PROJECT ABSTRACT

Campus Building Navigator Android App

WayPoint Campus is an Android app that answers “where is the CSE block?” and “how do I get there?” with an offline vector campus map, building search and turn-by-turn walking routes.

Project type: Software (Android application)

Availability: Built to order

Suitable branches: Computer / IT, Android

Technologies: Kotlin, Android Studio, On-device vector maps, Dijkstra · A* routing, Fused Location Provider, Room · SQLite, Material Design 3, Git

Prepared by Projectech · projectech.in — for academic project reference.

2. Introduction / Background

Large campuses are genuinely hard to navigate: thousands of newcomers each semester, buildings with similar names, auditorium entrances facing the wrong road, and a campus map PDF — if one exists — that is a blurry scan from years ago. Signboards help only if you can find one.

WayPoint Campus replaces the PDF with a living map: a clean vector rendering of roads, buildings, grounds and gardens that works fully offline, a search box that finds “Seminar Hall 2” faster than asking around, and walking routes from the user's live position with turn-by-turn steps. Building detail screens add floor guides, facilities, open hours and step-free access notes. The campus data lives in one versioned JSON dataset the estate office can update, so the map never goes stale the way the PDF did.

3. Problem Statement

Visitors, new students and even staff lose time to poor wayfinding: missed orientation sessions, late arrivals at interviews, and a constant load of “where is…?” questions on the administration. Commercial map apps cover public roads but not pedestrian campus paths, building entrances, or which floor holds the placement cell.

A campus navigator must work where commercial maps stop: offline (basements and patchy campus networks), pedestrian-first routing with step-free options, and building-level detail including floors and facilities. This project builds exactly that as a self-contained Android app with no map SDK keys and no

data connection required.

4. Objectives

- Render an offline vector campus map (roads, buildings, grounds, gardens) on-device.
- Search buildings, labs and halls with instant filtering and map-pin jumps.
- Compute turn-by-turn walking routes from the user's live GPS position.
- Show building detail: departments, floor guides, facilities, hours, step-free notes.
- Offer a step-free routing preference avoiding stairs and steep paths.
- Keep the campus dataset in one editable, versioned JSON file.

5. Proposed Methodology

The campus is surveyed once into a dataset: building footprints, pedestrian path segments as a graph, road centrelines and points of interest, encoded as versioned JSON with a documented schema (a sample 42-acre campus dataset ships with the build).

The app is built in Kotlin: a custom Canvas/SVG-based vector renderer draws the map fully offline with pan and zoom; a Dijkstra/A* router runs on-device over the pedestrian path graph with edge weights from path length (the step-free mode penalizes stair edges); the Fused Location Provider supplies the blue-dot position; Room caches the dataset and saved places. Testing covers routing correctness on the sample graph, step-free re-routing, offline operation with radios disabled, and GPS accuracy expectations (~5–10 m typical).

6. System Architecture / Working

The architecture is a self-contained offline mobile app. The data layer is the bundled JSON dataset (buildings, path graph, POIs) cached in Room/SQLite with user favourites. The domain layer runs the on-device router (Dijkstra/A*) and the search index. The presentation layer is the custom vector map renderer with pins, the highlighted route overlay, turn-by-turn step cards, building detail sheets and the nearby-buildings list; positioning comes from the Fused Location Provider.

Working flow: (1) the campus survey becomes the versioned dataset; (2) the app renders the vector map fully offline; (3) the user searches or taps a pin and reads the detail sheet (departments, floors, facilities); (4) “Navigate here” routes from the live position over the pedestrian graph; (5) the route draws with turn-by-turn steps naming distances and landmarks; (6) the step-free toggle re-routes around stairs; (7) the estate office edits the single JSON dataset and it ships with the next app update.

7. Hardware / Software Requirements

- Device: Android 8.0 (API 26) or above with GPS; no internet required.
- Data: campus survey encoded in the documented JSON dataset format (sample included).
- Development: Android Studio, Kotlin, Git.

8. Expected Outcomes

- A newcomer can find any building and walk to it with turn-by-turn guidance, offline.
- Building detail screens answer floor, facility and access questions signboards cannot.
- Step-free users get routes that avoid stairs where accessible alternatives are mapped.
- The estate office can keep the map current by editing one dataset file.
- Complete documentation: project report, presentation and viva Q&A.

9. Applications

- University and college campuses for students and visitors.
- Large school campuses and coaching hubs.
- Hospitals and corporate parks with complex layouts.
- Event venues needing temporary wayfinding datasets.

10. Future Scope

- Over-the-air dataset updates without an app release.
- Indoor floor-level positioning with beacons or Wi-Fi RTT.
- Live shuttle/bus positions overlaid on the campus map.
- Augmented-reality walking directions.
- Multi-campus support in one app.