myresearchgo Volume 2, Issue 3, March 2026, ISSN: 3107-3816 (Online)

SJCEM Navigator: A Sensor-Fusion Based Indoor Navigation and Academic Management Framework

Authors

Mr. Om Pradip Patil[1],Mrs. Karishma Tamboli[2],Ms. Simran Verma[3],Mr. Shivlal Yadav[4]
[1234] St. John College of Engineering and Management, Palghar

Abstract- Modern engineering campuses are becoming increasingly complex, often consisting of multiple multi-story buildings that can be confusing for students and visitors. While outdoor navigation is easily handled by GPS, these signals effectively disappear once a user enters a building. This paper introduces "SJCEM Navigator," a mobile solution designed to solve the "last-mile" navigation problem indoors without relying on expensive hardware like Bluetooth beacons. We developed a custom navigation engine using Flutter that fuses data from a smartphone's accelerometer, gyroscope, and magnetometer. By processing this raw sensor data through a Kalman Filter, we achieved stable indoor positioning. Beyond navigation, the system also acts as a centralized academic hub, integrating real-time timetables, faculty tracking, and secure peer-to-peer communication via a Supabase backend. The result is a unified platform that not only guides students to their classrooms but also keeps them connected to their academic life.

Keywords— Indoor Localization, Kalman Filter, Sensor Fusion, Flutter, Smart Campus, Supabase, Mobile Application.

fig 1. Splash Screen

I. INTRODUCTION

Navigating a large college campus should be simple, but for freshers, visitors, or even regular students looking for a specific lab or faculty cabin, it often isn't. Physical signage is static and can be easily missed. While we all have GPS on our phones for driving directions, it becomes useless the moment we step indoors due to signal blockage by concrete walls and ceilings.

The industry has tried to solve this with technologies like Wi-Fi fingerprinting and Bluetooth Low Energy (BLE) beacons. However, these solutions have a major downside: cost. Installing hundreds of beacons across a campus is expensive, and maintaining their batteries is a logistical headache. Wi-Fi fingerprinting, on the other hand, is unreliable because signal strength fluctuates when crowds move through hallways.

We took a different approach. We asked: Can we build a reliable navigation system using only the sensors already present in every student's pocket?

This paper presents **SJCEM Navigator**, an application that uses "Dead Reckoning" enhanced by a **Kalman Filter**. It calculates a user's position based on their steps and direction, correcting for the natural "noise" of smartphone sensors. Simultaneously, we realized that navigation is just one part of the problem. Students also struggle with scattered information—finding timetables, tracking down teachers, or getting study notes. Therefore, we integrated a full Academic Management System

(AMS) into the app, creating a single digital ecosystem for the SJCEM campus.

II. LITERATURE REVIEW & EXISTING SYSTEMS

We looked at several existing methods for indoor tracking before deciding on our approach.

A. Wi-Fi Signal Strength (RSSI) Many systems try to triangulate position based on Wi-Fi signal strength [1]. The logic is simple: the closer you are to a router, the stronger the signal. In practice, however, this is messy. The human body absorbs Wi-Fi signals, and a crowded corridor can drop the signal strength significantly, causing the blue dot on the map to jump around wildly.

B. Bluetooth Beacons BLE Beacons are the gold standard for accuracy [2]. They push notifications when you walk past them. The problem is infrastructure. Covering a 4-story engineering building would require dozens of beacons. For a college project with a limited budget, this wasn't feasible.

C. Pedestrian Dead Reckoning (PDR) This method uses the phone's accelerometer to count steps and the compass to find direction. The issue here is "drift." A small error in direction adds up over time—walking 100 meters might show you 10 meters off-target on the map. This is where our implementation of the **Kalman Filter** comes in, which mathematically smooths out these errors [3].

III. SYSTEM ARCHITECTURE

A. Technology Stack

- Frontend: Flutter 3.0 + Dart (Compiles to native ARM code).
- Backend: Supabase (PostgreSQL) with Row Level Security (RLS).
- Real-time: WebSockets for instant data push.

B. Real-Time Data Flow

1. Teacher Update: A faculty member updates their status (e.g., "In Staff Room").

2. Trigger: Supabase database triggers a change event.

3. Refresh: The student app receives the payload via WebSocket.

4. Recalculation: The UI updates the teacher's location marker on the map
Total Latency: ~80 ms.

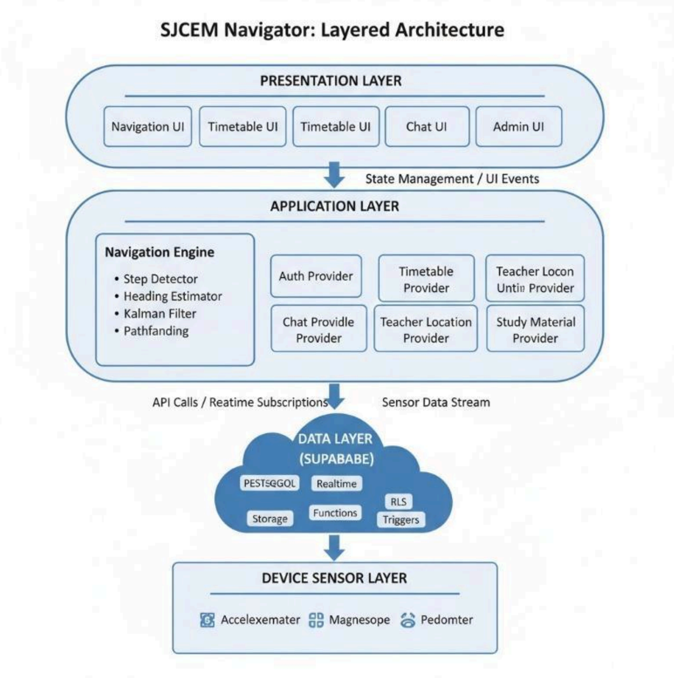

fig 2 Layered Architecture

We built SJCEM Navigator using **Flutter**, which allowed us to write code once and deploy it to both Android and iOS. The system follows the **MVVM (Model-View-ViewModel)** architecture, which keeps our code clean by separating the user interface from the business logic.

C. The "Brains" (Backend)

We moved away from traditional SQL servers and used **Supabase**, a backend-as-a-service built on PostgreSQL. This gave us two massive advantages:

1. **Real-time Sync:** When a teacher updates their location status to "In Staff Room," the database pushes this change to every student's app instantly using WebSockets.

2. **Row Level Security (RLS):** We programmed the database to be paranoid. A student from the Computer Dept cannot accidentally (or intentionally) read the private messages or polls of the Mechanical Dept. The security is baked into the database itself.

D. The "Body" (Frontend & Sensors)

The app uses the sensors_plus package to talk to the phone's hardware. We pull data from:

- **Accelerometer:** To detect the "thud" of a footstep.
- **Magnetometer:** To know where "North" is.
- **Gyroscope:** To detect smooth turns.

IV. METHODOLOGY: THE NAVIGATION ENGINE

This was the most challenging part of the project. Raw sensor data is incredibly noisy. If you look at the raw graph of an accelerometer while walking, it looks like a chaotic scribbling.

A. Step Detection

We implemented a peak-detection algorithm. We look for a sharp spike in vertical acceleration (Z-axis) that crosses a specific threshold. To prevent false positives (like shaking the phone in hand), we added a "timeout" of 300ms between steps—because no human runs fast enough to take two steps in under 0.3 seconds.

B. Mapping Physics to Pixels

We digitized the SJCEM floor plans into a high-resolution coordinate map (1007 * 989 pixels). We calibrated the "Step Length" to be exactly **8.5 pixels**.

Every time a step is confirmed, the app calculates the new coordinates (x, y) using basic trigonometry:

Fig 3 Sensor Fusion Based Navigation Pipeline

$$x_{new} = x_{old} + (8.5 \times \cos(\theta))$$

$$y_{new} = y_{old} + (8.5 \times \sin(\theta))$$

Where θ is the heading (direction) derived from the compass.

fig 4 Maths formula to count Steps

C. The Kalman Filter

To stop the location marker from jittering when the user is standing still, we passed the sensor data through a Linear Kalman Filter.

We tuned the filter with specific values:

- **Process Noise (Q = 0.01):** This tells the filter "Assume the user moves smoothly, not erratically."
- **Measurement Noise (R = 0.1):** This tells the filter "Trust the calculations, but admit the sensors are a bit cheap/noisy."

The result is a smooth, fluid movement of the location marker on the screen, rather than a jumpy one.

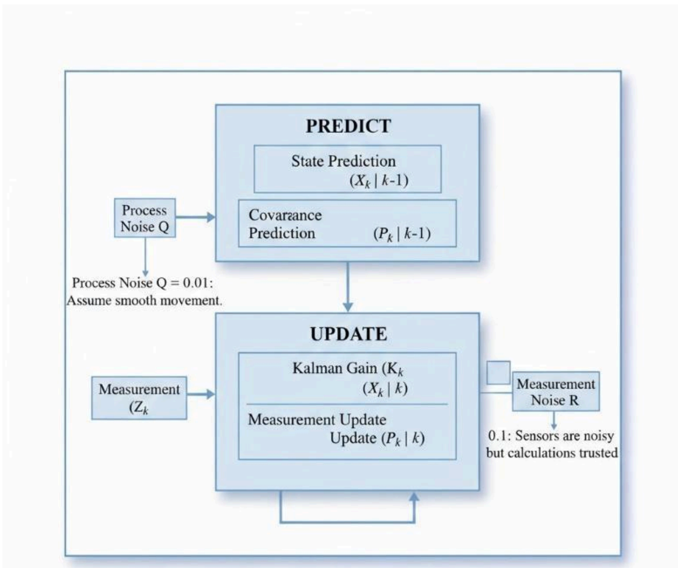

fig 5 Kalman Filter Prediction Update Cycle

V. APP FUNCTIONALITIES

A. Student Features

Indoor Navigation: Search for any room, view multi-floor routing, and see real-time distance/ETA.

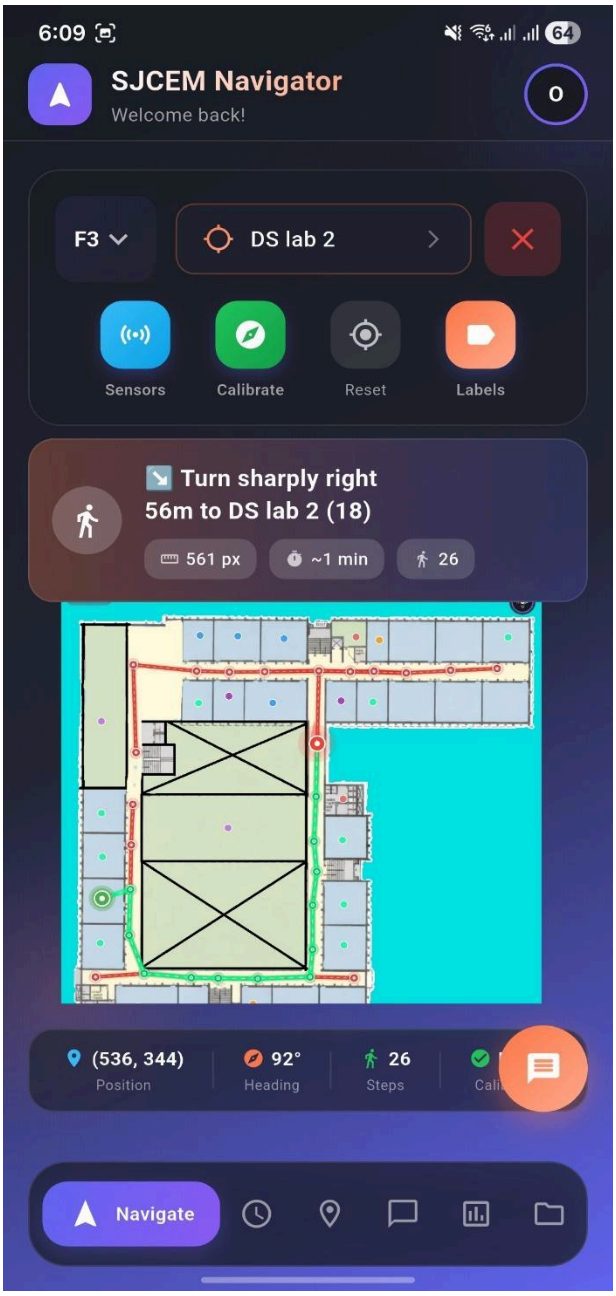

fig 6. Indoor Navigation

Smart Timetable: Shows "Today's View" with the current lecture highlighted. Tapping a subject initiates navigation to that room.

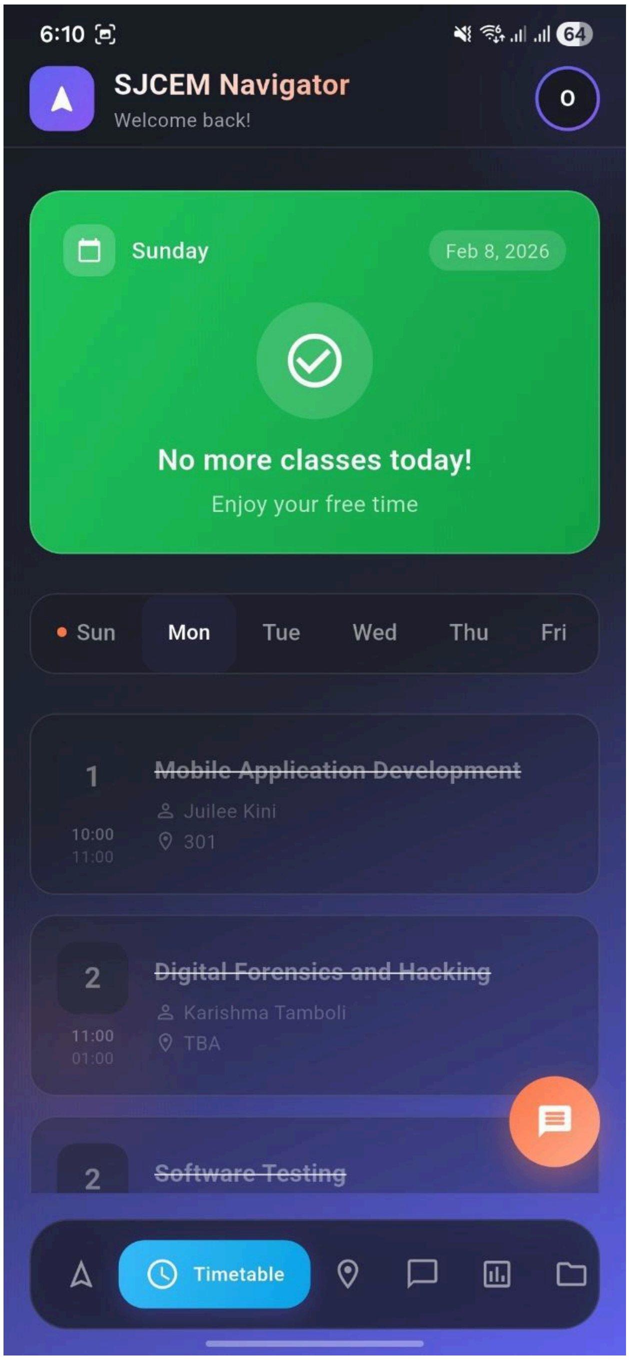

fig 7. Smart Timetable

Teacher Tracking: View current room location, availability status, and one-tap navigation to the teacher.

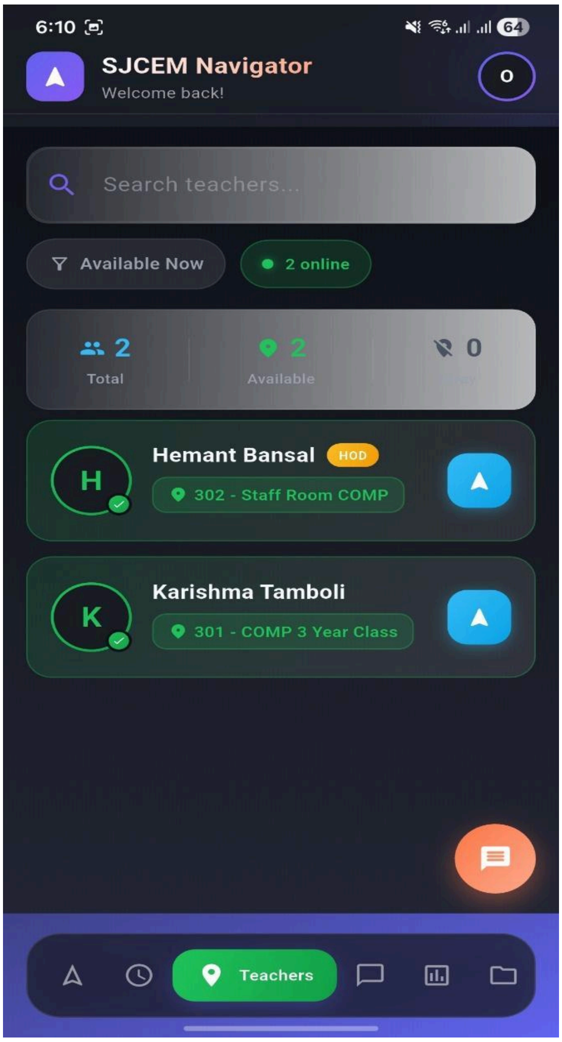

fig 8. Teacher Tracking

Study Material: Organized by Department > Semester > Subject. Files support offline caching with a 50MB size limit.

Communication: Anonymous branch-wise chat, polls, and private messaging with faculty.

B. Teacher Module

- Location status update toggle.
- Upload/Manage study materials.
- Create polls and announcements.
- Reply to student queries.

C. Administrator / HOD

- Timetable creation and editing.
- Room mapping and coordinate management.
- User role assignment and content moderation.

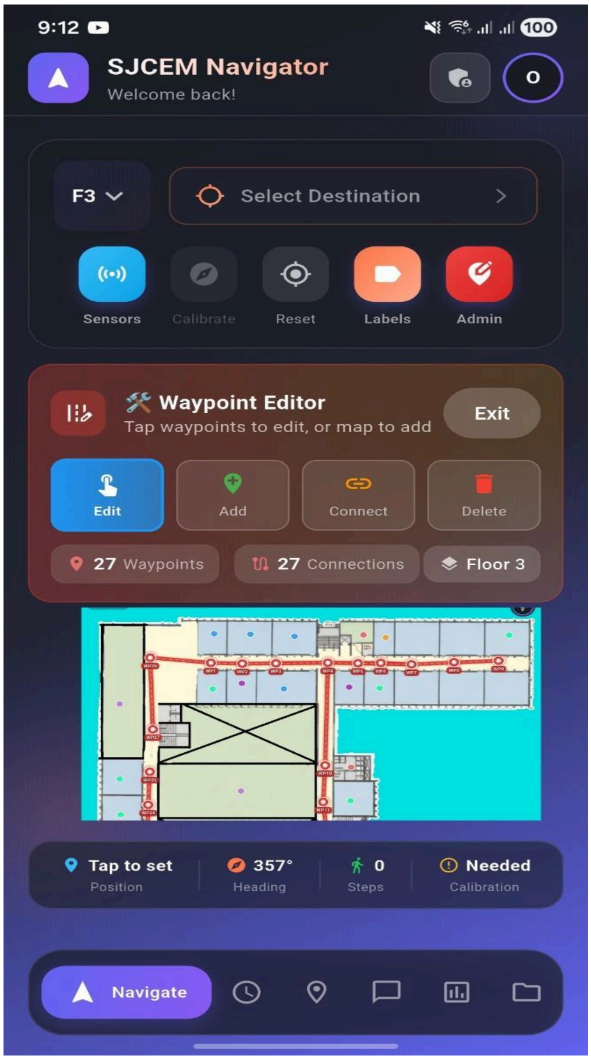

fig 9. Room mapping and coordinate management.

VI. RESULTS AND IMPLEMENTATION

We successfully deployed the application for testing with a group of students from the Third Year Computer Department.

A. Navigation Performance

The navigation module works offline. As shown in **Fig. 1**, the app plots a red path to the destination (e.g., "DS Lab 2"). The turn-by-turn instructions update in real-time. In our tests, the step counter was accurate.

B. Teacher Tracking & Timetable

One of the most used features was the Teacher

Tracker. **Fig. 2** shows the "Available" status. Instead of running to the 3rd floor to check if the HOD is in, students can check the dashboard. The Timetable (**Fig. 3**) auto-loads the day's schedule, highlighting ongoing classes.

C. Academic Resources

We replaced the typical WhatsApp sharing of notes with a structured "Study Materials" section (**Fig. 4**). Files are stored in Supabase Storage buckets, organized by subject, and can be downloaded for offline use.

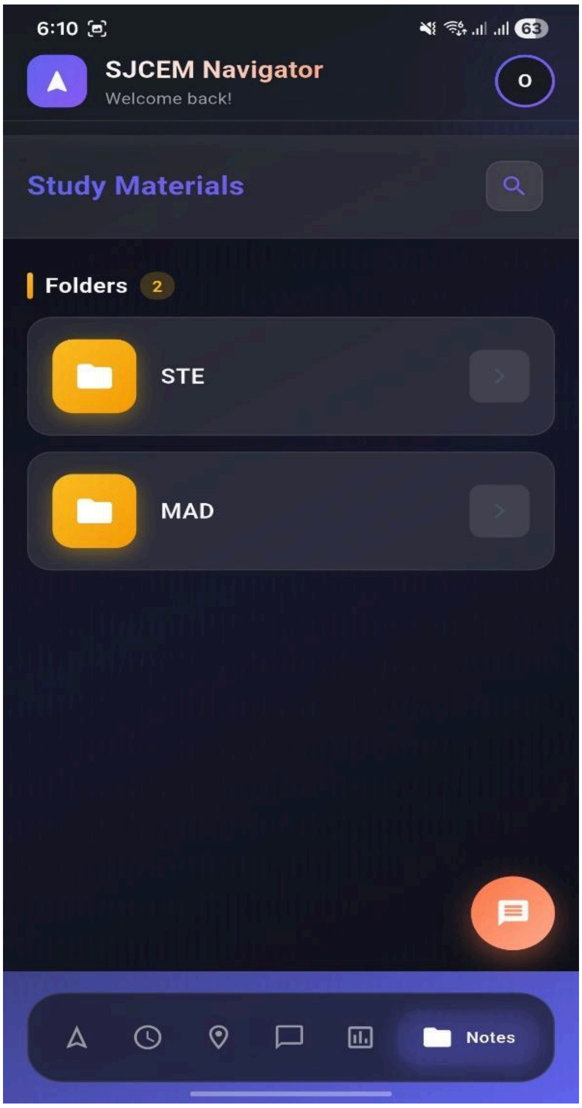

fig 10. Academic Resources

SJCEM Navigator proves that you don't need expensive hardware to build a smart campus. By carefully filtering the noisy data from standard smartphone sensors, we achieved a navigation experience that is accurate enough for daily use. Integrating this with academic tools like timetables and teacher tracking solved a major pain point for students—fragmented information.

The system is currently stable and serves the Computer Engineering department. Our next step is to introduce **Wi-Fi RTT (Round Trip Time)**. This will allow the app to "auto-correct" its position whenever the user passes a known Wi-Fi router, eliminating the small drift errors that accumulate over very long walks.

VII. LIMITATIONS AND ETHICS

A. Limitations

- **Magnetic Distortion:** Accuracy drops near elevators or heavy electrical equipment.
- **Carrying Style:** PDR accuracy varies if the phone is held in hand vs. in a pocket.
- **Calibration:** Requires initial setup of map coordinates.

B. Ethical Considerations

- **Teacher Privacy:** Location tracking is strictly opt-in; teachers can disable visibility at any time.
- **Data Security:** All data is protected by PostgreSQL Row Level Security (RLS) policies.
- **Institutional Consent:** Developed with the consent of SJCEM administration.

VIII. REFERENCES

[1] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," *IEEE INFOCOM*, 2000.

[2] R. Faragher and R. Harle, "Location Fingerprinting With Bluetooth Low Energy Beacons," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, 2015.

[3] G. Welch, "An Introduction to the Kalman Filter," University of North Carolina, Tech. Rep. TR 95-041, 2006.

[4] "Flutter State Management," *Flutter.dev Documentation*, 2024. [Online].