

Multi-Modal Route Recommender System for Bangkok Public Transportation

Kittiya Poonsilp

Abstract: Currently, Bangkok has a 151 kilometers service of a rail line, whereas the total plan is 540 kilometers. More rail lines are now under construction and supposed to be done by a few years. Regarding a massive public transportation network, we need a route recommender system to make traveling more efficient. This paper proposes the route recommender system which supports multi modes of transportation in Bangkok, including BTS, MRT, ARL, BMTA bus, and Chaophraya Riverboat. Users can see suggested routes and sort routes by travel time, fare, number of transfer, and overall score. The A* algorithm with the Heuristic formula as the heuristic function is used to calculate the possible routes. Then the best route is selected based on the score, which is calculated from four factors: travel time, fare, number of transfer, and distance. The database contains 13,510 stops, and the results show that the system can suggest accurate routes within a few seconds, which is fast enough for all use cases and achieved overall user satisfaction at 84.8% from our user experience survey.

Keywords: Public transit, Route recommender, Shortest path, Transportation.

I. INTRODUCTION

The rail transportation network in Bangkok and the metropolitan area is now expanding rapidly. According to the M-Map plan, 12 transit lines, which consist of two commuter lines, one airport line, five rapid transit lines, and four monorail lines, totaling 540 kilometers will be operated by 2029. Traveling in such a vast transportation network may require several transfers across several transit lines, which can be complicated, especially for a first-time traveler.

In order to make a traveler more comfortable to travel in Bangkok and the metropolitan area, we need to develop a route recommender system that supports all public transportation modes: BTS, MRT, ARL, BMTA bus, and Chaophraya Riverboat. A user journey should be straightforward by entering a starting location and desired destination with their optional preferences, e.g., preferred modes of transportation, sorting preferences: by fare, travel time, number of transfers, and walking distance. The system then calculates routes based on the user preferences and show all suggested routes back to the user.

This research also introduces the algorithm to calculate the best route based on scoring that balances between fare, travel time, and transfers.

II. RELATED WORK

A. Bozyigit et al. [1] proposed a similar system for public transport route recommendations. They focused and tested on the public transport network of Izmir, Turkey having 7,704 stations and 43,467 connections between these stations. The system recommends routes concerning multiple factors such that a number of transfers, total distance, and walking distance using the Space P modeling technique and a Dijkstra's algorithm. In the experimental results, their method finds the optimal route in milliseconds.

A. Bozyigit et al. [2] also proposed another technique to calculate the shortest path. Dijkstra's algorithm is mostly used in order to find the shortest path, but it is not efficient for public transport route planning because it ignores the number of transfer and walking distances. Thus, they modified Dijkstra's algorithm by implementing the penalty system. The proposed system is quite efficient for route planning in terms of the number of transfers, the distance of the proposed route, and walking distance.

W. Almobaideen et al. [3] proposed another exciting idea to calculate a route based on two service preferences. The first preference is the public transportation mode, which could include bus, train, metro, and walking. The second preference is the service that includes the best network connection available, such as GPRS, 3G, and 4G along the available paths. They think that a combination of these two preferences is of high importance in daily life.

III. ANALYSIS AND DESIGN

A. Bangkok Public Transportation Network Analysis

The transportation network data in Bangkok and metropolitan areas are kept in the GTFS format consisting of 12 transit agencies, 1,438 transit routes, and 13,510 stations/stops. The heatmap of all stops shown in Fig. 1 illustrates the density of the public transportation network in Bangkok. Fig. 2 shows the master plan of the rail transportation system in Bangkok and the metropolitan area (M-Map).

Revised Manuscript Received on February 18, 2020.

* Correspondence Author

Kittiya Poonsilp*, Department of Computer Science, Faculty of Science, Suan-Sunandha Rajabhat University, Bangkok, Thailand.
E-mail: kittiya.po@ssru.ac.th

Multi-Modal Route Recommender System for Bangkok Public Transportation

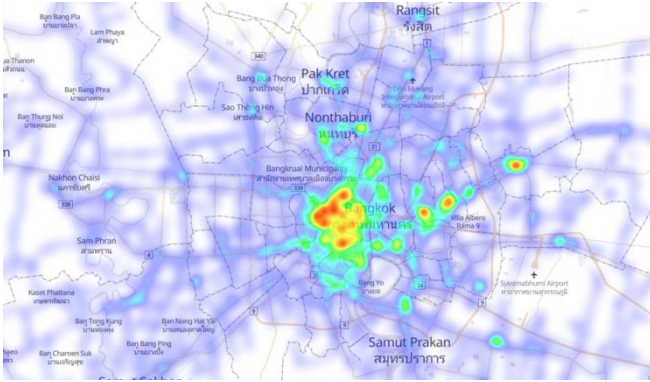


Fig. 1. Heatmap of the transportation network in Bangkok

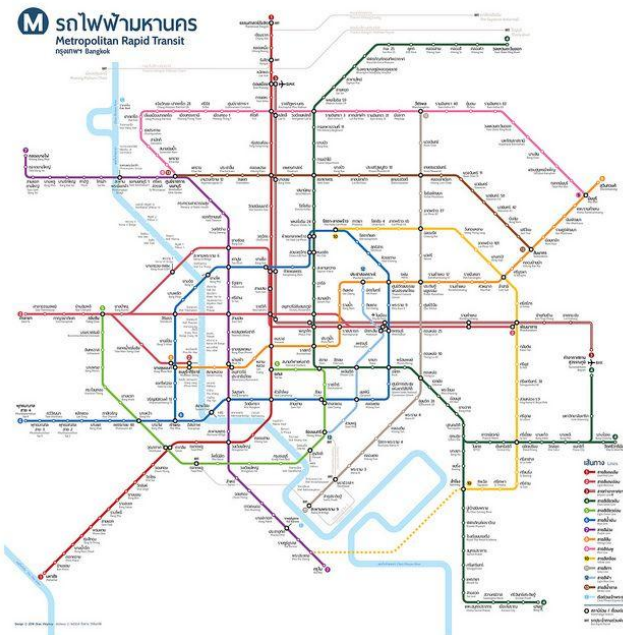


Fig. 2. M-Map (Master Plan) [4]

The most common transportation systems for Bangkok citizens nowadays are metro (BTS, MRT, ARL) and bus (BMTA). The metro is the primary and fastest public transportation mode to get into the inner area from the outer area of Bangkok, whereas the bus is the feeder to feed passengers from the outer area into the metro lines. Chaophraya Riverboat is another option to travel along the Chaophraya river.

B. System Architecture

The proposed architecture is illustrated in Fig. 3. Users can access the system via the app on a mobile device that currently supports both iOS and Android. The backend system is hosted on the cloud, providing the APIs for the frontend to request for routes based on user preferences.

Once a user enters a starting location and a destination, the mobile device makes a request to API Gateway. API Gateway authenticates the request and makes sure that the request comes from an authorized user. If the request is valid, then the system calculates the routes and return recommended routes back to the user.

In order to speed up the process, all possible routes are pre-calculated and stored in the cache. For each request, the system looks up the cache, retrieves, and calculate the final set of routes that fit the user preferences. Since there are

13,510 stops in the database, hence the total number of the possible shortest route would be $13,510^2 = 182M$ routes in the cache.

Since the cache needs to store a large set of key-value data, hence the NoSQL database is the best approach because it offers exceptional performance on a query on big data and can be scaled easily compared to the traditional relational database.

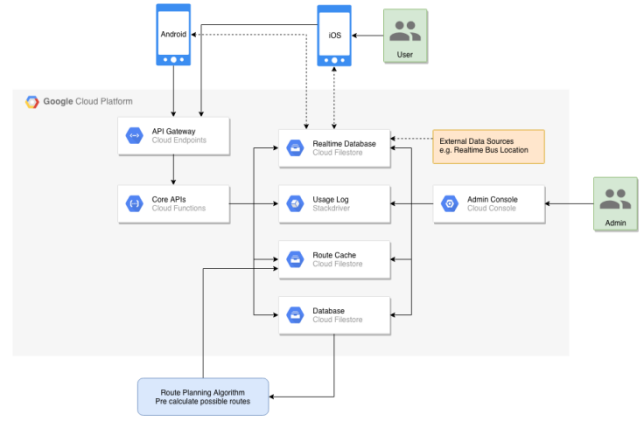


Fig. 3. Overall architecture

The real-time database is used to keep track of the bus locations, which are used in the travel time calculation process.

The system also allows the administrator to see the usage log, including most request routes, the number of users per day. The administrator also provisions the system via the admin console, which allows the administrator to add and update public transportation data in the database — for example, transit route, stations, as well as service interruption and accident.

C. GTFS Data

All transit data are stored in the General Transit Feed Specification (GTFS) file format, which is shown in Table-I below.

Table-I : GTFS data

Data	Filename	# of entries
Transit agencies	agency.txt	12
Exception for the services	calendar_dates.txt	253
Service dates	calendar.txt	12
Fare information	fare_attributes.txt	536,627
Rules to apply fares for itineraries	fare_rules.txt	536,627
Time between trips	frequencies.txt	7,535
Transit routes	routes.txt	1,438
Vehicle travel path	shapes.txt	505,682
Departure and arrival time	stop_times.txt	133,342
Stops where vehicles pick up or drop off riders.	stops.txt	13,510
Trips for each route	trips.txt	3,441

GTFS data used in this research is the part of the Open Data project [5] consolidated by the Office of Transport and Traffic Policy and Planning (OTP), Ministry of Transport, Thailand.

IV. ROUTE RECOMMENDATION ALGORITHM

A. Shortest Path Algorithm

There are several algorithms to find the shortest path. Dijkstra's algorithm [6] [7] is a popular one, but it has a significant drawback on the performance because it selects a vertex closest to the starting point even though that vertex leads in the wrong direction, and that causes the search space very large. The better approach is to use A* [8] algorithm for route finding. A* algorithm selects a vertex closest to the goal by using a heuristic to determine how far from the goal the vertex is. Hence the vertices used in the calculation will be moved toward the goal quickly. At each iteration of its main loop, A* selects the path that minimizes (1)

$$f(n) = g(n) + h(n) \quad (1)$$

Where n is the next node on the path, $g(n)$ is the cost of the path from the start node to n , and $h(n)$ is a heuristic function that estimates the cost of the cheapest path from n to the goal.

Several heuristic functions could be used in the A* algorithm, such as the straight-line distance or Euclidean distance between two points. The advantage of the straight-line distance is calculation speed but is not accurate enough, especially on a global scale due to the curvature of the globe surface. The better approach is to use the Haversine formula [9] (2) as the heuristic function to calculate distance on the surface to the goal.

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right) \quad (2)$$

Where

- φ_1, φ_2 : latitude of point 1 and latitude of point 2 (in radians).
- λ_1, λ_2 : longitude of point 1 and longitude of point 2 (in radians).
- d is the distance between two points along a great circle of the sphere.
- r is the radius of the sphere.

B. Route Scoring Algorithm

The system supports various user preferences, including sorting by travel time, fares, number of transfer, and the "best" mode. The best mode suggests routes based on the score calculated from 4 factors: travel time, fares, number of transfer, and distance. The weights for each factor used in this research are summarized from the survey taken by our focused group. The scoring formula is shown in (3)

$$Score(T, F, C, D) = \left(\left(\frac{T - T_{min}}{T_{max} - T_{min}} \right) \times 0.4 \right) + \left(\left(\frac{F - F_{min}}{F_{max} - F_{min}} \right) \times 0.3 \right) + \left(\left(\frac{C - C_{min}}{C_{max} - C_{min}} \right) \times 0.2 \right) + \left(\left(\frac{D - D_{min}}{D_{max} - D_{min}} \right) \times 0.1 \right) \quad (3)$$

Where

- T is a travel time
- T_{min}, T_{max} is the minimum travel time and maximum travel time of all suggested routes.
- F is a fare
- F_{min}, F_{max} is the minimum fare and maximum fare of all suggested routes.
- C is a number of transfers

- C_{min}, C_{max} is the minimum and maximum number of transfers of all suggested routes
- D is a distance from the source to destination
- D_{min}, D_{max} is the minimum and maximum distance of all suggested routes.

Once the score calculation is completed, the suggested routes are displayed to the user according to the score.

V. USER EXPERIENCE

In this research, we also studied the user experience aspect. The frontend application had been developed for both iOS and Android with these design criteria:

- The layout is designed to fit any size of a mobile screen. Any size of the user's finger must comfortably touch all UI elements, such as a button.
- Fast response, the data must be loaded within 5 seconds for all pages.
- Use font size, styles, colors as a visual cue for more natural to distinguish a piece of information.
- The app supports a place search feature. The search results are ordered by the popularity of the place.

As part of the study, we conducted usability testing from the focus group consisting of 50 people. This feedback was collected from the testing:

- How easy to use the app.
- How easy to find a place.
- How accurate is the suggested route.
- How satisfied with the suggested route.
- Overall satisfaction.

VI. RESULTS

The experimental results are divided into two parts: route calculation result and user experience survey result.

A. Route Calculation Result

The system has produced the results that meet expectations for both place searching and routes suggestion. Fig. 4 illustrates the UI for the user to search for a desired place or destination. The suggested places are shown and ordered by their popularity as soon as the user enters a keyword. Once a place is selected, the system queries the selected place location, current device location, user preferences, and sends all of this information to the backend to calculate possible routes. The list of suggested routes is shown on the screen in Fig. 5.

Fig. 5 illustrates the routes suggestion screen, which allows users to change some user preferences such as mode of transportation and sort option. All suggested routes are listed and ordered, depending on both user preferences.

Each suggested item consists of travel time, fare, walking distance, and steps or list of transportation modes that required a user to take until reaching the destination.

Users can see more details for each route by tapping on a suggested route item. The next screen in Fig. 6 shows the route information in more detail, such as walking steps and more information for bus or train line.

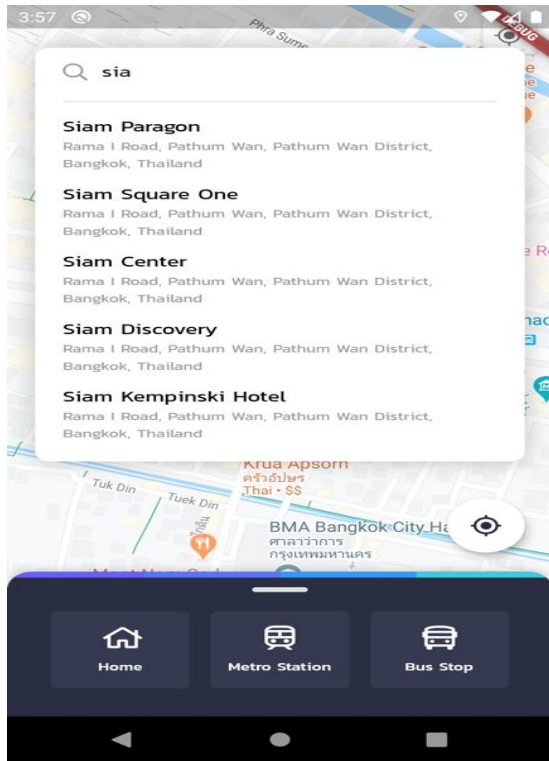


Fig. 4. Place search UI

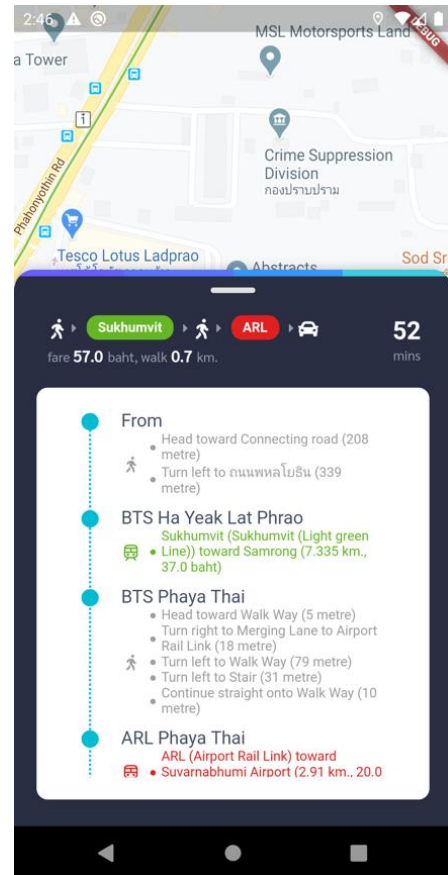


Fig. 6. Route details

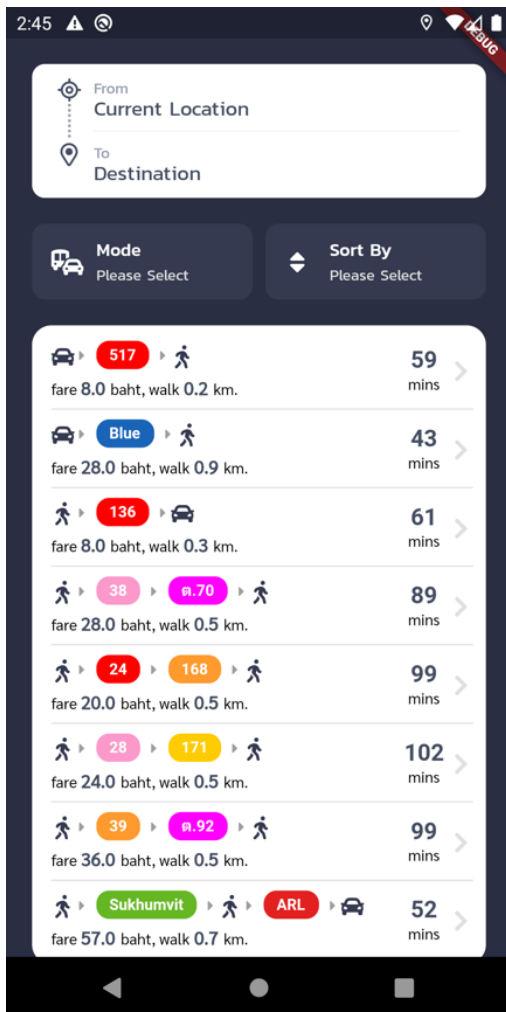


Fig. 5. Routes suggestion result

B. User Experience Survey Result

In this research, we conducted the user experience survey to the focus group, consisting of 50 people. The survey consisted of five questions, and the user gave the rating from 1 (less likely) to 5 (most likely) into each question. Table-II below summarizes the feedback and rating from the survey.

Table-II : User experience survey result

Topics	Rating	Percentage
How easy to use the app.	4.3	86
How easy to find a place.	4.06	81.2
How accurate is the suggested route.	3.84	76.8
How satisfied with the suggested route.	3.94	78.8
Overall satisfaction.	4.24	84.8

VII. CONCLUSION AND FUTURE WORK

This research was done to create the multi-modal route recommender system for public transportation in Bangkok, Thailand, which aims to help and assist passengers so they can find the route easily and quickly, especially when the construction of all metro lines and train lines is completed in the near future.

The result shows that most users are delighted with the system, which achieved 84.8 percent of the overall satisfaction.

For future work, this research can be enhanced in several areas:

- Use a machine learning technique to extract traveling patterns in Bangkok in support of the transit improvement plan in the future.

- Add more data such as nearby restaurants, shops so that a user can search for restaurants and shops along the way.
- Develop an algorithm to suggest a travel plan in Bangkok which specific for each traveler by considering several user criteria such as a period of stay in Bangkok.
- Improve the shortest path algorithm using A.I or machine learning. [10]
- Add an integration to a virtual sightseeing app and website [11]. So that user can request for the suggested routes and direction to the sightseeing location right in those apps.

REFERENCES

1. A. Bozyigit, E. Nasiboglu and S. Utku, "Public Transport Route Recommender Regarding Multiple Factors," in 2018 3rd International Conference on Computer Science and Engineering (UBMK), Sarajevo, 2018.
2. A. Bozyigit, G. Alankuş and E. Nasiboğlu, "Public transport route planning: Modified dijkstra's algorithm," in IEEE, Antalya, Turkey, 2017.
3. W. Almobaideen, M. Saadeh, N. Al-Anbaki, R. Zaghoul and A. Aladwan, "Geographical Route Selection Based on User Public Transportation and Service Preferences," in IEEE, Cambridge, UK, 2015.
4. "Wikipedia Thailand," [Online]. Available: <https://th.wikipedia.org/wiki/รถไฟฟ้าในกรุงเทพมหานครและปริมณฑล>.
5. "Open Data by OTP," [Online]. Available: <https://namtang.otp.go.th/opendata>.
6. A. Javaid, "Understanding Dijkstra Algorithm," SSRN Electronic Journal, 2013.
7. E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, pp. 269-271, 1959.
8. P. E. Hart, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," in IEEE Transactions on Systems Science and Cybernetics , 1968.
9. N. R. Chopde and M. K. Nichat, "Landmark based shortest path detection by using A* and Haversine formula," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, no. 2, pp. 298-302, 2013.
10. A. Singh, V. Shahare, N. Arora and A. ., "Path Finder : An Artificial Intelligence Based Shortest Path," *International Journal of Recent Technology and Engineering*, pp. 5177-5181, 2019.
11. P. Jomsri, "A preliminary development of virtual sightseeing website for thai temples on rattanakosin island," *International Journal of Social, Management, Economics and Business Engineering*, vol. 7, no. 12, 2013.

AUTHORS PROFILE



Kittiya Poonsilp, is the lecturer in the Department of Computer Science, Faculty of Science, Suan-Sunandha Rajabhat University, Bangkok, Thailand. She graduated with a master's degree in the management information system from Walailak University, Thailand in 2006. She is interested in Route Planning Algorithm, Computer Vision, Image Processing, and Knowledge Graph research area.