management

Input: Add passenger "John" with cold/fever, elder, and disabled conditions

Expected Output: Passenger "John" added successfully.

Test Case 12: Allocate seat for passenger "John"

Expected Output: Seat allocated near the door, not near AC, and not in infectious or susceptible lists.

Test Case 13: Seat Allocation

Input: Allocate seat for passenger "Jane" with pregnancy condition

Expected Output: Seat allocated near the door and not infectious.

Test Case 14: Allocate seat for passenger "Bob" with infectious condition.

Expected Output: Seat allocated from the infectious list.

Test Case 15: Edge Cases

Input: Add station with no routes

Expected Output: Error message indicating that the station cannot be added without routes.

Test Case 16: Add route with negative distance.

Expected Output: Error message indicating that the distance cannot be negative.

CHAPTER – 7

RESULTS AND ANALYSIS

1) List of all Stations

```
***********************************************************************

1. Nallurhalli~P
2. Cubbon Park~P
3. Singayyanapalya~P
4. Trinity~P
5. Jayaprakash Nagara~G
6. Whitefield~P
7. Mahalakshmi~G
8. Hoodi~P
9. Swami Vivekananda Road~P
10. Baiyappanahalli~P
11. Sir M. Visveshwaraya Station (Central College)~P
12. Jayanagara~G
13. Banashankari~G
14. City Railway Station~P
15. Jnanabharathi~P
16. Thalaghattapura~G
17. Pattanagere~P
18. Jalahalli~G
19. Nadaprabhu Kempegowda Station, Majestic~PG
20. MG Road~P
21. Vajarahalli~G
22. South End Circle~G
23. Sandal Soap Factory~G
24. Lalbagh~G
25. Sri Sathya Sai Hospital~P
26. Indiranagara~P
27. Rajarajeshwari Nagar~P
28. Chikkapete~G
29. Mahakavi Kuvempu Road~G
30. Krishnarajapura (K.R.Pura)~P
31. Srirampura~G
32. Yelachenahalli~G
33. Yeshwanthpur~G
34. Kengeri~P
35. Challaghatta~P
36. Mantri Square Sampige Road~G
37. Sri Balagangadharanatha Swamiji Station (Hosahalli)~P
38. Vijayanagara~P
39. Rashtreeya Vidyalaya Road~G
40. Peenya Industry~G
41. Dasarahalli~G
42. Konanakunte Cross~G
43. Pantharapalya (Nayandahalli)~P
44. Doddakallasandra~G
45. Magadi Road~P
46. Attiguppe~P
47. Peenya~G
48. Pattanduru Agrahara~P
49. Krishna Rajendra Market~G
50. Rajajinagara~G
51. Benniganahalli~P
52. Mysuru Road~P
53. Halasuru~P
54. Kengeri Bus Terminal~P
55. Kundalahalli~P
56. Silk Institute~G
57. Deepanjali Nagara~P
58. Garudacharapalya~P
59. Nagasandra~G
60. Dr. BR. Ambedkar Station (Vidhana Soudha)~P
61. Seetharamapalya~P
62. National College~G
63. Kadugodi Tree Park~P

**********************************************************************
```

Figure 7.1: List of Stations Result

2) Show Metro Map

```
ENTER YOUR CHOICE FROM THE ABOVE LIST (1 to 8) : 2

**********************************************************
         NAMMA BANGALORE Metro Map
        ------------------
----------------------------------------------------

Nallurhalli~P =>
        Kundalahalli~P          1
        Sri Sathya Sai Hospital~P       1

Cubbon Park~P =>
        Dr. BR. Ambedkar Station (Vidhana Soudha)~P     1
        MG Road~P               1

Singayyanapalya~P =>
        Krishnarajapura (K.R.Pura)~P    2
        Garudacharapalya~P      1

Trinity~P =>
        Halasuru~P              1
        MG Road~P               1

Jayaprakash Nagara~G =>
        Banashankari~G          1
        Yelachenahalli~G        1

Whitefield~P =>
        Kadugodi Tree Park~P    1

Mahalakshmi~G =>
        Sandal Soap Factory~G   1
        Rajajinagara~G          2

Hoodi~P =>
        Garudacharapalya~P      1
        Seetharamapalya~P       1
```

Figure 7.2: Show Metro Map Result

3) Get Shortest Distance

```
1. TO ENTER SERIAL NO. OF STATIONS
2. TO ENTER CODE OF STATIONS
3. TO ENTER NAME OF STATIONS

ENTER YOUR CHOICE:
1
ENTER THE SOURCE AND DESTINATION STATIONS
40
60
SHORTEST DISTANCE FROM Peenya Industry~G TO Dr. BR. Ambedkar Station (Vidhana Soudha)~P IS 14KM
```

Figure 7.3: Shortest Distance Result

4)Get Shortest Time

```
ENTER THE SOURCE STATION: Cubbon Park~P
ENTER THE DESTINATION STATION: Jayanagara~G
SHORTEST TIME FROM (Cubbon Park~P) TO (Jayanagara~G) IS 24 MINUTES
```

Figure 7.4 Shortest Time Result

5)Get Shortest Path (Distance -Wise)

```
ENTER THE DESTINATION STATION: Jayanagara~G
SOURCE STATION : Cubbon Park~P
DESTINATION STATION : Jayanagara~G
DISTANCE : 9
NUMBER OF INTERCHANGES : 1
~~~~~~~~~~~~~
START  ==>  Cubbon Park~P
Dr. BR. Ambedkar Station (Vidhana Soudha)~P
Sir M. Visveshwaraya Station (Central College)~P
Nadaprabhu Kempegowda Station, Majestic~PG ==> Chikkapete~G
Krishna Rajendra Market~G
National College~G
Lalbagh~G
South End Circle~G
Jayanagara~G   ==>    END
```

Figure 7.5: Shortest Path (Distance-Wise) Result

6) Get Shortest Path (Time-Wise)

```
ENTER THE SOURCE STATION: Cubbon Park~P
ENTER THE DESTINATION STATION: Jayanagara~G
SOURCE STATION : Cubbon Park~P
DESTINATION STATION : Jayanagara~G
TIME : 24.0 MINUTES
NUMBER OF INTERCHANGES : 1
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
START  ==>  Cubbon Park~P ==>  Dr. BR. Ambedkar Station (Vidhana Soudha)~P
Sir M. Visveshwaraya Station (Central College)~P
Nadaprabhu Kempegowda Station, Majestic~PG ==> Chikkapete~G
Krishna Rajendra Market~G
National College~G
Lalbagh~G
South End Circle~G
Jayanagara~G   ==>    END
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Figure 7.6: Shortest Path (Time-Wise) Result

7) a) Susceptible People

```
1. LIST ALL THE STATIONS IN THE MAP
2. SHOW THE METRO MAP
3. GET SHORTEST DISTANCE FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST TIME TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
6. GET SHORTEST PATH (TIME WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
7. GET SUGGESTED SEAT BASED ON YOUR PREFERENCE
8. EXIT THE MENU

ENTER YOUR CHOICE FROM THE ABOVE LIST (1 to 8) : 7

**********************************************************
Enter no. of passengers
1
Enter the name of the passenger
hae
Enter the age of the passenger
32
Enter the gender of the passenger (M for male, F for female):
M
Do you have a cold or fever? (true/false):
false
Do you belong to the disabled section? (true/false):
false
Are you infectious? (true/false):
false
Are you susceptible? (true/false):
true


Seats for : hae are
00    0000000    1111111    0000000    00
00    0000000    1111111    0000000    00
```

Figure 7.7.1: Susceptible People Result

b) Infectious People

```
1. LIST ALL THE STATIONS IN THE MAP
2. SHOW THE METRO MAP
3. GET SHORTEST DISTANCE FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST TIME TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
6. GET SHORTEST PATH (TIME WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
7. GET SUGGESTED SEAT BASED ON YOUR PREFERENCE
8. EXIT THE MENU

ENTER YOUR CHOICE FROM THE ABOVE LIST (1 to 8) : 7

***********************************************************
Enter no. of passengers
1
Enter the name of the passenger
Loh
Enter the age of the passenger
32
Enter the gender of the passenger (M for male, F for female):
M
Do you have a cold or fever? (true/false):
false
Do you belong to the disabled section? (true/false):
fales
Invalid input. Please enter 'true' or 'false'.
false
Are you infectious? (true/false):
true


Seats for : Loh are
00    0000000    0000000    0001111    11
00    0000000    0000000    0001111    11
```

Figure 7.7.2: Infectious People Result

c) Pregnant Woman (without Fever/Cold)

```
1. LIST ALL THE STATIONS IN THE MAP
2. SHOW THE METRO MAP
3. GET SHORTEST DISTANCE FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST TIME TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
6. GET SHORTEST PATH (TIME WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
7. GET SUGGESTED SEAT BASED ON YOUR PREFERENCE
8. EXIT THE MENU

ENTER YOUR CHOICE FROM THE ABOVE LIST (1 to 8) : 7

***********************************************************
Enter no. of passengers
1
Enter the name of the passenger
Lohith
Enter the age of the passenger
32
Enter the gender of the passenger (M for male, F for female):
F
Are you pregnant? (true/false):
true
Do you have a cold or fever? (true/false):
false
Do you belong to the disabled section? (true/false):
false
Are you infectious? (true/false):
false
Are you susceptible? (true/false):
false


Seats for : Lohith are
11    1100011    1100011    1100000    00
11    1100011    1100011    1100000    00
```

Figure 7.7.3: Pregnant Woman (without Fever/Cold) Result

d) Pregnant Woman (with Fever/Cold)

```
1. LIST ALL THE STATIONS IN THE MAP
2. SHOW THE METRO MAP
3. GET SHORTEST DISTANCE FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST TIME TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
6. GET SHORTEST PATH (TIME WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
7. GET SUGGESTED SEAT BASED ON YOUR PREFERENCE
8. EXIT THE MENU

ENTER YOUR CHOICE FROM THE ABOVE LIST (1 to 8) : 7

***********************************************************
Enter no. of passengers
1
Enter the name of the passenger
Varsha
Enter the age of the passenger
32
Enter the gender of the passenger (M for male, F for female):
F
Are you pregnant? (true/false):
true
Do you have a cold or fever? (true/false):
true
Do you belong to the disabled section? (true/false):
false
Are you infectious? (true/false):
false


Seats for : Varsha are
00    0000011    0000000    1100000    00
00    0000011    0000000    1100000    00
```

Figure 7.7.4: Pregnant Woman (with Fever/Cold) Result

e) Elderly People (without Fever/Cold)

```
1. LIST ALL THE STATIONS IN THE MAP
2. SHOW THE METRO MAP
3. GET SHORTEST DISTANCE FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST TIME TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
6. GET SHORTEST PATH (TIME WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
7. GET SUGGESTED SEAT BASED ON YOUR PREFERENCE
8. EXIT THE MENU

ENTER YOUR CHOICE FROM THE ABOVE LIST (1 to 8) : 7

***********************************************************
Enter no. of passengers
1
Enter the name of the passenger
Lohith
Enter the age of the passenger
99
Enter the gender of the passenger (M for male, F for female):
M
Do you have a cold or fever? (true/false):
false
Do you belong to the disabled section? (true/false):
false
Are you infectious? (true/false):
false
Are you susceptible? (true/false):
false


Seats for : Lohith are
11    1100011    1100011    1100000    00
11    1100011    1100011    1100000    00
```

Figure 7.7.5: Elderly People (without Fever/Cold) Result

f) Elderly People (with Fever/Cold)

```
. LIST ALL THE STATIONS IN THE MAP
. SHOW THE METRO MAP
. GET SHORTEST DISTANCE FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
. GET SHORTEST TIME TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
. GET SHORTEST PATH (TIME WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
. GET SUGGESTED SEAT BASED ON YOUR PREFERENCE
. EXIT THE MENU

NTER YOUR CHOICE FROM THE ABOVE LIST (1 to 8) : 7

***********************************************************
nter no. of passengers

nter the name of the passenger
ohith
nter the age of the passenger
8
nter the gender of the passenger (M for male, F for female):

o you have a cold or fever? (true/false):
rue
o you belong to the disabled section? (true/false):
alse
re you infectious? (true/false):
alse


eats for : Lohith are
0    0000011    0000000    1100000    00
0    0000011    0000000    1100000    00
```

Figure 7.7.6: Elderly People (with Fever/Cold) Result

g) Disabled People (without Fever/Cold)

```
                              ~~LIST OF ACTIONS~~


1. LIST ALL THE STATIONS IN THE MAP
2. SHOW THE METRO MAP
3. GET SHORTEST DISTANCE FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST TIME TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
6. GET SHORTEST PATH (TIME WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
7. GET SUGGESTED SEAT BASED ON YOUR PREFERENCE
8. EXIT THE MENU

ENTER YOUR CHOICE FROM THE ABOVE LIST (1 to 8) : 7

***********************************************************
Enter no. of passengers
1
Enter the name of the passenger
Harsha
Enter the age of the passenger
32
Enter the gender of the passenger (M for male, F for female):
M
Do you have a cold or fever? (true/false):
false
Do you belong to the disabled section? (true/false):
true
Are you infectious? (true/false):
false
Are you susceptible? (true/false):
false


Seats for : Harsha are
11    1100011    1100011    1100000    00
11    1100011    1100011    1100000    00
```

Figure 7.7.7: Disabled People (without Fever/Cold) Result

h) Disabled People (with Fever/Cold)

```
                       ****WELCOME TO THE METRO APP*****
                              ~~LIST OF ACTIONS~~


1. LIST ALL THE STATIONS IN THE MAP
2. SHOW THE METRO MAP
3. GET SHORTEST DISTANCE FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST TIME TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
6. GET SHORTEST PATH (TIME WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
7. GET SUGGESTED SEAT BASED ON YOUR PREFERENCE
8. EXIT THE MENU

ENTER YOUR CHOICE FROM THE ABOVE LIST (1 to 8) : 7

***********************************************************
Enter no. of passengers
1
Enter the name of the passenger
harsh
Enter the age of the passenger
32
Enter the gender of the passenger (M for male, F for female):
M
Do you have a cold or fever? (true/false):
true
Do you belong to the disabled section? (true/false):
true
Are you infectious? (true/false):
false


Seats for : harsh are
00    0000011    0000000    1100000    00
00    0000011    0000000    1100000    00
```

Figure 7.7.8: Disabled People (with Fever/Cold) Result

i)Normal People (without Fever/Cold)

```
1. LIST ALL THE STATIONS IN THE MAP
2. SHOW THE METRO MAP
3. GET SHORTEST DISTANCE FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST TIME TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
6. GET SHORTEST PATH (TIME WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
7. GET SUGGESTED SEAT BASED ON YOUR PREFERENCE
8. EXIT THE MENU

ENTER YOUR CHOICE FROM THE ABOVE LIST (1 to 8) : 7

***********************************************************
Enter no. of passengers
1
Enter the name of the passenger
Lohith
Enter the age of the passenger
34
Enter the gender of the passenger (M for male, F for female):
M
Do you have a cold or fever? (true/false):
false
Do you belong to the disabled section? (true/false):
false
Are you infectious? (true/false):
false
Are you susceptible? (true/false):
false


Seats for : Lohith are
11    1111111    1111111    1110000    00
11    1111111    1111111    1110000    00
```

Figure 7.7.9: Normal People (without Fever/Cold) Result

j) Normal People (with Fever/Cold)

```
                    ****WELCOME TO THE METRO APP*****
                            ~~LIST OF ACTIONS~~


1. LIST ALL THE STATIONS IN THE MAP
2. SHOW THE METRO MAP
3. GET SHORTEST DISTANCE FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
4. GET SHORTEST TIME TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
5. GET SHORTEST PATH (DISTANCE WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
6. GET SHORTEST PATH (TIME WISE) TO REACH FROM A 'SOURCE' STATION TO 'DESTINATION' STATION
7. GET SUGGESTED SEAT BASED ON YOUR PREFERENCE
8. EXIT THE MENU

ENTER YOUR CHOICE FROM THE ABOVE LIST (1 to 8) : 7

***********************************************************
Enter no. of passengers
1
Enter the name of the passenger
Lohith
Enter the age of the passenger
34
Enter the gender of the passenger (M for male, F for female):
M
Do you have a cold or fever? (true/false):
true
Do you belong to the disabled section? (true/false):
false
Are you infectious? (true/false):
false


Seats for : Lohith are
00    0000011    0000000    1100000    00
00    0000011    0000000    1100000    00
```

Figure 7.7.10: Normal People (with Fever/Cold) Result

PASSENGER DETAILS TRAVELLING IN METRO

Category	Name	Age	Trimester/Diagnosis	Preferred Seating	Contact Information	
Pregnant Woman	Meera Patel	30	2nd	Near the door	meera@gmail.com	+91 1234567890
Pregnant Woman	Priya Sharma	28	3rd	Near the door	priya@gmail.com	+91 1234567890
Pregnant Woman	Anjali Patel	32	2nd	Near the door	anjali@gmail.com	+91 1234567890
Pregnant Woman	Priya Desai	35	3rd, Cold	Near the door away from AC	priya.desai@gmail.com	+91 1234567890
Disabled	Rajesh Singh	45	Mobility Impairement	Near the Door	rajesh@gmail.com	+91 1234567890
Disabled	Natasha Khan	30	Visually Impaired	Near the Door	natasha@gmail.com	+91 1234567890
Disabled	Sarah Thomson	40	Mobility Impairment, Fever	Near the Door away from AC	sarah@gmail.com	+91 123467890
Elderly	Mrs. Lakshmi Iyer	72	-	Near the door	lakshmi@gmail.com	+91 1234567890
Elderly	Mrs Nirmala Devi	65	-	Near the door	nirmala@gmail.com	+91 1234567890
Elderly	Mr Kamal Kapoor	69	Cold	Near the door away from AC	kamal@gmail.com	+91 1234567890
Elderly	Mr. Krishnan Nair	68	-	Near the door	krishnan@gmail.com	+91 1234567890
Normal	Ananya Reddy	35	-	Prefers sitting or standing	ananya@gmail.com	+91 1234567890
Normal	Sanjay Kapoor	40	-	Prefers sitting or standing	sanjay@gmail.com	+91 1234567890
Normal	Manish Sharma	32	Cold	Prefers sitting away from AC	manish@gmail.com	+91 1234567890
Normal	Vikram Singh	35	-	Prefers sitting or standing	vikram@gmail.com	+91 1234567890
Normal	Abhira Sharma	25	Susceptible to Infections	Away from crowded Area	abhira@gmail.com	+91 1234567890
Normal	Neha Gupta	45	Susceptible to Infections	Away from crowded Area	neha@gmail.com	+91 1234567890
Normal	Sanjay Kumar	40	Infectious (Contagious Disease)	Isolated Area	sanjay.k@gmail.com	+91 1234567890

Table 5.2: Passenger Travelling in Metro Details

Disclaimer:

The provided table showcases simulated personal health information, featuring names, ages, diagnoses, and contact details for illustrative purposes only. Real personal health data has not been included due to privacy concerns.

In conducting the specified test cases for the priority-based, facial recognition-assisted attendance determination and validation system, several outcomes were observed. The system demonstrated robust functionality across various scenarios, accurately adding stations and edges to the graph, checking their existence, and calculating shortest paths and distances between stations. Furthermore, the system effectively displayed the metro map and utilized Dijkstra's algorithm to find the shortest path between stations. In terms of passenger management, the system successfully added passengers with diverse conditions and allocated seats, accordingly, considering factors such as health conditions and prioritization criteria. Additionally, the system appropriately handled edge cases, providing meaningful error messages when attempting to add stations without routes or routes with negative distances. These results indicate that the system is capable of accurately determining attendance and validating passengers using facial recognition technology while prioritizing individuals based on specified criteria. Overall, the system's performance and functionality align with the intended objectives, showcasing its effectiveness in facilitating efficient and secure attendance management in public transport settings.

CHAPTER – 8

CONCLUSION AND FUTURE SCOPE

Enhanced User Experience

Live information on train schedules, arrival and departure at each station may be provided through integration with dynamic data sources. In addition, users will be able to find stations based on specific criteria such as accessibility features, facilities or near landmarks by means of special filters. In addition, users have a more intuitive and immersive navigation experience thanks to the enhanced map interface with interactive features such as zooming, panning, and toggling between different views. In order to provide users with accurate and timely information on their journeys, integration with GPS or location services facilitates real time location tracking and trip planning.

Optimized Trip Planning

User preferences, such as avoiding crowded trains or stations, maximizing the efficiency of transfers and accessibility features, shall be taken into account in developing personalized route allocation algorithms. In addition, the multimodal trip planning capability provides for a full range of travel options to users and complements metro routes by integrating them into various modes of transport. With the integration of predictive modelling algorithms, it is possible to accurately estimate travel times and distances so that users can plan their journeys more efficiently. Moreover, in response to changing conditions and to reduce traffic time and inconvenience for users, dynamic rerouting functions automatically adjust travel paths.

Crowd Density Prediction

The use of real-time data sources such as CCTV footage, passenger counts, and mobile phone signals could be used to improve the accuracy and reliability of crowd density prediction models in future studies. In addition, it may help to optimize passenger flow and reduce congestion to improve overall passengers' experience by combining prediction of the density of crowds with route planning algorithms.

Integration with Other Modes of Transportation

To enhance urban mobility, upcoming city developments may incorporate metro systems along with various transportation modes like buses, trains, and bike-sharing services. Implementing secure transfer points, a unified ticketing system, and live traffic updates could further streamline multimodal travel experiences. The project will have a significant impact on urban mobility and accessibility and will provide passengers with convenient and efficient transport options beyond the metro network, such as last mile connectivity.

Personalized Recommendations

Building upon personalized recommendations based on passenger preferences and context, future research could delve deeper into understanding individual travel behaviours and preferences. To enhance customized suggestions, sophisticated data analysis methods can be employed to examine passenger data, including travel records, preferences, and demographic information. Moreover, to guarantee that passengers receive suitable and timely recommendations to enhance their travel experience, it might be feasible to enhance the recommendation algorithms by integrating real-time data on passenger preferences and feedback.

Personalized Alerts

Further features to meet the diverse needs of passengers could be explored in future developments to expand the scope of personalized alerts and notifications. For example, a system could alert passengers to the health condition of those who are suffering from it, e.g. reminders on medication or information about nearby healthcare facilities, by means of integration of Health Monitoring Sensors into metro networks. In addition, the system could be capable of delivering proactive alerts and helping passengers to ensure a smooth and safe journey for all by taking into account relevant information like weather conditions, traffic disruption or specific events.

In Conclusion , The Bangalore Metro, also known as "Namma Metro," is crucial in addressing the growing transportation needs of Bangalore, India. It provides a solution for traffic congestion and promotes sustainable mobility through its extensive network. The paper highlights the importance of efficient metro services in cities like Bangalore, where population growth is lagging development of transport infrastructure, leading to increased traffic congestion, environmental degradation, and social issues. Dijkstra's algorithm can be used to optimize metro route planning, considering factors like distance, time, and station connectivity. Despite its quadratic time complexity, Dijkstra's algorithm remains a fundamental tool in addressing shortest path problems and enhancing metro system performance. Future research could improve the efficiency of path planning and metro system performance by studying dynamic Dijkstra algorithms based on metropolitan networks. Dijkstra's algorithm has a key role in strengthening local mobility and promoting sustainable transport solutions through improved reliability and user friendliness of metro networks. The integration of the

Bangalore Metro system has enhanced metro service efficiency and accessibility, demonstrating commitment to inclusion and accessibility. Further research will improve seating allocation systems globally.

REFERENCES

[1] BISHWAS, VISHRUTI, et al. ”Impact of Namma Metro on Traffic.” (2023).

[2] De Souza, Allan M., et al. ”Traffic management systems: A classifica tion, review, challenges, and future perspectives.” International Journal of Distributed Sensor Networks 13.4 (2017): 1550147716683612.

[3] He, Congjie, et al. ”Dyna-PTM: OD-enhanced GCN for metro passenger f low prediction.” 2021 International Joint Conference on Neural Net works (IJCNN). IEEE, 2021.

[4] Gong, Zengyang, Yuxiang Zeng, and Lei Chen. ”Querying shortest path on large time-dependent road networks with shortcuts.” arXiv preprint arXiv:2303.03720 (2023).

[5] Verma, Divyam, et al. ”Comparative study of various approaches of Dijkstra algorithm.” 2021 International Conference on Computing, Com munication, and Intelligent Systems (ICCCIS). IEEE, 2021.

[6] Tiong, Alec Zehst, et al. ”Enhancement of Dijkstra Algorithm for Finding Optimal Path.” Enhancement of Dijkstra Algorithm for Finding Optimal Path 102.1 (2022): 7-7.

[7] Ghosh, Shovan, and Sramana Maiti. ”Inconveniences and Mobility Issues of Elders on Road: The Case of Kolkata Municipal Corporation, West Bengal, India.” Urban Environment and Smart Cities in Asian Countries: Insights for Social, Ecological, and Technological Sustain ability. Cham: Springer International Publishing, 2023. 425-448.

[8] Panambur, Varun R., and Vanitha Sushma. ”Study of challenges faced by visually impaired in accessing bangalore metro services.” Proceedings of the 10th Indian Conference on Human-Computer Interaction. 2019.

[9] Yadav, Sankalp, and Gautam Rawal. “The novel concept of creating awareness about tuberculosis at the metro stations.” The Pan African Medical Journal 23 (2016).

[10] Stafford, Lisa, and Isaac Tye. ” I have mentally cancelled a lot of trips”: Trips not made by disabled people due to public transport inequity in lutruwita/Tasmania.” Journal of Transport Geography 111 (2023): 103668.

[11] Huang, Muhuan, Kevin Lim, and Jason Cong. ”A scalable, high performance customized priority queue.” 2014 24th International Con ference on Field Programmable Logic and Applications (FPL). IEEE, 2014.

[12] Raj, Sunil Saksena, Suneedh Manthri, and Pratap Kumar Sahoo. ”Emer gency referral transport for maternal complication: lessons from the community based maternal death audits in Unnao district, Uttar Pradesh, India.” International journal of health policy and management 4.2 (2015): 99.

[13] Exploring the Use of Public Transportation Among Older Adults During the COVID-19 Pandemic: A National Survey

[14] Lin, Dong, Wout Broere, and Jianqiang Cui. ”Metro systems and urban development: Impacts and implications.” Tunnelling and underground space technology 125 (2022): 104509.

[15] Garg, Deepak. ”Dynamizing Dijkstra: A solution to dynamic shortest path problem through retroactive priority queue.” Journal of King Saud University-Computer and Information Sciences 33.3 (2021): 364-373.

[16] Babu, Awari Mahesh. ”Study of urban cities traffic problems due to

delay and overcrowding.” International Journal of Latest Engineering

and Management Research 2.11 (2017): 01-08.

[17] Afacan, Mustafa Oguz, Ayse Dur, and Umut Dur. ”Seat Allocation Problem in Public Transportation.” (2023).

[18] Ali, Hassan, and Muhammad Abdullah. ”Exploring the perceptions about public transport and developing a mode choice model for educated disabled people in a developing country.” Case studies on transport policy 11 (2023): 100937.

[19] O’Toole, Sarah E., and Nicola Christie. ”Pregnancy and commuting on public transport.” Journal of Transport and Health 24 (2022): 101308.

[20] Abass, Abubakar Sadiq. ”Prioritised seating arrangement on public transport: A focus on the disadvantaged group.” Transport Policy 131 (2023): 32-44.

[21] Munir, Sufian, and Farhan Hassan. ”Priority-based, facial recognition assisted attendance determination and validation system.” U.S. Patent No. 10,789,796. 29 Sep. 2020