Asia Pacific Science Press

Journal of Advances in Engineering and Technology
ISSN(O):3005-7507
ISSN(P):3005-7493
Vol. 2 No.4 (2025)

Indoor Pathfinding with the A* Algorithm: A Cross-Platform Mobile Implementation Case

Kongwen Zhang[1]*, Boris Massesa[1], Jingwei Gao[2]

1.School of Computing, University of the Fraser Valley, Abbotsford, British Columbia, Canada

2.College of Computer and Cyber Security, Hebei Normal University, Shijiazhuang, Hebei, 050000, China

Corresponding author: Kongwen Zhang, Frank.Zhang@ufv.ca

Abstract: This study presents the development of a fully integrated mobile module that enables indoor pathfinding functionality from the front end to the back end. The module is implemented using the A* algorithm for route optimization and a NestJS framework with PostgreSQL and PostGIS for spatial data management. Designed as part of the University of the Fraser Valley (UFV) Campus App project, this cross-platform mobile application is built with React Native to ensure seamless usability across devices. Core functionalities include intelligent room search, interactive floor plan visualization, and real-time, turn-by-turn navigation powered by WebSocket communication. The system demonstrates the feasibility of combining spatial databases, efficient routing algorithms, and real-time communication technologies to enhance campus navigation and user experience.

Keywords: Indoor Pathfinding; A* Algorithm; NestJS, Mobile Application

Published: Dec 9, 2025

DOI: https://doi.org/10.62177/jaet.v2i4.899

1.Introduction

Navigating complex university buildings can be challenging for students, faculty, and visitors at the University of the Fraser Valley (UFV). Traditional wayfinding methods such as printed maps, artistic paintings, static signage, and verbal directions are often outdated, unclear, or inaccessible during emergencies [1]. As academic buildings expand and space usage becomes more dynamic, the lack of a reliable digital indoor navigation system increasingly hinders accessibility and operational efficiency. New students and visitors often struggle to locate classrooms, offices, and facilities, which can lead to frustration, reduced productivity, and lower campus satisfaction.

In recent years, the development of indoor navigation systems has gained growing attention due to their potential to enhance accessibility, user experience, and spatial awareness within enclosed environments [2]. Unlike outdoor navigation systems that rely on satellite-based positioning such as GPS, indoor navigation faces unique challenges. These include limited signal penetration, complex building geometries, and multi-level routing constraints [3]. Addressing these challenges requires intelligent algorithms for route optimization, efficient data structures for spatial representation, and an integrated platform capable of real-time user interaction [4].

The A* (A-star) algorithm has been widely recognized as one of the most effective pathfinding algorithms due to its balance between accuracy and computational efficiency [3]. By integrating A* with spatial databases such as PostgreSQL and PostGIS, pathfinding performance and spatial query efficiency can be significantly improved. Moreover, recent advances in web frameworks, such as NestJS, and cross-platform mobile technologies, like React Native, enable the development of scalable, responsive, and user-friendly applications that can operate seamlessly across Android and iOS platforms.

The objective of this project is to design and implement a full-stack, cross-platform indoor navigation module as part of the UFV Campus App [5]. The system incorporates intelligent room search, interactive floor plan visualization, and real-time turn-by-turn navigation using WebSocket communication. The backend, developed with NestJS and PostGIS, manages spatial data and routing operations, while the React Native frontend provides an intuitive interface for end users. This work demonstrates how the integration of modern software frameworks, spatial databases, and pathfinding algorithms can provide an efficient and scalable indoor navigation solution for complex institutional environments.

2.Literature Review

Indoor navigation differs from outdoor routing in important ways: GPS is unreliable indoors, building interiors present complex multi-level geometries, and dynamic obstacles or temporary closures can require online replanning. Surveys of indoor positioning and wayfinding systems summarize these constraints and classify approaches by sensing modality (RF-based, vision-based, sensor fusion), map representation, and routing strategy [6, 7]. These reviews emphasize that an effective indoor navigation solution must couple accurate indoor positioning with efficient pathfinding and a performant spatial data backend.

2.1 Classical and refined shortest-path algorithms (graph search)

The shortest path on graphs is a fundamental building block for routing. Dijkstra's algorithm (1959) [8] provides a shortest path solution for nonnegative edge weights and remains a baseline for network routing. Its worst-case time complexity and exhaustive expansion motivate the use of heuristics for faster single-destination queries in large graphs.

The most popular autonomous agent approach for path finding is A* search [9]. A* (A-star) extends Dijkstra's algorithm by adding an admissible heuristic to guide search toward the goal, drastically reducing explored nodes in many practical cases while preserving optimality when the heuristic is admissible [10]. A* balances exploring the cheapest paths and moving towards the destination, using a simple but effective cost function shown in Equation 1 [8]. A*'s simplicity and proven performance explain its widespread adoption in robotics, games, and GIS-based routing.

$$f(n) = g(n) + h(n) \quad \text{Equation 1}$$

where *g(n)* is the actual cost from the start node to the current node n, and *h(n)* is the heuristic estimate of the cost from n to the destination.

For environments where costs change or sensing updates reveal previously unknown obstacles, incremental replanning algorithms like D* (Dynamic A*) [10] and Focused D* [11] support efficient online repair of previously computed paths rather than recomputing from scratch, an important capability for mobile agents navigating dynamic indoor environments, e.g., temporary blockages, emergency routing.

Several algorithmic refinements improve A*'s runtime or produce more realistic results, such as less grid-constrained routes. Theta* [12] and related any-angle variants produce shorter, more natural paths on grid-based maps by allowing straight-line (non-grid-aligned) transitions between nodes, trading off some theoretical guarantees for better path quality in practice. Jump Point Search (JPS) [13] and its variants prune symmetric or redundant expansions on uniform-cost grids, delivering substantial speedups while preserving optimality on grid maps [14]. These methods are particularly relevant when using rasterized floor plans or occupancy grids as the routing substrate, which is common in robotics and certain indoor navigation implementations.

For high-dimensional motion planning, such as robot arms and mobile robots with kinematic constraints, sampling-based planners like PRM and RRT [15-17] are the dominant approaches. While PRM/RRT are less common for simple pedestrian navigation inside buildings, where 2D graph-based routing suffices, they matter when vehicle dynamics, nonholonomic constraints, or continuous collision-free motion in cluttered 3D spaces are required.

There are some other approaches or different data work reported for indoor pathing finding, such as Qi et al. used building information model [18], Zhou et al. 2022 hierarchical landmark representation [19], Rodenberg et al. 2016 used A* on the point cloud data [20], Gorro et al 2024 used a computer vision You Only Look Once (YOLO) model to avoid obsectlces [21], Khairnar et al 2024 proposed a Adaptive Multi-critieria Indoor Pather (AMIP) algorithm [22], Zhao et al. 2022 used a weighted octree-based 3D approach [23].

2.2 Spatial databases and server-side routing (PostGIS / pgRouting)

For campus-scale systems that must serve many clients and support querying of semantic indoor objects (rooms, amenities), using a geospatial RDBMS is common. PostGIS [23] provides robust spatial types and operations; pgRouting (an extension) implements graph routing functions, such as A*, Dijkstra, bidirectional variants, etc., directly inside the database, enabling server-side shortest-path computation and integration with Geographic Information System (GIS) workflows. Several applied studies demonstrate pgRouting for indoor/outdoor routing and multi-floor adaptations (with caveats around 3D topology handling). Using a PostGIS + pgRouting backend simplifies spatial queries, topology management, and integration with web/service layers (e.g., NestJS).

From the literature, several practical insights emerge for indoor campus routing systems: A remains a solid default* for database-backed routing, especially when you can precompute a compact graph, but specialized variants, such as JPS [14], Theta* [12], Polyanya can reduce latency or improve path realism depending on the map representation [24]. Replanning is essential in dynamic settings, D* variants provide efficient online repairs when edges become blocked or costs change.

Server-side routing with PostGIS/pgRouting is attractive for maintainability, semantic queries, and multi-client scale, but for ultra-low-latency guidance or offline operation, it may be beneficial to cache or precompute routes on the client. Multi-floor 3D routing requires careful topology design (vertical connectors, z-values) because many routing libraries are primarily 2D. Integration with positioning remains a systems challenge: localization noise should be reflected in the guidance logic (e.g., tolerance for corridor vs. room-level instructions), and UX design must handle ambiguity gracefully.

Conventional mobile mapping solutions [25] using SVG or HTML5 Canvas can become inefficient when rendering complex geometric data, especially at interactive frame rates. To address this, the project employs React Native Skia, a high-performance graphics engine that leverages GPU acceleration for real-time rendering. Skia enables the system to smoothly handle floor plans

composed of hundreds of polygonal elements, supporting pan, zoom, and room highlighting without performance degradation.

3.Methodology

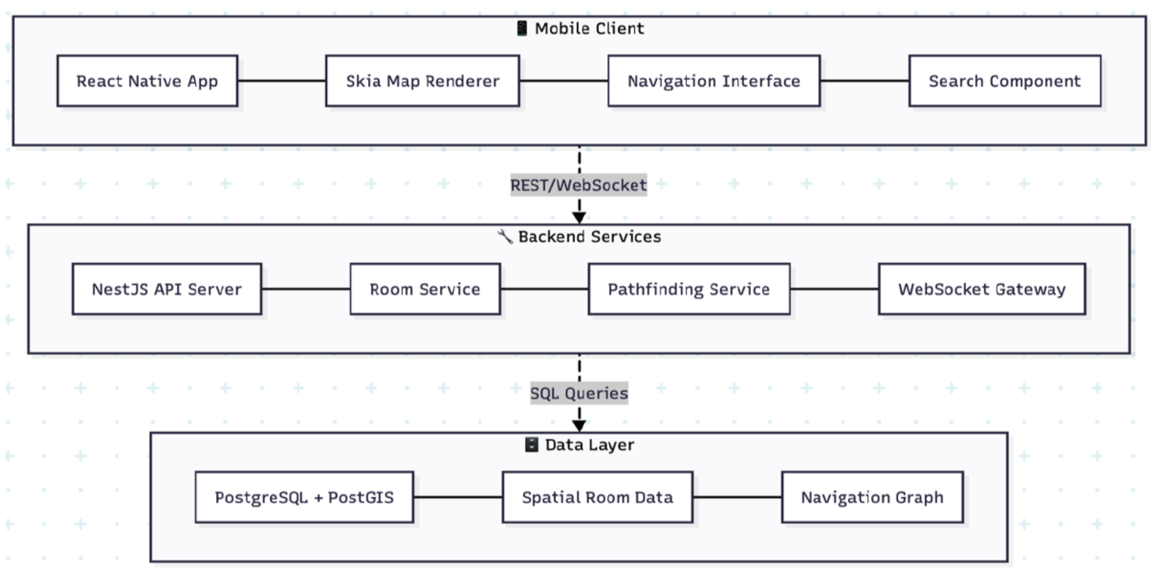
Our work consists of four major phases. (1) Facility map digitization: Using ArcGIS, we digitized the floor plan by creating a centerline representation of each hallway as the path network and defining door center locations as point features (Figure 1). Each door point was assigned a room number to support location-based search. We also conducted polygon closure and intersection checks. (2) System analysis and design: We conducted a comprehensive review of the relevant literature and evaluated existing GitHub projects. Based on this assessment, we selected the A* algorithm as the primary routing method. (3) Software development and system implementation: We developed the UFV App interface and configured a local host environment to manage and serve the backend database. The communication plan is shown in Table 1. (4) Quality assurance and user testing: We completed unit tests using Jest, integration tests using Supertest, and accessibility tests using React native testing library. The UFV App prototype is currently being presented to the internal stakeholders for user evaluation.

Table 1: Communication Patterns

Type	Protocol	Use Case	Performance
Room Search	REST	Autocomplete queries	<100 ms
Map Data	REST	Floor plan geometries	<50 ms
Navigation	WebSocket	Real-time updates	<50 ms

Figure 1: Sample floor plan provided by UFV in the public domain and digitized in our work.

Figure 2: The overall architecture of our system.



4.Results

All functional requirements defined for the UFV indoor navigation system were successfully achieved. The room search module supports fast and intuitive lookup through an autocomplete interface enhanced with category-based filtering and fuzzy string matching, enabling users to locate rooms even with partial or approximate queries. Floor plan rendering operates with high smoothness and visual accuracy, powered by GPU-accelerated Skia rendering that supports seamless zooming and panning. The turn-by-turn navigation feature is fully implemented, delivering real-time routing updates through WebSocket communication and supporting voice-guidance preparation for future enhancements. The application provides verified cross-platform support, operating consistently on both iOS and Android devices. Furthermore, the system meets or exceeds all performance requirements, achieving a stable 60 FPS rendering rate and maintaining navigation query response times under 50 ms, ensuring a reliable, responsive user experience.

The system includes several key architectural and implementation-level accomplishments. The modular software architecture provides clear separation among rendering, routing, and data-access layers, enabling independent maintenance and future extensibility. The spatial database backbone, built on an optimized PostGIS schema, employs targeted spatial indexing to support low-latency navigation queries and is scalable for campus-wide expansion. TypeScript-based shared interfaces reduce integration overhead across components, improving maintainability. Production deployment was streamlined through Docker containerization, enabling consistent server configuration and simplifying updates.

The project also incorporates notable innovations. The adoption of React Native Skia enables GPU-accelerated vector mapping, providing performance and visual fidelity uncommon in mobile indoor navigation systems. The real-time navigation streaming mechanism built on WebSockets allows for continuous route updates and highly dynamic interactions. Additionally, the tight integration of spatial database optimization demonstrates how PostGIS can be effectively adapted for low-latency, mobile-oriented pathfinding.

All the codes are available on GitHub [26].

5.Discussion

During development, several technical challenges were encountered and systematically addressed, contributing valuable insights into mobile indoor navigation design.

One major challenge involved SVG rendering performance. Early prototypes showed limitations in rendering complex vector floor plans using traditional SVG engines. By transitioning to Skia's GPU-accelerated rendering pipeline, we achieved substantial improvements in frame stability and visual accuracy. This highlights the importance of performing early performance benchmarking in mobile applications where GPU constraints vary significantly across devices.

A second issue involved UTM-to-screen coordinate distortions, which arose when mapping georeferenced building data to device-level coordinates. We addressed this by calibrating a precise affine transformation using known building metrics. This effort demonstrated that reliable navigation requires rigorous validation of coordinate systems using real-world reference points rather than relying solely on exported CAD/GIS data.

Ensuring UI responsiveness during WebSocket-driven updates presented another medium-level challenge. Without careful state management, rapid routing updates caused UI blocking and stuttering. By batching updates and employing Reanimated worklets, we achieved smooth animations even under high update frequency. This underscores that real-time mobile applications require dedicated design strategies to avoid overwhelming the rendering thread.

Cross-platform rendering consistency also required extensive testing. While React Native provides shared logic, platform-specific rendering differences—particularly in graphical surfaces and gesture handling—necessitated targeted optimizations. This finding reinforces that achieving consistent design systems on iOS and Android requires platform-aware implementation rather than assuming uniform behavior.

Finally, spatial data accuracy emerged as a critical concern. Errors in room positions, hallway geometry, or door alignment directly degrade navigation accuracy. A validation pipeline with manual verification checkpoints proved essential. This experience confirms that data quality assurance is non-negotiable in navigation applications, especially those intended for institutional use.

6.Conclusion

In summary, this project delivers both technical innovation and meaningful practical impact for the UFV community. Technically, the system pioneers the use of React Native Skia to achieve high-performance, vector-based indoor mapping, incorporates an optimized PostGIS schema with spatial indexing for low-latency navigation queries, and implements a WebSocket-driven routing service capable of providing real-time updates. Practically, the solution enhances campus accessibility and wayfinding by replacing static floor maps with an interactive mobile application, establishes a scalable foundation for campus-wide deployment with minimal infrastructure overhead, and demonstrates the viability of in-house development for specialized indoor navigation systems. The system is production-ready for pilot deployment and is supported by comprehensive documentation to ensure maintainability, scalability, and future feature development. Overall, this work fulfills the CIS 440 capstone requirements by effectively integrating theoretical knowledge with practical implementation to address a real-world institutional need.

Funding

This research is partially supported by the University of the Fraser Valley Scholarly Activity Fund 2025.

Conflict of Interests

The authors declare that there is no conflict of interest regarding the publication of this paper.

Reference

[1] Wu, M., Qiao, L., Wu, Z., Hou, Z., Chen, S., & Lv, G. (2024). What can cartographers learn from artistic paintings when stylizing maps? A preliminary synthesis from the perspective of visual neuroscience. Cartography and Geographic Information Science, 52(1), 35–54. https://doi.org/10.1080/15230406.2024.2370878

[2] Zhang, J., Han, G., Sun, N., & Shu, L. (2017). Path-loss-based fingerprint localization approach for location-based services in indoor environments. IEEE Access, 5, 13756–13769. https://doi.org/10.1109/ACCESS.2017.2728789

[3] Simões, W. C. S., Machado, G. S., Sales, A. M. A., de Lucena, M. M., Jazdi, N., & de Lucena, V. F., Jr. (2020). A review of technologies and techniques for indoor navigation systems for the visually impaired. Sensors (Basel), 20(14), 3935. https://doi.org/10.3390/s20143935

[4] Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, 4(2), 100–107. https://doi.org/10.1109/TSSC.1968.300136

[5] Kamboj, N., & Zhang, K. (2025). Low-cost CCTV repurposing for sustainable parking management: A non-AI computer vision case study. Journal of Advances in Engineering and Technology, 2(4). https://doi.org/10.62177/jaet.v2i4.666

[6] El-Sheimy, N., & Li, Y. (2021). Indoor navigation: State of the art and future trends. Satellite Navigation, 2(7). https://doi.org/10.1186/s43020-021-00041-3

[7] Rahmani, V., & Pelechano, N. (2022). Towards a human-like approach to path finding. Computers & Graphics, 102, 164–174. https://doi.org/10.1016/j.cag.2021.08.020

[8] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. Numerische Mathematik, 1, 269–271. https://doi.org/10.1007/BF01386390

[9] Rachmawati, R., & Gustin, L. (2020). Analysis of Dijkstra's algorithm and A* algorithm in shortest path problem. Journal of Physics: Conference Series, 1566(1), 012061. https://doi.org/10.1088/1742-6596/1566/1/012061

[10] Stentz, A. (1994). Optimal and efficient path planning for partially-known environments. In Proceedings of the IEEE International Conference on Robotics and Automation (Vol. 4, pp. 3310–3317). IEEE.

[11] Stentz, A. (1995). The focussed D* algorithm for real-time replanning. In Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI '95) (pp. 1652–1659). Morgan Kaufmann.

[12] Daniel, K., Nash, A., Koenig, S., & Felner, A. (2010). Theta*: Any-angle path planning on grids. Journal of Artificial Intelligence Research, 39, 533–579. https://doi.org/10.1613/jair.2994

[13] Harabor, D., & Grastien, A. (2011). Online graph pruning for pathfinding on grid maps. In Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (pp. 1118–1123). AAAI Press.

[14] Harabor, D., & Grastien, A. (2012). The JPS pathfinding system. In Proceedings of the Fifth Annual Symposium on Combinatorial Search (Vol. 3, No. 1, pp. 207–208). AAAI Press.

[15] Kavraki, L. E., Švestka, P., Latombe, J.-C., & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation, 12(4), 566–580. https://doi.org/10.1109/70.508439

[16] LaValle, S. M., & Kuffner, J. J. (2001). Randomized kinodynamic planning. The International Journal of Robotics Research, 20(5), 378–400. https://doi.org/10.1177/02783640122067453

[17] Sriramulu, R., Yadav, A., & Pal, S. B. (2025). Fast and efficient indoor navigation: A hybrid pathfinding approach using rapidly-exploring random tree (RRT)-connect and Dijkstra's algorithm. PeerJ Computer Science, 11, e3028. https://doi.

org/10.7717/peerj-cs.3028

[18] Qi, T., Liu, D., Guo, Y., Zhou, X., Zhao, X., Huang, X., & Wang, Z. (2025). Toward efficient and agent-scalable indoor pathfinding: Intelligent navigation-ability-driven indoor map generation using building information model. Architectural Engineering and Design Management, 1–22. https://doi.org/10.1080/17452007.2025.2451817

[19] Zhou, J., Yang, H., Shen, J., & Zhu, L. (2024). Indoor navigation map design based on spatial complexity. Cartography and Geographic Information Science, 52(1), 69–81. https://doi.org/10.1080/15230406.2024.2339296

[20] Rodenberg, O. B. P. M., Verbree, E., & Zlatanova, S. (2016). Indoor A* pathfinding through an octree representation of a point cloud. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2/W1, 249–255. https://doi.org/10.5194/isprs-annals-IV-2-W1-249-2016

[21] Gorro, K., Roble, L., Magana, M. A., & Buot, R. P. (2024). Prototype of an indoor pathfinding application with obstacle detection for the visually impaired. International Journal of Advanced Computer Science and Applications, 15(9). https://doi.org/10.14569/IJACSA.2024.0150987

[22] Khainar, A., Dhaske, A., Patil, A., Dhote, S., & Tamkhade, J. (2024). Adaptive multi-criteria indoor pathfinding algorithm using dynamic user preference and real-time data. Journal for Research in Applied Science and Engineering Technology. https://doi.org/10.22214/ijraset.2024.65564

[23] PostGIS Project Steering Committee. (2024). PostGIS: Spatial and geographic objects for PostgreSQL (Version 3.5.4) [Computer software]. https://postgis.net

[24] Zhou, B., Li, Q., Mao, Q., Tu, W., & Zhang, X. (2015). Activity sequence-based indoor pedestrian localization using smartphones. IEEE Transactions on Human-Machine Systems, 45(5), 562–574. https://doi.org/10.1109/THMS.2014.2368092

[25] Zhao, J., Xu, Q., Zlatanova, S., Liu, L., Ye, C., & Feng, T. (2022). Weighted octree-based 3D indoor pathfinding for multiple locomotion types. International Journal of Applied Earth Observation and Geoinformation, 112, 102900. https://doi.org/10.1016/j.jag.2022.102900

[26] Massesa, B., & Zhang, F. (2025). UFV-Pathfinding: Indoor navigation system for the University of the Fraser Valley [Computer software]. GitHub. https://github.com/borismassesa/UFV-Pathfinding