

An Android-Based Campus Navigation System Using GPS, Graph-Based Routing and Direction Overlay

Shaivi Barwe¹, Chandra Parakash Singar², Puja Gupta³, Neha Agrawal⁴,
Deepesh Agrawal⁵

^{1,2,3,4,5}Department of Information Technology, Shri G.S. Institute of Technology and Science, Indore
Indore, India

Abstract:

Campus navigation becomes difficult for new students and visitors due to the large infrastructure and complex layout of educational institutions. Traditional static maps and signboards often fail to provide efficient real-time guidance. This paper presents an Android-based campus navigation system that combines GPS-based location tracking, graph-based shortest path calculation, map visualization, and real-time directional guidance using mobile device sensors. The proposed system is developed using Java in Android Studio and utilizes OpenStreetMap through the OSMDroid library for map rendering. A campus graph containing predefined nodes and paths is used to calculate the optimal route between the user's current position and the selected destination using Dijkstra's shortest path algorithm. The application further integrates device orientation sensors to provide directional arrow guidance on the screen. The system aims to simplify navigation inside the campus environment while maintaining low computational complexity and an easy-to-use interface. Experimental testing demonstrates accurate route generation and smooth user interaction for multiple campus locations.

Keywords: Android Navigation, GPS Tracking, Dijkstra Algorithm, Campus Navigation, Mobile Navigation System.

I. Introduction

Large educational campuses often create navigation challenges for new students, parents, visitors, and faculty members. Finding departments, hostels, laboratories, auditoriums, and administrative buildings using traditional methods such as static signboards or printed maps becomes time-consuming and inefficient. Existing navigation systems mainly focus on city-scale transportation and may not provide optimized guidance for campus-level environments.

With the increasing use of smartphones and mobile sensors, mobile-based navigation systems have become more accessible and effective. Android devices provide location services, orientation sensors, and map integration capabilities that can be utilized to build efficient navigation applications.

The proposed system introduces an Android-based campus navigation application designed specifically for campus environments. The application uses GPS services to obtain the user's current location and a

graph-based shortest path algorithm to calculate the optimal route to the selected destination. The route is displayed on the map interface along with real-time directional guidance using on-screen arrows.

The primary objectives of the proposed system are:

- To provide efficient campus navigation using smartphones.
- To generate shortest paths between source and destination.
- To provide real-time directional guidance.
- To improve user experience through an interactive interface.
- To reduce confusion for first-time visitors inside the campus.

The proposed system combines mobile mapping, routing algorithms, and sensor-based direction assistance to create a lightweight and practical campus navigation solution.

II. Literature Review

Several navigation systems have been developed for urban transportation and indoor positioning. Traditional GPS-based applications such as Google Maps provide navigation for large-scale outdoor environments; however, they are often not optimized for campus-specific navigation.

Research in mobile navigation systems has shown that graph-based routing algorithms such as Dijkstra's algorithm are effective for shortest path calculation in constrained environments. Many systems use node-based path structures where roads and pathways are represented as interconnected graph vertices.

Augmented and sensor-assisted navigation systems have also been explored to improve user interaction. These systems use mobile sensors such as accelerometers and gyroscopes to determine device orientation and provide directional assistance.

Open-source mapping frameworks like OpenStreetMap and OSMDroid have gained popularity in Android-based geographic applications due to their flexibility and reduced dependency on proprietary services.

The proposed system differs from existing approaches by combining:

- Campus-specific graph routing
- Real-time GPS location tracking
- Directional guidance using orientation sensors
- Lightweight Android implementation
- Open-source map integration

This integration provides a practical and cost-effective navigation solution suitable for educational institutions.

III. Proposed System

The proposed system is an Android-based navigation application developed for campus environments. The application assists users in navigating from their current location to a selected destination within the campus.

The system architecture consists of the following major components:

1. User Interface Module
2. GPS Location Module
3. Campus Graph Module
4. Route Calculation Module
5. Direction Overlay Module

6. Map Visualization Module

The user selects a destination from the available campus locations. The application retrieves the current GPS location of the user and identifies the nearest graph node. Dijkstra's shortest path algorithm is then applied to determine the optimal route.

The generated path is displayed using map polylines, while directional arrows guide the user toward the destination using sensor-based orientation detection.

A. System Workflow

The overall workflow of the system is as follows:

1. User opens the application.
2. Current location is obtained using GPS.
3. User selects a destination.
4. Nearest graph node is identified.
5. Shortest path is calculated.
6. Route is displayed on the map.
7. Direction arrows guide the user.
8. Navigation ends when destination is reached.

IV. System Architecture

The proposed architecture integrates location services, routing algorithms, map rendering, and sensor-based directional guidance.

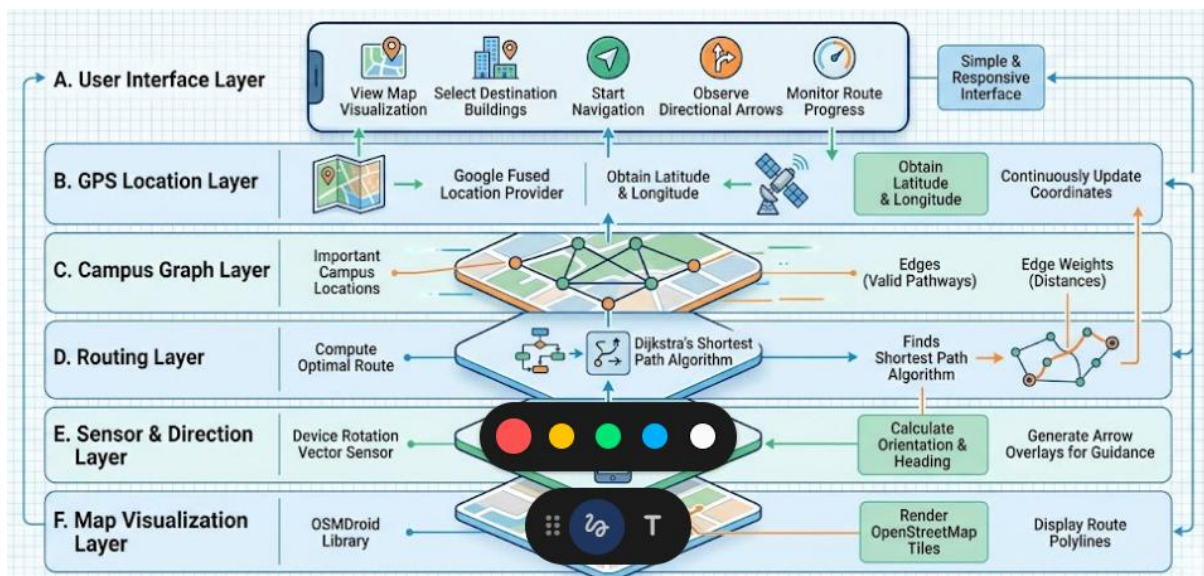


Figure 1: System architecture of proposed system

A. User Interface Layer

The interface allows users to:

- View map visualization
- Select destination buildings
- Start navigation

- Observe directional arrows
- Monitor route progress

The interface is designed to remain simple and responsive for smooth user interaction.

B. GPS Location Layer

The application uses Google's Fused Location Provider to obtain the user's current latitude and longitude coordinates. The location is continuously updated to maintain accurate navigation.

C. Campus Graph Layer

The campus map is represented as a weighted graph where:

- Nodes represent important campus locations.
- Edges represent valid pathways.
- Edge weights represent distances between locations.

This graph structure enables efficient shortest path computation.

D. Routing Layer

The routing layer applies Dijkstra's shortest path algorithm to determine the optimal route between source and destination nodes.

E. Sensor and Direction Layer

The application uses the device rotation vector sensor to calculate orientation and heading direction. Based on the calculated direction, arrow overlays guide the user toward the next navigation point.

F. Map Visualization Layer

The OSMDroid library is used for rendering OpenStreetMap tiles and displaying route polylines.

V. Methodology

A. Location Acquisition

The user's real-time position is obtained using GPS services. The application requests location permissions and continuously receives location updates using the Fused Location Provider API.

The location data includes:

- Latitude
- Longitude
- Accuracy
- Movement updates

This information is used for route calculation and navigation tracking.

B. Campus Data Management

Campus locations are stored as predefined nodes with corresponding coordinates. Each building or landmark is represented using:

- Building name
- Unique identifier
- Latitude
- Longitude

The predefined graph structure allows efficient traversal and route generation.

C. Graph-Based Routing

The navigation system uses Dijkstra's shortest path algorithm for route calculation.

The graph is constructed using:

- Vertices representing campus nodes

- Weighted edges representing distances

The algorithm evaluates all possible paths and selects the route with minimum total distance.

Advantages of Dijkstra's algorithm:

- Guaranteed shortest path
- Efficient for campus-scale navigation
- Low computational complexity
- Suitable for predefined route structures

D. Route Visualization

After shortest path generation, the route coordinates are rendered on the map using polyline overlays.

This allows users to visually track the navigation path.

The route visualization process includes:

- Coordinate extraction
- Polyline generation
- Map overlay rendering
- Real-time updates

E. Direction Overlay System

The direction guidance module uses device orientation sensors.

The system calculates:

- Device heading angle
- Direction toward next node
- Relative angular difference

Based on these calculations, arrow overlays guide the user toward the correct direction.

F. Destination Detection

The application continuously checks the distance between the user and destination coordinates. When the user reaches the target area within a predefined threshold, navigation is completed.

VI. Implementation

The application is implemented using Android Studio with Java programming language.

A. Software Requirements

Component	Technology
Development Platform	Android Studio
Programming Language	Java
Mapping Library	OSMDroid
Location Services	Fused Location Provider
Sensor API	Android Sensor Framework
Operating System	Android

B. Main Modules**1. Main Activity**

Controls:

- User interaction
- Map initialization
- Navigation workflow
- Sensor integration

2. CampusData Module

Stores predefined campus locations and building information.

3. CampusGraph Module

Handles:

- Node connections
- Edge creation
- Shortest path generation

4. Arrow Overlay View

Displays directional arrows according to sensor orientation.

5. Route Rendering Module

Displays path polylines on the map.

C. Sensor Integration

The Android rotation vector sensor is used to determine the orientation of the device.

The system processes:

- Azimuth angle
- Device rotation
- Navigation heading

This enables real-time directional guidance.

D. User Navigation Flow

1. User selects destination.
2. Current GPS location is detected.
3. Nearest node is identified.
4. Shortest route is calculated.
5. Route is displayed.
6. Arrow guidance starts.
7. Navigation ends upon reaching destination.

VII. Results and Discussion

The proposed system was tested on Android smartphones in a campus environment.

The application successfully:

- Obtained real-time GPS coordinates
- Generated shortest routes
- Displayed map-based navigation
- Provided directional guidance
- Detected destination completion

A. Performance Evaluation

Parameter	Result
GPS Tracking	Successful
Route Generation	Accurate
Map Rendering	Stable
Direction Guidance	Functional
User Interface	Responsive

B. Observations

- The graph-based approach reduced route complexity.
- Direction arrows improved user navigation experience.
- OpenStreetMap integration reduced dependency on proprietary map services.
- The application performed smoothly on Android devices.

C. Advantages

- Lightweight implementation
- Easy-to-use interface
- Real-time navigation support
- Efficient shortest path generation
- Cost-effective open-source solution

D. Limitations

- GPS accuracy may vary in crowded areas.
- Indoor navigation is not fully supported.
- The graph requires manual campus mapping.
- Navigation performance depends on device sensors.

VIII. Conclusion and Future Scope

The proposed Android-based campus navigation system provides an effective solution for guiding users inside campus environments. By integrating GPS tracking, graph-based shortest path calculation, map visualization, and directional guidance, the system improves navigation efficiency and user convenience.

The implementation demonstrates that mobile devices can provide practical navigation assistance using lightweight algorithms and open-source technologies.

The system successfully:

- Calculates optimal routes
- Displays navigation paths
- Provides directional guidance
- Enhances campus accessibility

Future Scope

The system can be further improved by integrating:

- Indoor navigation support

- Voice-assisted guidance
- Real-time traffic and crowd analysis
- Dynamic path optimization
- QR-based location identification
- Augmented Reality object rendering
- Multi-floor navigation support
- Cloud database synchronization

These improvements can enhance scalability, accuracy, and user experience.

REFERENCES:

1. Cheliotis, K., Liarokapis, F., Kokla, M., Tomai, E., Pastra, K., Anastopoulou, N., Bezerianou, M., Darra, A. and Kavouras, M., 2023. A systematic review of application development in augmented reality navigation research. *Cartography and geographic information science*, 50(3), pp.249-271.
2. Dijkstra, E.W., 2022. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: his life, work, and legacy* (pp. 287-290).
3. Gupta, P. and Kulkarni, N., 2013. An introduction of soft computing approach over hard computing. *International Journal of Latest Trends in Engineering and Technology (IJLTET)*, 3(1), pp.254-258
4. Gupta, P., Sharma, V. and Varma, S., 2021. People detection and counting using YOLOv3 and SSD models. *Materials Today: Proceedings*
5. Peterson, M.P., 2014. Evaluating mapping APIs. In *Modern Trends in Cartography: Selected Papers of CARTOCON 2014* (pp. 183-197). Cham: Springer International Publishing.
6. Curran, K., Crumlish, J. and Fisher, G., 2013. OpenStreetMap. In *Geographic Information Systems: Concepts, Methodologies, Tools, and Applications* (pp. 540-549). IGI Global Scientific Publishing.
7. Gupta, P., Kulkarni, A. and Sarda, A., 2013. An embedded health care supervisory systems. *International Journal of Latest Trends in Engineering and Technology (IJLTET)*, 3, pp.379-386
8. Chaturvedi, A. and Gupta, P., 2021. The Cloud: Features, Challenges and Scope. *International Journal of Progressive Research in Science and Engineering*, 2(8), pp.466-471.
9. Gupta, P., Varma, S., Arya, N. and Bhagel, R., 2023. Intelligent Security System Based on the Internet of Things (IoT). In *Intelligent Sensor Node-Based Systems* (pp. 177-191). Apple Academic Press
10. Da Forno, R. and Dissler, B., 2013. Oppmap: Opportunistic map and location sharing. *Student Thesis (ETH Zurich)*.
11. Gupta, P., Sharma, V. and Varma, S., 2022, September. An Algorithm for Counting People using Dense Nets and Feature Fusion. In *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)* (pp. 1248-1253). IEEE.
12. Rathore, P., Gupta, P., Jain, S. and Shrivastava, Y., 2022. A Study of the Automated Vehicle Number Plate Recognition System. *i-manager's Journal on Pattern Recognition*, 9(2), p.30.
13. Jain, R., Gupta, P., Bansal, S. and Singh, V., 2022. An AI-Based Online Exam Proctoring Framework. *I-Manager's Journal on Computer Science*, 10(3)

14. Gupta, P., Kumar, S., Singh, Y.B., Singh, P., Sharm, S.K. and Rathore, N.K., 2022. The impact of artificial intelligence on renewable energy systems. *NeuroQuantology*, 20(16), p.5012
15. Gupta, P., Arya, N., Singar, C.P., Chaudhari, A., Singh, U. and Gupta, S., 2025. Safety of Pedestrians in AI-Optimized VANETs for Autonomous Vehicles via Real-Time Vehicle-to-Vehicle Communication. In *AI-Driven Transportation Systems: Real-Time Applications and Related Technologies* (pp. 169-181). Cham: Springer Nature Switzerland.
16. Bardram, J. and Friday, A., 2018. Ubiquitous Computing Systems. In *Ubiquitous computing fundamentals* (pp. 51-108). Chapman and Hall/CRC.
17. Rhead, I., Merabti, M., Mokhtar, H. and Fergus, P., 2008. Worldwide Sensor Web Framework Overview. In *Proceedings of the 9th Annual Postgraduate Symposium, The Convergence of Telecommunications, Networking and Broadcasting*.
18. Hartley, R. and Zisserman, A., 2003. *Multiple view geometry in computer vision*. Cambridge university press.
19. Gupta, P, N. Rathore Comparison of cloud and edge computing based face recognition for security enhancement 2025 *Journal of Discrete Mathematical Sciences and Cryptography*
20. Gupta, P. and Singh, U., 2025. Evaluation of several yolo architecture versions for person detection and counting. *Multimedia Tools and Applications*, pp.1-24.