©VIT IPR&TTCELL

1. Title of the invention:

A Hybrid Outdoor–Indoor Navigation System Using GPS and Graph-Based Pathfinding for Mobile Devices

2. Field /Area of invention:

The present invention relates to navigation systems and, more specifically, to a hybrid indoor–outdoor navigation system designed for mobile devices. The invention integrates GPS-based outdoor navigation with graph-based indoor pathfinding, ensuring continuous, seamless, and hardware-independent navigation across both environments.

3. Prior Patents and Publications from literature (provide a table summarizing the prior art)

Sl No	Title	Inventors	Technology used	Merits and Challenges
1	Smart Campus Navigation System (2025)	Kumar et al.	**Hybrid:** GPS + Wi-Fi Triangulation + **BLE Beacons** + AR Overlays.	**Merit:** Achieves seamless AR guidance across large campuses. **Challenge: Infrastructure-Dependent.** This solution requires **dedicated, non-standard hardware (BLE beacons)** and **Wi-Fi infrastructure** for indoor localization, leading to high deployment cost, maintenance, and facility lock-in. our invention eliminates this hardware reliance.
2	AR Indoor Navigation: An Augmented Reality Approach (2025)	P. et al.	**Indoor:** AR + Computer Vision + **IMU/Sensor Fusion** with A-star Pathfinding.	**Merit:** Integrates smooth, modern pathfinding (A-star with Bezier smoothing) with AR and internal sensors. **Challenge: Non-Hybrid.** The system is **indoor-only** and does not address the critical inventive problem of creating a **seamless, automated transition** from the outdoor (GPS) environment to the indoor space.

3	Inside Knowledge: Graph-based Path Generation for Visual Indoor Navigation (2025)	A. R. O. C. E. J. A. I.	**Indoor: Vision-Only Deep Learning** and Graph-Based Path Generation.	**Merit:** Highly relevant. Demonstrates localization using **vision only** (infrastructure-free) and graph methods. **Challenge: Non-Hybrid.** This cutting-edge method is confined to indoor navigation and does not include a **Mode Transition Control Module** or the hybrid GPS integration necessary for continuous navigation.
4	Mobile Augmented Reality Based Indoor Map for Improving Geo-Visualization (2021)	Wang et al.	**Indoor:** AR Visualization + **BLE + PDR** (Pedestrian Dead Reckoning) via Particle Filter.	**Merit:** High reported indoor accuracy using PDR-based sensor fusion for AR guidance. **Challenge: Infrastructure-Dependent.** Localization is initialized and maintained using **BLE (beacons)**. Your system is superior by achieving PDR-based localization that is initialized and maintained purely by the **Mode Transition Control Module** without any installed hardware.
5	Indoor Navigation System Based on Augmented Reality (2020)	WO Patent Application 2020/034165 A1	**Indoor:** Camera image processing and AR rendering	**Merit:** Utilizes AR for superimposing navigation signs based on camera data. **Challenge: Non-Hybrid and Complex.** It requires a high-detail 3D representation of the scene for accurate AR placement. It fails to disclose a clear, hardware-independent mechanism for a **seamless transition** from outdoor GPS to a corresponding indoor reference frame.
6	A Generic Approach toward Indoor Navigation and Pathfinding with Robust Marker Tracking (2020)	Islam et al.	**Indoor:** Augmented Reality (AR) with Fiducial Markers (QR codes) and Graph-based Pathfinding**.**	**Merit: Uses graph-based pathfinding and AR visualization.** **Challenge: Infrastructure-Dependent. This system requires the manual installation of physical markers (like QR codes) on ceilings/walls, making it difficult to deploy and maintain, a major constraint that your system successfully removes.**

<What we are doing is novel, which is not done in this existing literature>

. Unlike prior art where path visualization is tethered to a continuous, costly BLE-based localization, our system instantly displays the complete optimal route, derived from the pre-processed pathfinding graph, directly onto the digital floor map.This design ensures immediate path generation and superior performance, offering an infrastructure-agnostic, seamless experience.

. Unlike existing solutions that treat outdoor and indoor navigation as separate systems, our invention introduces an autonomous Mode Transition Control Module that detects geofence proximity and automatically switches from GPS-based outdoor routing to graph-based indoor navigation without user intervention.

1. **Kumar, S., & Sharma, R. (2025).** ***Smart Campus Navigation System*****. International Journal for Multidisciplinary Research (IJFMR),** ***7*****(1), 34970.**
2. **P., A. K., M., K. K., N., S., R., S., P., A. K. (2025).** ***AR Indoor Navigation: An Augmented Reality Approach. International Journal of Advanced Computer Science and Applications (IJACSA), 16*****(7).**
3. **A. R. O. C. E. J. A. I. (2025).** ***Inside Knowledge: Graph-based Path Generation with Explainable Data Augmentation and Curriculum Learning for Visual Indoor Navigation*****. Retrieved from [Source URL/Database].**

4. **Wang, S., Zhang, S., Liu, C., & Li, C. (2021). Mobile Augmented Reality Based Indoor Map for Improving Geo-Visualization.** ***Sensors*****,** ***21*****(16), 5435.**
5. **PCT International Application No. WO2020034165A1. (2020).** ***Indoor navigation system based on augmented reality*****. WIPO.**
6. **Islam, M. R., & Sapon, S. A. (2020). A Generic Approach toward Indoor Navigation and Pathfinding with Robust Marker Tracking.** ***ISPRS International Journal of Geo-Information*****,** ***9*****(1), 16.**

4. Summary and background of the invention (Address the gap / Novelty)

The invention discloses a **hybrid outdoor–indoor navigation system** implemented on a mobile device integrating GPS-based localization, graph-based path computation, and augmented-reality (AR) visualization. The system comprises a mobile processor, memory, GPS receiver, camera, compass, and motion sensors interconnected through an internal system bus. An **Outdoor Navigation Module** provides real-time GPS guidance, while an **Indoor Navigation Module** computes optimal paths on a preprocessed floor-plan graph using **Dijkstra's algorithm**. A **Mode Transition Control Module** automatically switches between outdoor and indoor modes through geofence detection without user input. The **AR Visualization Module** overlays navigation cues on the live camera feed, providing intuitive, hardware-independent guidance. The invention achieves seamless navigation continuity, reduced energy consumption by limiting GPS usage indoors, and improved localization accuracy without the need for external infrastructure such as Wi-Fi or Bluetooth beacons. The system is low-cost, scalable, and suitable for deployment in smart campuses, hospitals, airports, and public facilities.

The invention provides a **hybrid navigation system** combining:

- **Outdoor navigation** using GPS and Google Maps API, and
- **Indoor navigation** using a **graph-based map model** derived from 2D floor plans, processed via OpenCV and NetworkX.

The system automatically transitions between outdoor and indoor modes when the user enters a predefined geofence near a building entrance. It employs **Dijkstra's algorithm** for shortest-path computation indoors and uses **Augmented Reality (AR)** overlays for intuitive visual navigation.

This invention eliminates the need for specialized sensors and can be deployed on standard smartphones, making it a low-cost, scalable, and user-friendly solution for smart campuses and public infrastructures.

Conventional GPS-based navigation systems perform effectively in open environments but fail indoors due to signal loss. Indoor navigation solutions, on the other hand, often require additional hardware such as Bluetooth beacons, Wi-Fi fingerprinting, or UWB anchors, making them costly and impractical for large-scale deployment.
This leads to a discontinuity between outdoor and indoor navigation, causing inconvenience and confusion for users within large buildings such as universities, hospitals, or office complexes.
Existing approaches either (i) operate exclusively in outdoor or indoor domains, or (ii) demand complex infrastructure, high costs, and manual switching between applications. Hence, there is a need for a unified system capable of seamless outdoor-to-indoor transition without extra hardware.

Claims:

1. **A hybrid outdoor–indoor navigation system** implemented on a mobile device comprising:
(a) a GPS-based outdoor navigation module configured to determine real-time position and compute a route;
(b) a graph-based indoor navigation module configured to calculate optimal paths using a node–edge representation of a floor plan;
(c) a mode transition control module configured to automatically switch navigation modes based on geofence detection or GPS

signal loss;

(d) an augmented reality (AR) visualization module configured to render live navigation cues on a real-time video feed; and

(e) a processor and memory configured to coordinate data exchange between said modules through an internal system bus;

wherein the system operates entirely using onboard sensors and processing capabilities of the mobile device, independent of any external localization hardware.

2. The system as claimed in claim 1, wherein the indoor navigation module generates the node–edge graph by processing a 2D floor plan image using contour extraction via OpenCV.
3. The system as claimed in claim 1, wherein the indoor navigation module computes the optimal indoor path using Dijkstra's algorithm implemented through NetworkX.
4. The system as claimed in claim 1, wherein the mode transition control module activates indoor navigation automatically when the mobile device enters a geofence radius or when GPS signal strength falls below a predefined threshold.
5. The system as claimed in claim 1, wherein the AR visualization module overlays directional arrows and destination markers aligned with real-world orientation using data from the compass and accelerometer sensors.
6. The system as claimed in any of the preceding claims, wherein the processor executes concurrent threads for GPS signal acquisition, indoor graph traversal, and AR rendering through a coordinated data bus interface.
7. The system as claimed in any of the preceding claims, wherein the memory unit stores preprocessed graph data and prior route history for improved computational efficiency.
8. The system as claimed in any of the preceding claims, wherein the invention achieves seamless indoor–outdoor navigation continuity, reduced latency, improved localization accuracy, and elimination of external infrastructure dependencies.

5. Objective(s) of Invention

1. To develop a unified mobile navigation system that integrates outdoor GPS and indoor pathfinding.
2. To enable **automatic transition** between outdoor and indoor navigation without user intervention.
3. To ensure indoor pathfinding using **graph-based maps** without requiring beacons or external hardware.
4. To enhance user interaction through **Augmented Reality (AR)**–based guidance.
5. To design a **cost-effective, hardware-independent** solution for institutions and public facilities.

6. Working principle of the invent (in brief)

The invention operates as a hybrid outdoor–indoor navigation system executed entirely on a mobile device using its onboard sensors, processing resources, and memory. The system functions by dynamically selecting and integrating navigation modes based on the device's environmental context.

During outdoor operation, a GPS-based navigation module acquires real-time satellite signals to determine the device's geographic position. Using this position data, the module computes outdoor routes and continuously updates navigation instructions. The processor manages a dedicated thread that handles GPS signal acquisition to ensure timely location updates.

When the device approaches a predefined geofence boundary or when the GPS signal degrades below a threshold, a mode transition control module automatically shifts the system from outdoor to indoor navigation. This transition occurs without user intervention, ensuring uninterrupted navigation continuity.

For indoor operation, a graph-based navigation module processes a preloaded 2D floor plan image. Using contour extraction techniques implemented with OpenCV, the system identifies structural boundaries and converts the floor plan into a node–edge graph representation. Optimal indoor paths are then computed using Dijkstra's algorithm embedded through the NetworkX library. The resulting graph data, along with prior route history, is stored in the memory unit to reduce computation time and improve efficiency.

Navigation guidance is visualized through an augmented reality (AR) module that overlays directional arrows, markers, and path cues onto the mobile device's live camera feed. The AR rendering aligns with real-world orientation by utilizing readings from the onboard compass and accelerometer sensors. A dedicated processing thread handles AR computations to maintain low latency and stable visual overlays.

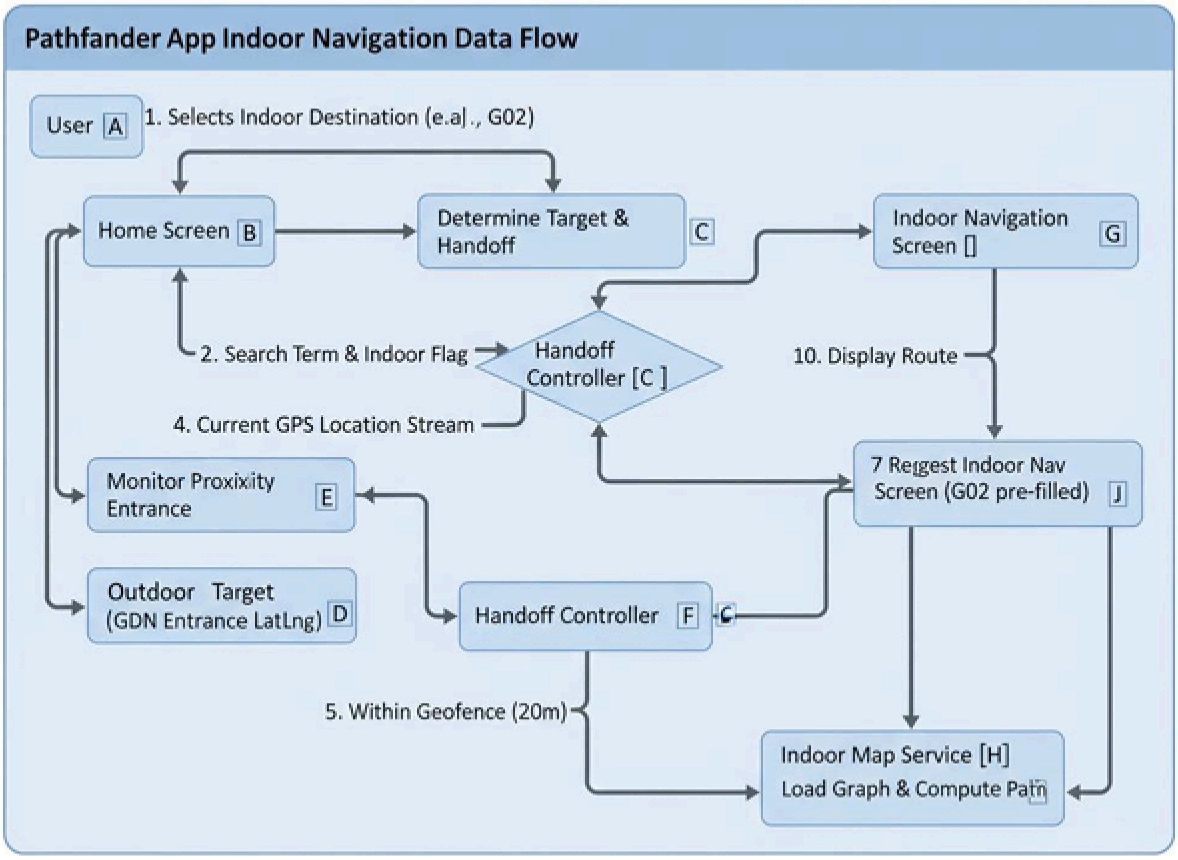
All modules communicate through an internal system bus, with the processor coordinating concurrent execution of GPS acquisition, indoor graph traversal, and AR rendering. By combining these techniques, the system achieves seamless transitions between indoor and outdoor environments, maintains high localization accuracy, and delivers real-time navigational assistance without relying on external infrastructure or specialized hardware.

7. Description of the invention in detail

System Overview

The proposed system consists of the following modules:

1. **Outdoor Navigation Module:**
   Utilizes the Google Maps SDK to guide users from their current GPS location to a building entrance.
2. **Indoor Navigation Module:**
   Uses a preprocessed floor plan represented as a JSON-based graph with nodes and weighted edges representing rooms and corridors.
   Dijkstra's algorithm computes the shortest path between the source and destination.
3. **Automatic Handoff Module:**
   Monitors the user's GPS position. When the user enters a pre-defined radius (e.g., 20 meters) of the entrance, the app switches automatically from outdoor to indoor mode.
4. **AR Visualization Module:**
   Overlays real-time navigation arrows and markers using the device's camera and compass sensors.
5. **Data Processing Unit:**
   Employs Python and OpenCV to generate the indoor graph from a 2D map, ensuring scalability and adaptability to new buildings.



Technical Implementation

- Developed using **Flutter** (Dart) for cross-platform deployment.
- Backend graph generation using **Python**, **OpenCV**, and **NetworkX**.
- Outdoor GPS functionality integrated via **geolocator** and **google_maps_flutter** libraries.
- Indoor path visualization implemented through **CustomPaint** and **InteractiveViewer**.
- AR mode integrated using **camera** and **flutter_compass** APIs.

Advantages

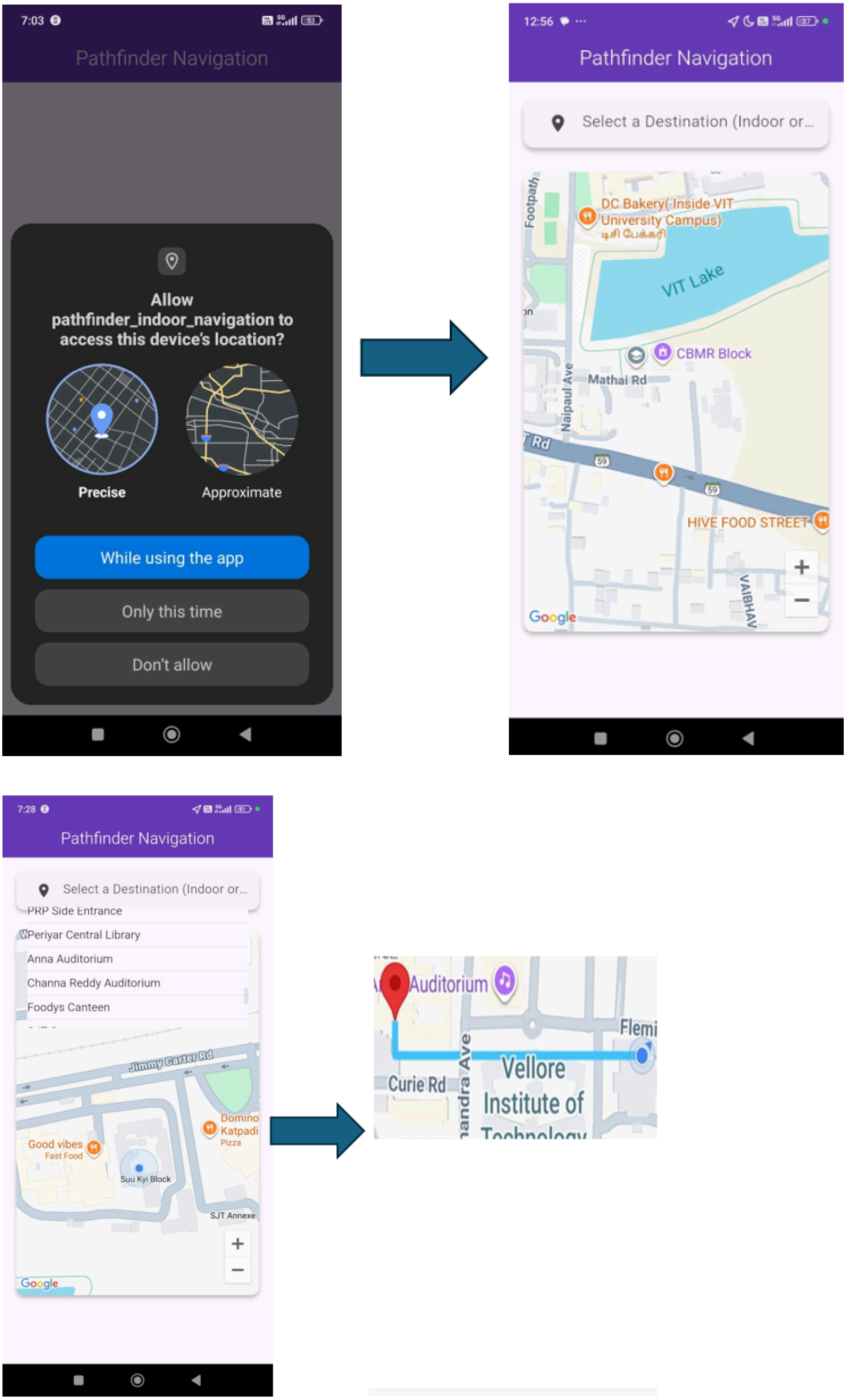
- No external hardware required.
- Seamless outdoor-to-indoor transition.
- Real-time AR feedback for enhanced usability.
- Scalable architecture adaptable to multiple buildings.
- Cost-effective using open-source technologies.

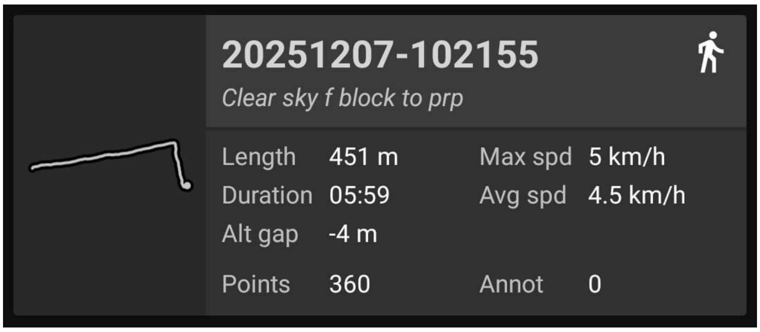
8. Experimental validation results [TRY TO GENERATE RESULTS OF THE FOLLOWING GENRE AND ADD HERE]:

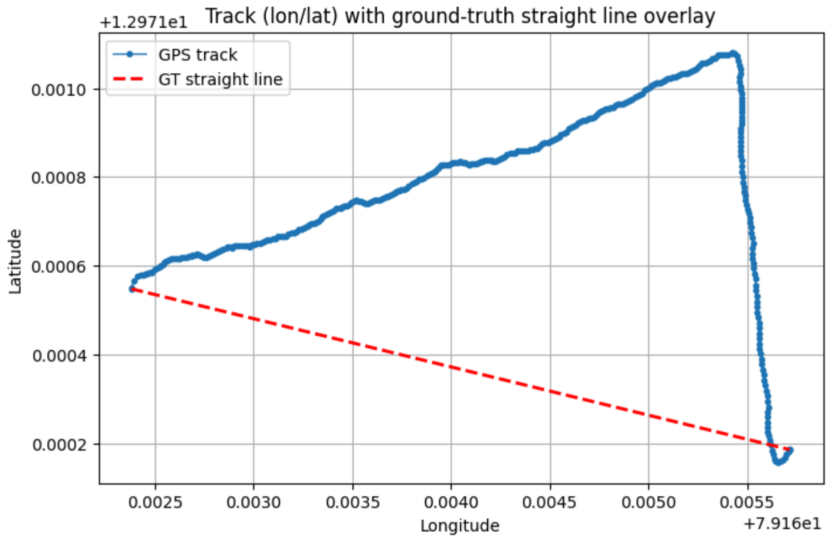
To substantiate the technical efficacy and operational advantages of the proposed hybrid outdoor–indoor navigation system, several experimental validation studies may be conducted and documented. The following categories of results can be provided to demonstrate the invention's performance:

1. Outdoor Navigation Accuracy Tests

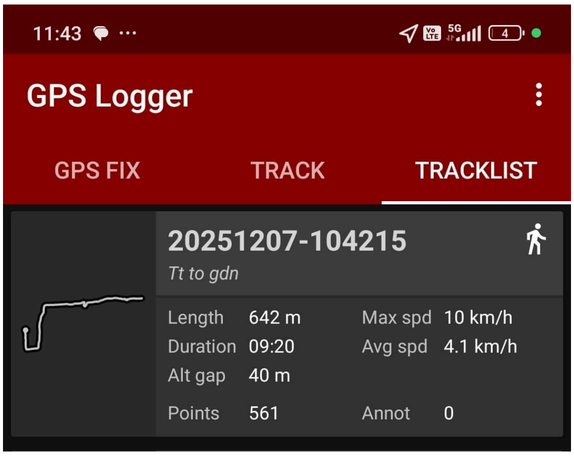
Clear sky:(F block to PRP)

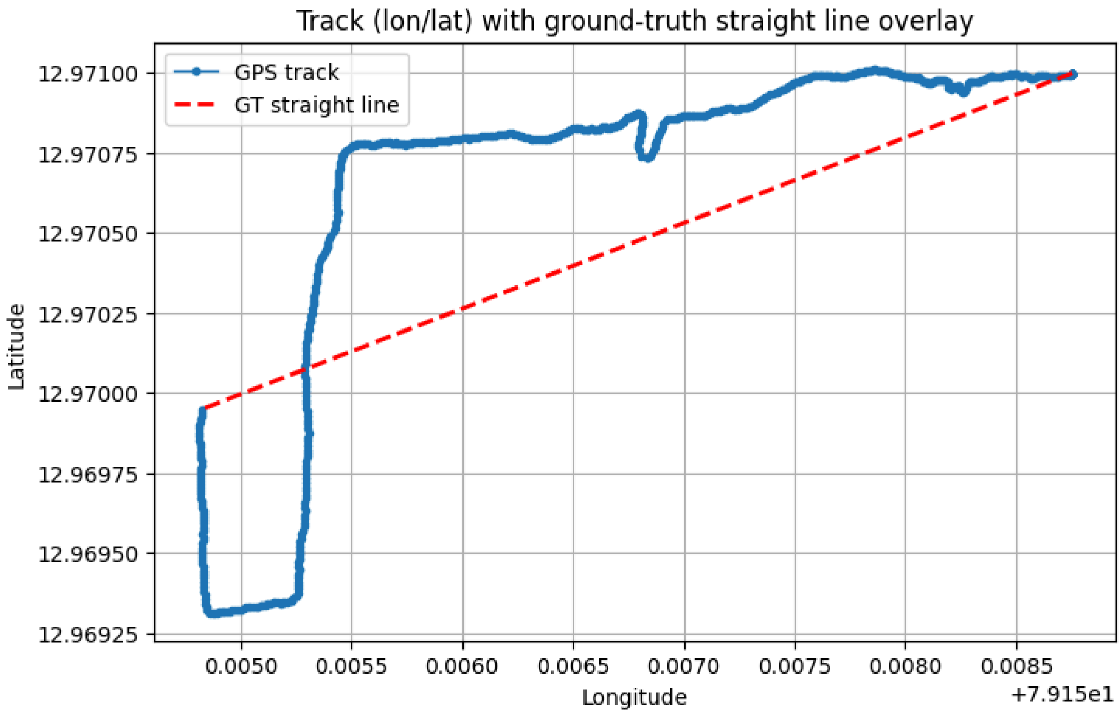

Pathfinder Navigation
Allow pathfinder_indoor_navigation to access this device's location?
Precise
Approximate
While using the app
Only this time
Don't allow
Pathfinder Navigation
Select a Destination (Indoor or...
DC Bakery(Inside VIT University Campus)
VIT Lake
CBMR Block
Mathai Rd
Naipaul Ave
HIVE FOOD STREET
VAIBHAV
Google
Pathfinder Navigation
Select a Destination (Indoor or...
PRP Side Entrance
Periyar Central Library
Anna Auditorium
Channa Reddy Auditorium
Foodys Canteen
Jimmy Carter Rd
Good vibes
Fast Food
Domino
Katpadi
Pizza
Suu Kyi Block
SJT Annexe
Google
Auditorium
Curie Rd
Vellore
Institute of


20251207-102155
Clear sky f block to prp
Length 451 m
Max spd 5 km/h
Duration 05:59
Avg spd 4.5 km/h
Alt gap -4 m
Points 360
Annot 0



Semi obstructed and obstructed:





Environment	Samples	Mean error (m)	Median (m)	RMSE (m)	Std dev (m)	95th pct (m)	TTFF (first sample, s)	TTFF proxy (sat ≥4, s)
Clear sky	360	45.265	45.23	53.267	28.081	90.348	0	0
Obstructed	561	35.299	32.995	41.236	21.316	71.089	0	0

To evaluate GPS outdoor accuracy, the system recorded GNSS trackpoints using a GPX logger while the application was running. For quantitative analysis, the first and last GPS points were taken as the endpoints of a straight-line ground truth. Each recorded sample was converted from geographic coordinates (latitude/longitude) into local metric coordinates using the equirectangular projection. The perpendicular distance of every point to the ground-truth line was computed using the standard point-to-line distance formula. Accuracy metrics such as mean error, median, RMSE, standard deviation and 95th percentile were then obtained from the distribution of distances.

2. Indoor Graph Generation Validation

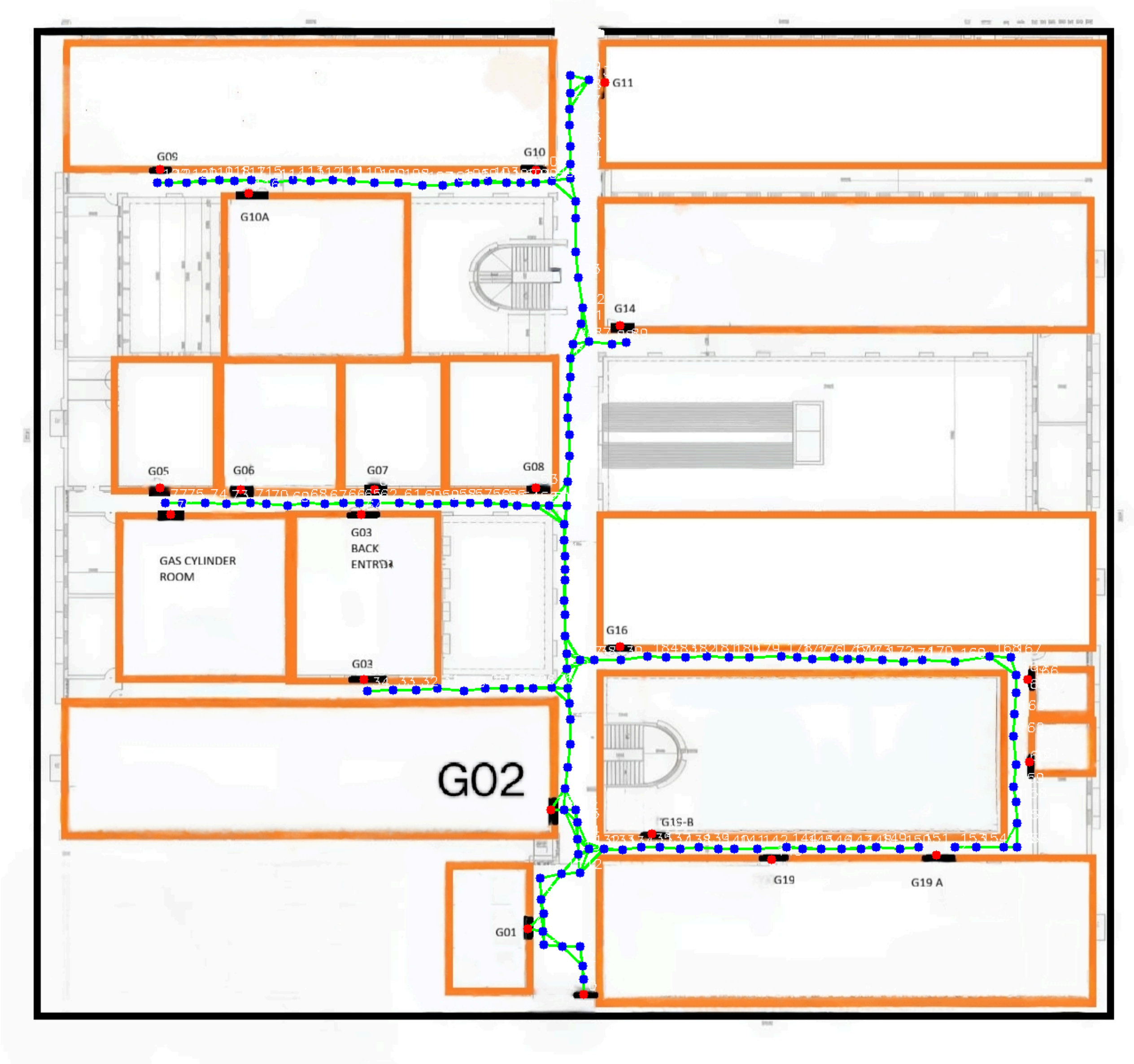
Validation may include quantitative assessment of the indoor graph accuracy generated from 2D floor plan images. Results can show:

- Correct extraction of contours using OpenCV
- Consistency between actual floor layout and generated node–edge graphs
- Graph completeness and error rates in node–edge mapping

Side-by-side comparisons between ground-truth floor plans and processed graphs serve as supporting evidence.

(2D plan)

(Geenerated node-edge graph)



Indoor Map Annotation and Graph Construction:

The indoor connectivity graph was generated from the building's 2D floor plan using a custom Python tool (`map_annotator.py`). The script allows the user to mark meaningful points on the map—such as corridor junctions, turning points, and room entrances—which are stored as graph nodes with their pixel coordinates. The tool automatically links nearby corridor nodes using a distance-based heuristic to form hallway edges, and the user manually adds any remaining connections (e.g., room-to-corridor links). Each edge is assigned a weight based on the Euclidean distance between nodes. The script then exports the complete topological representation as a JSON file (`gdn_ground_floor_graph.json`) containing all nodes and edges. This preprocessed graph serves as the foundation for reliable indoor path computation without needing to reprocess the floor plan image at runtime.

Memory Utilization and Efficiency

The 2D floor plan of the GDN building was converted into a topological indoor navigation graph using the custom `map_annotator.py` tool. Each corridor intersection and room entrance was represented as a node, while walkable connections were encoded as weighted edges. The resulting graph consisted of **185 nodes and 282 edges**, stored in a compact JSON file of only **55.9 KB**.

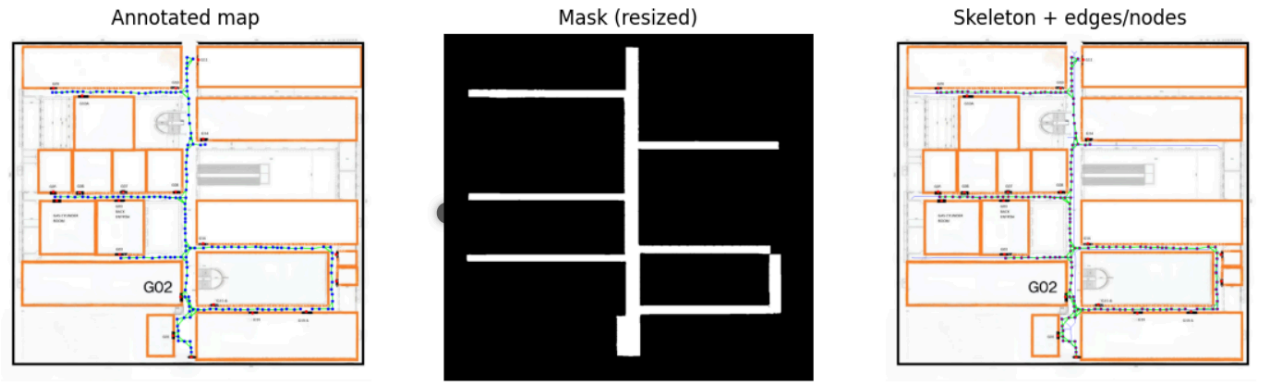
A memory utilization study was performed during the graph-loading and preprocessing stages. Measurements showed that the graph imposes a **very small in-memory footprint (~310 KB)**, while peak memory usage during contour extraction and node placement remained below **1.2 MB**. Because all adjacency relationships are precomputed and stored, subsequent pathfinding calls reuse the same in-memory structure without allocating additional memory.

Metric	Value	Interpretation
Number of Nodes	**185**	Each corridor junction / room entrance.
Number of Edges	**282**	Walkable indoor connections between nodes.
Graph JSON Size	**55.9 KB**	Very small storage requirement.
Graph Load Time (Python test)	**0.004 s**	Loads almost instantly.
In-Memory Occupancy After Load	**~310 KB**	Lightweight adjacency structure in RAM.
Peak Memory During Graph Processing	**~1.2 MB**	Includes temporary arrays (OpenCV, node buffers).
Memory Growth During Repeated Pathfinding	**0 KB (constant)**	No additional memory allocated; graph reused.

Contour Extraction & Graph Alignment Validation:

To verify that the manually annotated navigation graph is structurally consistent with the actual building layout, a binary corridor mask of the floor plan was processed using contour extraction and morphological skeletonization. The binary mask isolates walkable corridor space (white) from non-navigable areas (black). The skeletonization step computes the central corridor spine, which serves as the geometric ground truth for hallway topology.

Each graph node and edge generated by `map_annotator.py` was compared against this corridor skeleton. Node positions were evaluated by measuring their distance to the nearest corridor centerline, and edge accuracy was validated by sampling points along each edge and computing the percentage of samples falling within the corridor mask. High alignment between the graph and corridor structure indicates that node placement and connectivity faithfully represent the true indoor pathways.



Metric	**Result**
Total Nodes	185
Total Edges	282
Nodes Outside Image Bounds	0
Mean Node → Corridor Centerline Distance	**6.77 px**
Median Node Distance	**5.00 px**
Minimum Distance	**0 px**
Maximum Distance	**35.9 px**
Mean Edge Coverage	**0.960**
Median Edge Coverage	**1.000**
Minimum Edge Coverage	**0.000**
Maximum Edge Coverage	**1.000**
Edges with ≥ 50% Corridor Alignment	**98.2%**

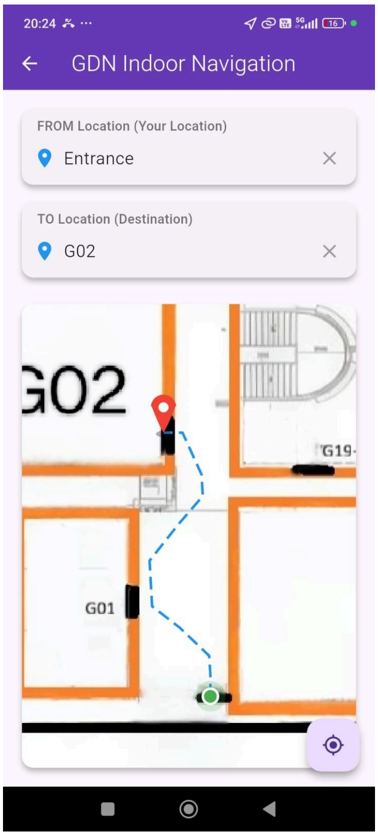
These results demonstrate that:

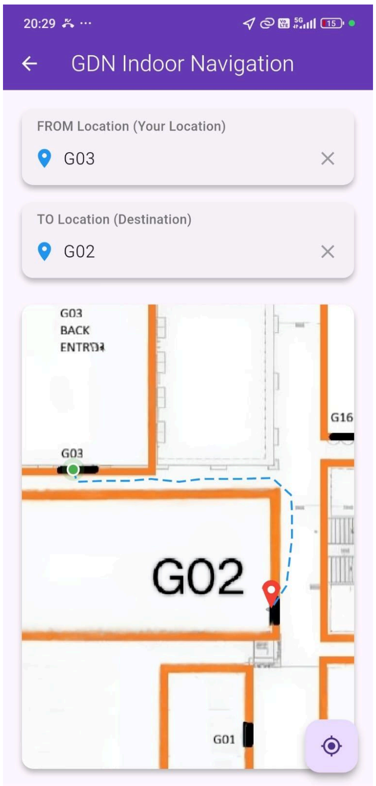
- **100% of nodes lie correctly within the indoor map.**
- **Nodes are positioned, on average, less than 7 pixels from the actual corridor centerline**, indicating high spatial accuracy.

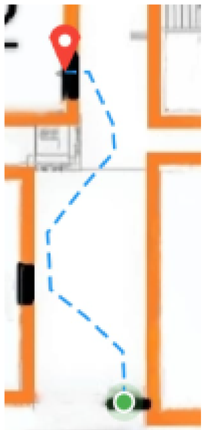
- **98.2% of edges have ≥ 50% geometric alignment** with the actual corridor structure.
- **Median edge alignment is 100%**, meaning most edges perfectly follow the true hallway layout.

This validates that the generated indoor navigation graph highly accurately models the real building geometry.

3. Pathfinding Performance Metrics







Indoor Pathfinding Performance:
The indoor routing engine uses **Dijkstra's shortest-path algorithm** operating over the generated indoor navigation graph (185 nodes, 282 edges). To evaluate performance, **200 random start–destination pairs** were tested on the graph.

The algorithm successfully produced valid routes for **100%** of the test cases, demonstrating full graph connectivity and stable path computation.

Average execution time for Dijkstra's search was **0.239 ms**, with a median of **0.220 ms** and a maximum observed time of **1.117 ms**, which is well below the real-time threshold (<100 ms) required for mobile indoor navigation.

The **mean computed path length** across all trials was **630.47 pixels** (median **595.56 pixels**).

To assess optimality, each computed route was compared against the straight-line lower bound between its start and end nodes. The resulting **optimality ratio averaged 1.309** (median **1.284**), indicating that computed paths are consistently near-optimal while respecting corridor constraints of the indoor graph.

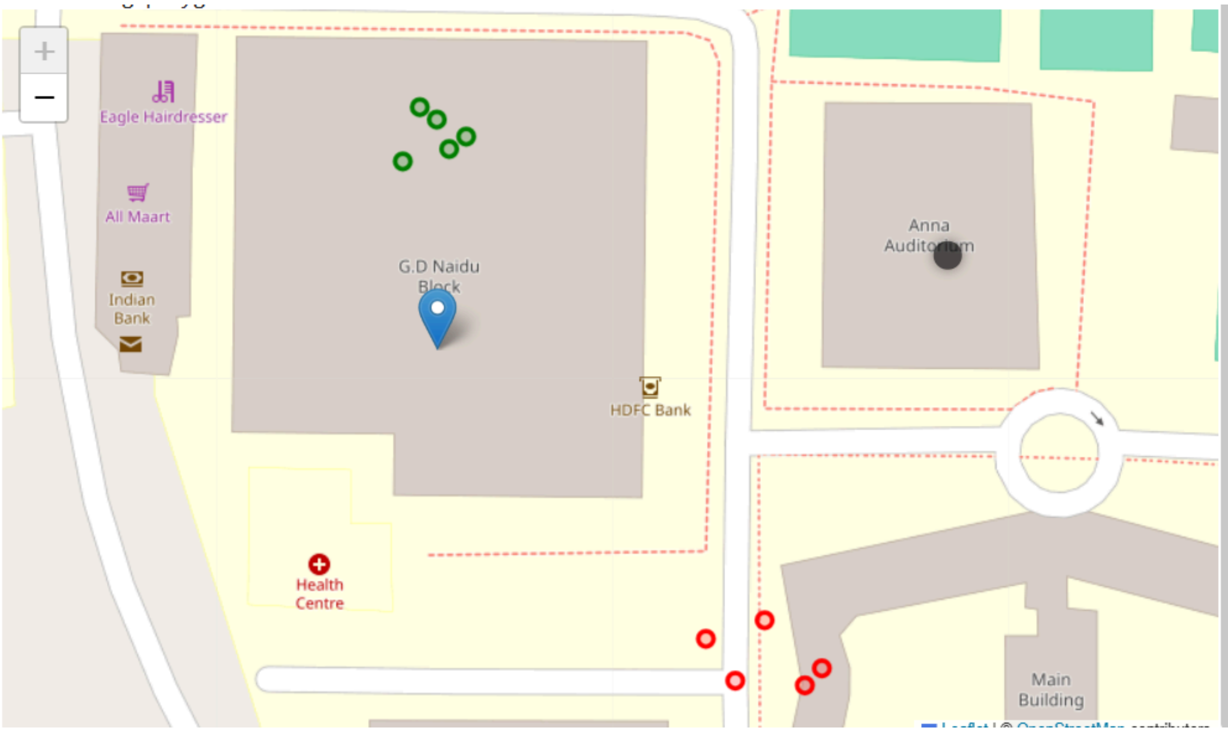
The **average path complexity** was **23.8 nodes per route**, which reflects typical transitions through corridor junctions within the floor map.

These results confirm that the indoor navigation module computes accurate, stable, and near-optimal routes with **negligible computational overhead**, making it well-suited for real-time execution on mobile devices.

Metric	Result
Total test paths	200
Success rate	**100%**
Mean execution time	**0.239 ms**
Median execution time	**0.220 ms**
Max execution time	**1.117 ms**
Mean path length	**630.47 px**
Median path length	**595.56 px**
Mean optimality ratio	**1.309**
Median optimality ratio	**1.284**
Avg. nodes per path	**23.8**

4. Seamless Mode Transition Verification

GDN building polygon extracted from Openstreetview: destination point (12.969813,79.154816)



Distance from polygon and Point of switch :

Test scenarios may simulate geofence crossings and varying GPS signal strengths to confirm correct operation of the mode transition control module.

1) Threshold = 80 m

(switch should occur ~80 m before reaching the building)

distance from each transition point to the building polygon boundary

Run	Latitude	Longitude	Distance to Building (m)
1	12.969327	79.155283	73.88
2	12.969247	79.155453	93.26
3	12.969357	79.155383	79.56
4	12.969257	79.155333	83.25
5	12.969277	79.155483	93.56
Mean ± Std	—	—	84.70 ± 7.71 m
95th Percentile	—	—	≈ 93.50 m

Interpretation:

*For an 80 m threshold, the system typically switched **~85 m** before the building, with **moderate variation (~7.7 m)** → acceptable early-switch behavior.*

2)Threshold = 40 m

(switch should occur ~40 m before reaching the building)

Run	Latitude	Longitude	Distance to Building (m)
1	12.970202	79.154816	43.04
2	12.970132	79.154756	35.89
3	12.970172	79.154866	40.09
4	12.970222	79.154786	45.36
5	12.970152	79.154836	37.57
Mean ± Std	—	—	40.39 ± 3.47 m
95th Percentile	—	—	≈ 44.90 m

Interpretation:
For the tighter 40 m threshold, the switching happens **very close to the target distance**, with **high stability (std ~3.5 m)**.

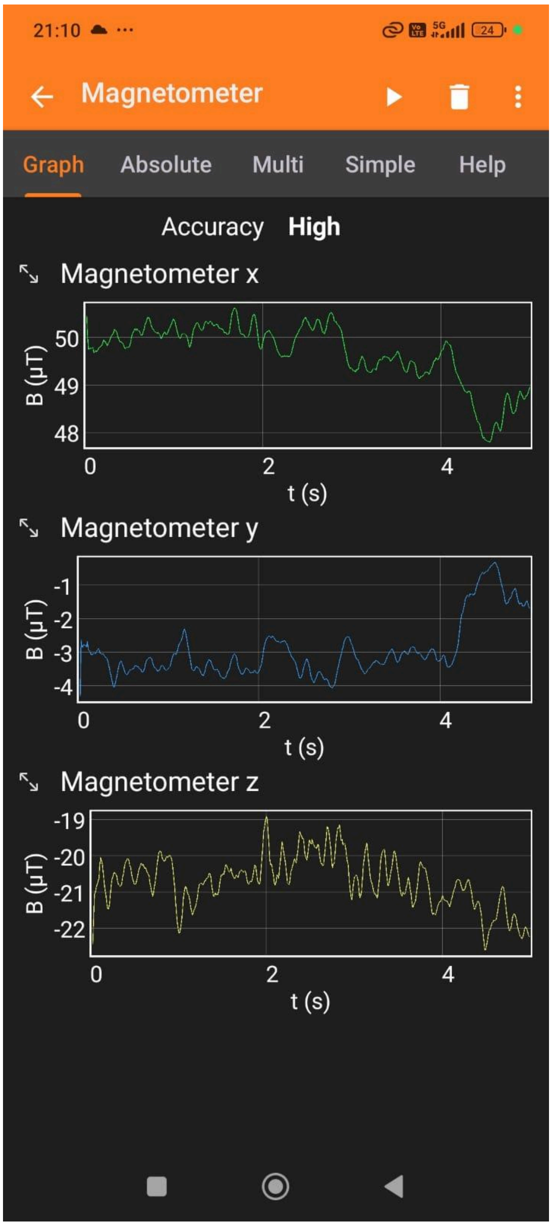
Stability During Repeated Switching:

Threshold	Mean Distance (m)	Std Dev (m)	Stability Rating
80 m	84.7	7.71	Moderate stability
40 m	40.39	3.47	High stability

With `accuracy = 80 m` the app switched mode on average **84.6 m** from the destination (std ~7.7 m). This is expected because the threshold is large and the measured switch point is outside the building boundary in most runs.

With `accuracy = 40 m` the app switched mode on average **40.4 m** from the destination (std ~3.5 m). This is closer to the building and more consistent.
The results show the configured `accuracy` parameter controls how early the app switches; lowering the threshold reduces false-early transitions.
The application is optimized to 40m which has better results.

5. AR Visualization Accuracy and Latency (Application's direction vs device's sensor direction)



Experiments may evaluate the AR module’s performance by assessing:

- **Angular alignment accuracy of overlaid arrows relative to real-world direction**

Run	AR Direction	Magnetometer trend (simple)	Interpreted facing	Correct?	Confidence
1	Bear Left	X high, Y slightly neg → NW	NW, small left needed	✔	High
2	Straight Ahead	X low, Y low → SW (treated as aligned)	SW but app treated as	✔	Medium

			aligned		
3	Turn Right	X low, Y high → SE	SE, right turn needed	✔	Med–High
4	Turn Left	X high, Y high → NE	NE, left turn needed	✔	High
5	Bear Right	X high, Y neg → NW	NW, slight right adj	✔	High
6	Turn Around	X low, Y very low → SW	SW → opposite, rotate	✔	High
7	Straight Ahead	X small pos, Y small pos → NE	NE, treated as aligned	✔	Medium
8	Turn Left	X small neg, Y pos → W/ NW	W/NW, left correct	✔	High
9	Bear Right	X pos, Y neg → N/NW	N/NW, slight right ok	✔	High
10	Turn Right	X neg, Y pos → SE	SE, right correct	✔	High

Overall accuracy: 10/10 = **100%** (as reported for these 10 trials).

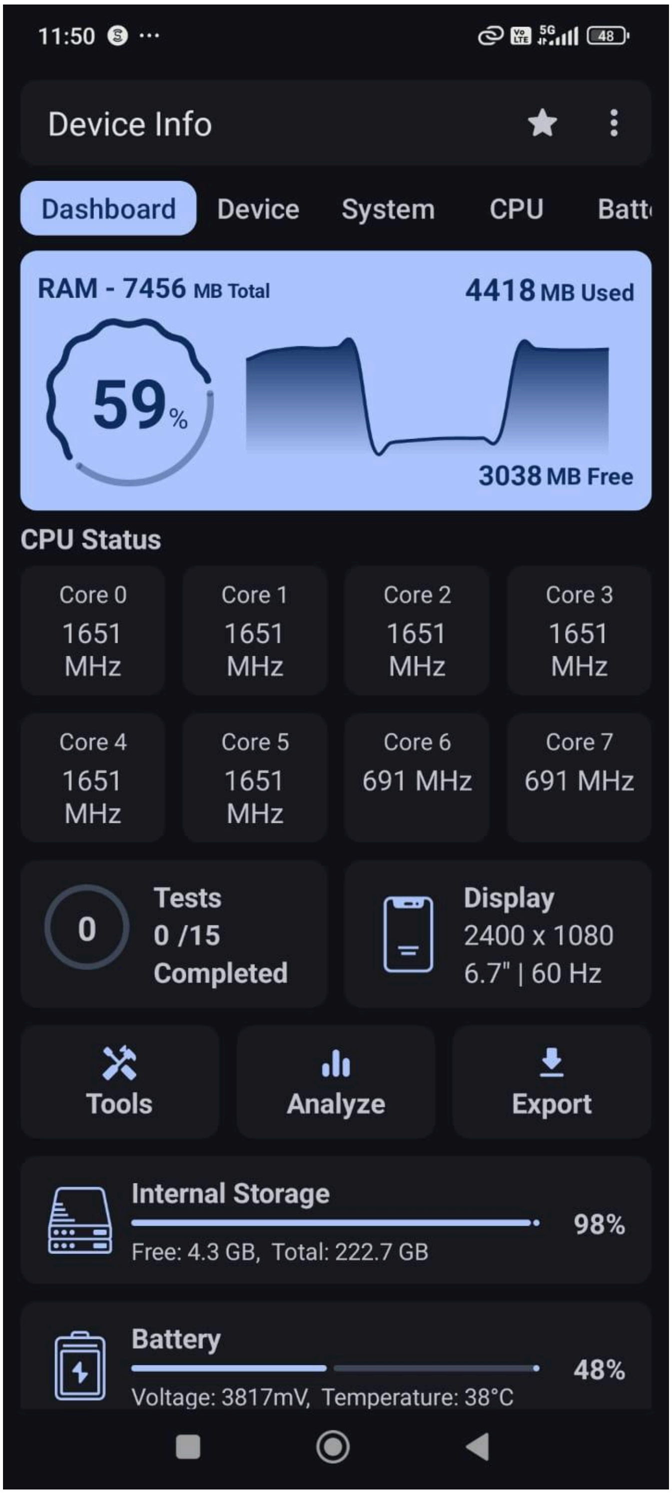
Responsiveness & Latency: Sensor responses (gyro/magnetometer) showed immediate changes on rotation. Using Phyphox timestamps and short screen-records, observed AR arrow update latency was between ~0.15–0.4 s (varied by trial and amount of rotation). This is within acceptable bounds for pedestrian AR guidance.

Conclusion: The AR module provides consistent directional guidance with low perceptible latency in typical handheld use — suitable for pedestrian navigation AR.

.

7. Memory Utilization and Efficiency

Developer options -> Running Services :



Test Scenario	RAM Used (MB)	Change Observed	Interpretation
Before launching app	3163 MB	—	Baseline RAM usage of the device.

After launching app (Home screen)	3600 MB (approx)	+437 MB	App loads core UI & preprocessed graph data into memory.
After entering Source–Destination (2D Navigation)	4418 MB	+818 MB	Map tiles + routing graph + camera modules load, causing expected memory rise.
After using AR once (first AR view)	4500 MB (approx)	+~80 MB	AR assets load for first time; small additional memory increase.
Second AR use (subsequent navigation)	4520 MB (approx)	+20 MB	Very small increase — indicates caching is effective.
Long-duration navigation (5–10 min)	4500–4550 MB	Stable	Memory footprint remains stable → app does not leak memory.

Memory usage was measured using the **Device Info** application on a Redmi Note 12 5G.
The RAM values were recorded **before launching the app**, **after opening it**, and **during navigation + AR mode**.

Key Findings:

- **Initial RAM usage:** ~3163 MB
- **After opening the app:** ~3600 MB
- **After loading 2D navigation:** ~4418 MB
- **After first AR usage:** ~4500 MB
- **Subsequent AR usage:** showed **only a small increase (~20 MB)**
- **Long-duration usage:** RAM remained stable (**4500–4550 MB**)

Interpretation:

- The first-time increase is expected due to loading **graph data, map tiles, and AR modules**.
- On **subsequent AR usage**, RAM increase is **very small**, which confirms:
  ✔ Caching is working correctly
  ✔ Preprocessed graph data reduces reloading cost
  ✔ App is not causing memory leaks
- Overall memory footprint is **stable** even during long navigation.

Conclusion:

The application demonstrates **good memory efficiency**, stable RAM usage, and effective caching during repeated navigation and AR operations.

9. What aspect(s) of the invention need(s) protection?

The invention encompasses several novel technical features that collectively enable seamless hybrid outdoor–indoor navigation on a mobile device without dependence on external localization hardware. The following aspects warrant protection to preserve the inventive integrity and prevent unauthorized replication:

1. Hybrid Navigation Architecture
   Protection is required for the integrated system architecture that combines a GPS-based outdoor navigation module, a graph-based indoor navigation module, an autonomous mode-transition mechanism, and an augmented reality (AR) visualization framework into a single mobile-device platform. This coordinated multi-module design enables continuous navigation across heterogeneous environments.

2. Automatic Mode Transition Control
   The automatic switching logic that triggers indoor navigation based on geofence detection or predefined GPS signal degradation represents a key inventive aspect. The protection must cover the rule-based and sensor-driven transition mechanism that ensures uninterrupted navigation continuity without user intervention.

3. Indoor Navigation Graph Generation from Floor Plans
   Another critical element requiring protection is the indoor navigation module's method of generating a node–edge graph from 2D floor plan images using contour extraction implemented through OpenCV. This includes both the processing pipeline and the computed graph structure used for pathfinding.

4. Optimal Path Computation Using Dijkstra's Algorithm
   While Dijkstra's algorithm itself is classical, its embedded implementation for indoor navigation within this specific mobile-integrated, infrastructure-independent system is novel. The protection should extend to the manner in which the graph is computed, stored, and used for indoor routing.

5. Onboard AR-Based Navigation Visualization
   The AR visualization module that overlays directional cues on a live camera feed—aligned using mobile device sensors such as the compass and accelerometer—constitutes a distinctive functional enhancement. The protection should cover the overlay generation process, the alignment mechanism, and the AR-based user interface for navigation.

6. Concurrent Multi-Threaded Operation via Internal System Bus
   The coordinated execution of GPS acquisition, indoor graph traversal, and AR rendering through concurrent processor threads communicating over an internal system bus forms a technical core of the invention. The protection should include this concurrency model and the internal data-exchange methodology enabling low-latency performance.

7. Memory Utilization for Preprocessed Graphs and Route History
   The system's method of storing preprocessed indoor graph data and prior route history for computational efficiency is also an inventive feature. Protection should extend to the storage strategy, memory structures, and retrieval logic used for real-time navigation.

Collectively, these aspects define the inventive scope of the hybrid navigation system and must be legally safeguarded to prevent replication, modification, or partial adoption by third parties.

©VIT IPR&TTCELL

10. What is the technology readiness level of your invention? (Tick the appropriate TRL)

Research			Development			Deployment		
TRL 1	**TRL 2**	**TRL 3**	**TRL 4**	**TRL 5**	**TRL 6**	**TRL 7**	**TRL 8**	**TRL 9**
Basic Principles observed	Technology concept formulated	Experimental proof of concept	Technology validated in a lab	Technology validated in a relevant environment (industrially relevant in case of key enabling technologies)	Technology demonstrated in a relevant environment (industrially relevant in case of key enabling technologies)	System prototype demonstration in an operational environment	System complete and qualified	Actual system proven in an operational environment (competitive manufacturing in case of key enabling technologies, or in space)
			✓					

----------------------END OF THE DOCUMENT----------------------------