**Live Tracking Implementation Plan – Parrot Analyzer**

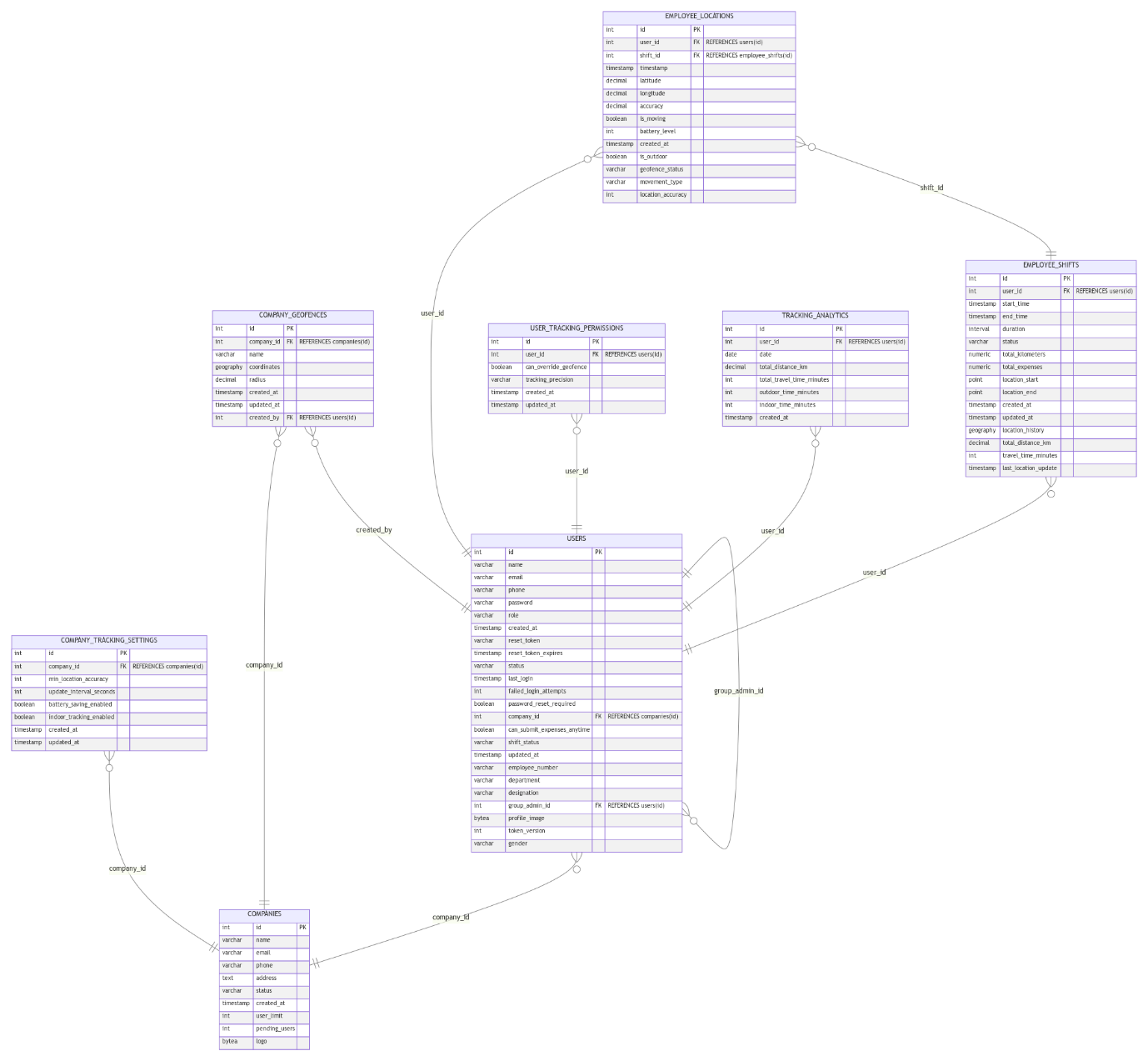
**1. Introduction & Overview**

* **Objective:**
  + Enable real-time tracking of employee locations.
  + Calculate travel metrics and enforce geofencing rules.
  + Provide Group Admins with a live monitoring dashboard.
* **Key Benefits:**
  + Enhanced operational oversight.
  + Improved safety and compliance.
  + Real-time analytics and actionable alerts.

**2. Database Schema Updates**

* **New Tables:**
  + **employee\_locations:**
    - Stores live location data (latitude, longitude, timestamp, accuracy, movement status, battery level, etc.).
    - Includes indexing for faster querying.
  + **company\_geofences:**
    - Defines geofence areas with spatial data (using PostGIS for polygon support).
    - Includes a spatial index for efficient geo-queries.
  + **user\_tracking\_permissions:**
    - Manages individual tracking settings (e.g., override permissions, tracking precision).
  + **company\_tracking\_settings:**
    - Global company-level settings (location accuracy, update intervals, battery saving, etc.).
  + **tracking\_analytics:**
    - Aggregates travel metrics such as distance, travel time, and indoor/outdoor durations.
* **Schema Modifications:**
  + Updates to **employee\_shifts** (adding columns for location history, total distance, travel time, and last update).

**Database Schema:**

****

**3. Backend Implementation**

* **Real-Time Communication:**
  + **Socket.IO Integration:**
    - Implements real-time broadcasting of location updates.
    - Utilizes caching (via Redis) to reduce latency and a retry mechanism to handle failures.
    - Middleware for user authentication.
* **API Endpoints:**
  + **Location Management:**
    - POST /api/location/update – Accepts live updates.
    - GET /api/location/current/:userId – Retrieves current location.
    - GET /api/location/history/:userId – Retrieves historical location data.
    - GET /api/admin/locations/active – Retrieves active location data for admins.
  + **Geofencing:**
    - Endpoints for creating, updating, deleting, and validating geofences.
  + **Shift Management:**
    - Endpoints for shift start/end, including location validation and permission checks.
* **Services:**
  + **LocationService, GeofenceService, and DistanceCalculationService** to encapsulate business logic.
  + Additional utility services for notification delivery and location validation.

**4. Frontend Implementation**

* **Employee App Components:**
  + **LocationTracker.tsx:** For continuous background tracking.
  + **GeofenceAlert.tsx:** For real-time geofence entry/exit notifications.
  + **LocationPermissionHandler.tsx:** To manage permissions.
* **Group Admin Dashboard:**
  + **LiveMap.tsx:** Displays real-time employee locations.
  + **EmployeeLocationList.tsx:** Lists employee statuses.
  + **GeofenceEditor.tsx & TrackingMetrics.tsx:** For managing geofences and analytics.
* **Management Dashboard:**
  + Components for managing employee permissions and viewing geofence-related data.
* **Hooks and Contexts:**
  + Custom hooks like useLocationTracking, useGeofencing, and useTrackingPermissions to manage state and interactions.

**5. Mobile-Specific Implementation**

* **Background Location Tracking:**
  + Utilizes Expo Location to support background tasks.
  + Implements native geofencing for entry/exit detection.
  + Battery optimization strategies (adaptive polling and intelligent updates).

**6. Integration Points**

* **Database Integration:**
  + Use of PostGIS functions for spatial queries and distance calculations.
* **Real-Time Updates:**
  + Socket.IO for broadcasting and Redis for caching.
* **Data Structures:**
  + Efficient indexing and data batching for performance.

**7. Security Considerations**

* **Data Protection:**
  + Encryption of data in transit and at rest.
  + Strict access control measures.
  + Audit logging for access to sensitive data.
* **Privacy Controls:**
  + Options for employees to adjust tracking precision.
  + Transparent data usage and retention policies.

**8. Performance Optimization**

* **Backend:**
  + Batch processing of location data.
  + Optimized spatial queries using indexes.
* **Frontend:**
  + Map clustering and WebSocket data compression.
  + Efficient rendering for handling 100+ markers.

**9. Testing Strategy**

* **Unit Tests:**
  + Validate individual components (e.g., location and geofence validators).
* **Integration Tests:**
  + End-to-end testing of Socket.IO communications and API endpoints.
* **Performance Tests:**
  + Load testing for high volume concurrent updates.
* **Security Tests:**
  + Tests for unauthorized access and data encryption.

**10. Notification System**

* **Push Notifications:**
  + Sends alerts for geofence violations and prolonged stationary statuses.
  + Uses different notification types (e.g., GEOFENCE\_VIOLATION, EMPLOYEE\_STATIONARY) to trigger admin alerts.

**11. Deployment Strategy**

* **Database Migration:**
  + Scripts to create new tables and modify existing ones.
  + Ensures correct ordering based on dependencies.
* **Service Deployment:**
  + Docker configurations for Socket.IO and API services.
  + Separate configurations for mobile app builds.
* **Rollback Plan:**
  + Scripts for both database and service rollback to restore previous versions if needed.

**12. Documentation & Training**

* **Technical Documentation:**
  + API specs, database schema details, and Socket event documentation.
* **User Documentation:**
  + Guides for the employee app and admin dashboards.
  + Troubleshooting and privacy policy updates.

**13. Timeline & Risk Mitigation**

* **Week 1:** Database schema updates, backend foundation, and real-time location tracking (Socket.IO).
* **Week 2:** Geofencing, location history, frontend components, and map integration.
* **Week 3:** Mobile tracking, notifications, testing, optimization, and deployment.

**Risk Mitigation:**

* + Battery drain management, data privacy compliance, performance optimizations, and offline support strategies.

**14. Shift Management and Permissions**

* **Shift Validation:**
  + API endpoints for starting/ending shifts with location verification.
  + Backend service (ShiftPermissionService) that checks permissions and geofence compliance.
* **Management Interface:**
  + UI components for updating employee permissions and monitoring shift statuses.