**Thiết Kế Kiến Trúc Hệ Thống SSB 1.0 – Smart School Bus Tracking**

**1. Tổng Quan Kiến Trúc**

Hệ thống Smart School Bus Tracking (SSB 1.0) được thiết kế để quản lý và giám sát xe đưa đón học sinh, đảm bảo an toàn, minh bạch và hiệu quả cho trường học, tài xế và phụ huynh. Hệ thống sử dụng kiến trúc microservices kết hợp với event-driven architecture để đáp ứng yêu cầu theo dõi thời gian thực (độ trễ tối đa 3 giây) cho ít nhất 300 xe hoạt động đồng thời. Giao diện người dùng hỗ trợ tiếng Việt (có thể mở rộng sang tiếng Anh) và được triển khai trên cả web dashboard và ứng dụng mobile (PWA).

Hệ thống bao gồm ba tầng chính:

* Tầng Giao Diện (Client Layer): Web Dashboard cho quản trị viên, ứng dụng PWA cho tài xế và phụ huynh, hỗ trợ cả Android/iOS.
* Tầng Ứng Dụng (Application Layer): Các microservices xử lý nghiệp vụ cốt lõi, tích hợp WebSocket để truyền dữ liệu thời gian thực.
* Tầng Dữ Liệu (Data Layer): Kết hợp cơ sở dữ liệu quan hệ (PostgreSQL), dữ liệu thời gian thực (TimescaleDB), và cache (Redis) để đảm bảo hiệu suất và lưu trữ.

Hệ thống sử dụng RESTful API cho các giao tiếp đồng bộ và WebSocket cho truyền dữ liệu thời gian thực. Dữ liệu vị trí từ thiết bị GPS được gửi qua Ingest API, xử lý và phát đến client qua Redis pub/sub.

**2. Sơ Đồ Kiến Trúc**

Dưới đây là sơ đồ kiến trúc tổng quan của hệ thống SSB 1.0, mô tả bằng văn bản :

[Tầng Giao Diện - Client Layer]

Web Dashboard (Quản trị viên): Giao diện quản lý tổng quan

PWA (Tài xế): Ứng dụng web tiến bộ cho tài xế

PWA (Phụ huynh): Ứng dụng web tiến bộ cho phụ huynh

Ứng dụng Mobile (Android/iOS, tùy chọn): Hỗ trợ tích hợp

[Tầng Ứng Dụng - Application Layer]

Cổng API (API Gateway - Nginx): Phân phối yêu cầu

Dịch vụ Xác thực (Authentication Service): JWT, OAuth2

Dịch vụ Quản lý Người dùng & Thiết bị (User & Device Management)

Dịch vụ Quản lý Học sinh & Phụ huynh (Student & Parent Management)

Dịch vụ Quản lý Xe & Tài xế (Bus & Driver Management)

Dịch vụ Quản lý Tuyến đường & Điểm dừng (Route & Stop Management)

Dịch vụ Lập lịch & Phân công (Scheduling & Assignment)

Dịch vụ Theo dõi Thời gian thực (Real-time Tracking - WebSocket)

Dịch vụ Thông báo & Cảnh báo (Notification & Alert)

Dịch vụ Báo cáo & Phân tích (Reporting & Analytics)

[Tầng Dữ Liệu - Data Layer]

PostgreSQL: Lưu trữ dữ liệu có cấu trúc

TimescaleDB: Lưu trữ dữ liệu GPS theo thời gian

Redis: Cache và pub/sub cho dữ liệu thời gian thực

[Tích Hợp Bên Ngoài - External Integration]

Thiết bị GPS: Gửi dữ liệu qua Ingest API

Dịch vụ Thông báo Bên ngoài: SMS (Twilio), Email (SendGrid)

Luồng Dữ Liệu Chính

1. Tương tác người dùng:
   * Quản trị viên sử dụng Web Dashboard để quản lý xe, tài xế, học sinh, tuyến đường và lịch trình.
   * Tài xế sử dụng PWA để xem lịch làm việc, danh sách học sinh, và cập nhật trạng thái đón/trả.
   * Phụ huynh sử dụng PWA để theo dõi vị trí xe, nhận thông báo khi xe đến gần hoặc bị trễ.
   * Yêu cầu từ người dùng được gửi đến Cổng API (API Gateway).
2. Xử lý yêu cầu:
   * Cổng API (Nginx) chuyển tiếp yêu cầu đến các microservices phù hợp (ví dụ: Authentication, Route Management).
   * Các microservices xử lý nghiệp vụ và trả về kết quả qua API Gateway.
3. Theo dõi thời gian thực:
   * Thiết bị GPS trên xe gửi dữ liệu vị trí (latitude, longitude, timestamp) qua Ingest API.
   * Dữ liệu được lưu vào TimescaleDB để phân tích lịch sử và cache vào Redis để xử lý thời gian thực.
   * Dịch vụ Real-time Tracking phát dữ liệu vị trí qua WebSocket đến client (Dashboard/PWA) với độ trễ tối đa 3 giây.
4. Thông báo và cảnh báo:
   * Dịch vụ Notification & Alert gửi thông báo qua SMS (Twilio), Email (SendGrid), hoặc push notification (Firebase) dựa trên sự kiện (xe sắp đến, xe trễ, sự cố).
   * Tài xế có thể gửi cảnh báo khẩn cấp qua PWA, được chuyển tiếp đến quản trị viên/phụ huynh.
5. Báo cáo và phân tích:
   * Dịch vụ Reporting & Analytics truy vấn PostgreSQL và TimescaleDB để tạo báo cáo về số chuyến, trạng thái học sinh, hiệu suất tài xế/xe.

**3. Chi Tiết Chức Năng và Công Nghệ**

**3.1. Tầng Giao Diện**

* Công nghệ sử dụng:
  + React.js: Phát triển Web Dashboard cho quản trị viên, đảm bảo giao diện tiếng Việt (có thể mở rộng sang tiếng Anh).
  + React Native hoặc PWA: Ứng dụng cho tài xế và phụ huynh, hỗ trợ Android/iOS, sử dụng service workers để hoạt động offline.
  + Mapbox API: Hiển thị bản đồ và vị trí xe thời gian thực.
  + Socket.IO: Nhận dữ liệu vị trí real-time qua WebSocket.
  + Tailwind CSS: Thiết kế giao diện responsive, thân thiện với người dùng.
* Chức năng chi tiết:
  + Web Dashboard (Quản trị viên):
    - Hiển thị tổng quan: danh sách xe, học sinh, tài xế, tuyến đường, cảnh báo.
    - Tạo/cập nhật lịch trình xe (tuần/tháng) với giao diện calendar.
    - Phân công tài xế/xe, quản lý tuyến đường qua giao diện kéo-thả.
    - Theo dõi vị trí xe thời gian thực và xem lại hành trình.
  + PWA (Tài xế):
    - Xem lịch làm việc hàng ngày, danh sách học sinh và điểm đón.
    - Cập nhật trạng thái đón/trả học sinh (check-in/check-out).
    - Gửi cảnh báo sự cố (tai nạn, trễ giờ) qua nút bấm.
  + PWA (Phụ huynh):
    - Theo dõi vị trí xe con mình đang đi trên bản đồ.
    - Nhận thông báo khi xe đến gần điểm đón (dựa trên geo-fencing).
    - Xem lịch sử hành trình và nhận cảnh báo nếu xe trễ.

**3.2. Tầng Ứng Dụng**

* Công nghệ sử dụng:
  + Node.js (Express) hoặc Python (FastAPI): Xây dựng các microservices.
  + Nginx: Cổng API để phân tải và bảo mật.
  + Socket.IO: Hỗ trợ WebSocket cho dữ liệu thời gian thực.
  + RabbitMQ: Xử lý sự kiện bất đồng bộ (thông báo, cảnh báo).
* Chi tiết các dịch vụ:
  + Dịch vụ Xác thực (Authentication):
    - Sử dụng JWT để quản lý đăng nhập/đăng xuất.
    - Phân quyền: Quản trị viên (quyền cao nhất), Tài xế (quyền cập nhật trạng thái), Phụ huynh (quyền xem).
  + Dịch vụ Quản lý Người dùng & Thiết bị:
    - Quản lý tài khoản người dùng và thiết bị GPS (device\_key, trạng thái).
  + Dịch vụ Quản lý Học sinh & Phụ huynh:
    - Lưu thông tin học sinh, liên kết với phụ huynh.
    - Ghi nhận trạng thái đón/trả qua API từ tài xế.
  + Dịch vụ Quản lý Xe & Tài xế:
    - Quản lý thông tin xe (biển số, số ghế) và phân công tài xế.
  + Dịch vụ Quản lý Tuyến đường & Điểm dừng:
    - Tạo/sửa/xóa tuyến đường, danh sách điểm dừng (tọa độ địa lý).
  + Dịch vụ Lập lịch & Phân công:
    - Tạo lịch trình xe (ngày/tuần/tháng), gán xe/tài xế/học sinh.
  + Dịch vụ Theo dõi Thời gian thực:
    - Nhận dữ liệu GPS qua Ingest API, lưu vào TimescaleDB, cache vào Redis.
    - Phát dữ liệu qua WebSocket với độ trễ tối đa 3 giây.
  + Dịch vụ Thông báo & Cảnh báo:
    - Gửi thông báo tự động (xe sắp đến, xe trễ) qua SMS/Email/push.
    - Xử lý cảnh báo từ tài xế, chuyển tiếp đến quản trị viên/phụ huynh.
  + Dịch vụ Báo cáo & Phân tích:
    - Tạo báo cáo: số chuyến, trạng thái học sinh, hiệu suất xe/tài xế.

**3.3. Tầng Dữ Liệu**

* Công nghệ sử dụng:
  + PostgreSQL: Lưu trữ dữ liệu tĩnh (học sinh, xe, tuyến đường).
  + TimescaleDB: Lưu trữ dữ liệu GPS theo thời gian.
  + Redis: Cache vị trí xe và pub/sub cho WebSocket.
* Cấu trúc dữ liệu:
  + PostgreSQL:
    - users: id, role (admin/driver/parent), email, password\_hash, device\_key.
    - students: id, name, parent\_id, status (on\_bus/off\_bus).
    - buses: id, license\_plate, capacity, status.
    - routes: id, name, stops (array tọa độ).
    - schedules: id, route\_id, bus\_id, driver\_id, date.
  + TimescaleDB:
    - gps\_data: device\_id, timestamp, latitude, longitude.
  + Redis:
    - Key: bus:<id>:location (lưu vị trí hiện tại).
    - Channel: bus:<id>:updates (phát dữ liệu real-time).

**4. Tích Hợp và Yêu Cầu Kỹ Thuật**

* Tích hợp bên ngoài:
  + Thiết bị GPS: Gửi dữ liệu qua HTTP/MQTT đến Ingest API.
  + Dịch vụ thông báo: Twilio (SMS), SendGrid (Email), Firebase (Push).
  + Bản đồ: Mapbox hoặc Google Maps API.
* Yêu cầu kỹ thuật:
  + Hỗ trợ tiếng Việt cho giao diện và thông báo, với khả năng mở rộng sang tiếng Anh.
  + Đảm bảo xử lý 300 xe đồng thời với độ trễ dữ liệu thời gian thực tối đa 3 giây.
  + Hỗ trợ cả web dashboard và PWA (Android/iOS).

**5. Khả Năng Mở Rộng và Hiệu Suất**

* Mở rộng ngang: Microservices triển khai trên Docker/Kubernetes.
* Tối ưu hiệu suất: Redis cache giảm tải cho cơ sở dữ liệu; Nginx cân bằng tải.
* Độ tin cậy cao: Sử dụng replica cho PostgreSQL/TimescaleDB, Redis Cluster.

**6. Bảo Mật**

* Xác thực: JWT, OAuth2, kiểm tra device\_key.
* Mã hóa: HTTPS cho API, mã hóa dữ liệu GPS khi truyền.
* Bảo vệ API: Giới hạn tốc độ (rate limiting), danh sách IP cho phép (whitelisting).

**Phân rã Module (Module Breakdown)**

Dựa trên kiến trúc tổng thể và yêu cầu hệ thống BBS – Smart School Bus, hệ thống được chia thành các module chính như sau:

**1. Frontend Modules (Web Dashboard & PWA)**

Mục tiêu: Cung cấp giao diện tương tác cho các đối tượng người dùng: Quản trị viên, Tài xế và Phụ huynh.

* Dashboard Management:
  + Hiển thị tổng quan hệ thống (số xe hoạt động, số học sinh đã đón/trả, cảnh báo).
  + Quản lý danh sách xe, học sinh, tài xế, tuyến đường.
* Route & Schedule UI:
  + Giao diện tạo, chỉnh sửa tuyến đường và điểm dừng (kéo thả thứ tự điểm đón).
  + Giao diện tạo và theo dõi lịch trình (theo tuần/tháng).
* Real-time Tracking UI:
  + Hiển thị vị trí xe trên bản đồ theo thời gian thực.
  + Xem lại hành trình (route replay).
* Driver & Parent Apps (PWA):
  + Tài xế: Xem lịch làm việc, danh sách điểm đón, gửi cảnh báo, cập nhật trạng thái đón/trả.
  + Phụ huynh: Theo dõi xe, nhận thông báo xe sắp đến, xem lịch sử hành trình của con.

**2. Backend Modules (Application Layer)**

Mục tiêu: Cung cấp API và xử lý toàn bộ nghiệp vụ cốt lõi của hệ thống.

1. Authentication & Authorization Module
   * Quản lý đăng nhập, đăng xuất, cấp JWT token.
   * Phân quyền truy cập: Admin, Tài xế, Phụ huynh.
   * Bảo vệ API bằng xác thực thiết bị và token.
2. User & Device Management Module
   * Quản lý tài khoản người dùng (admin, tài xế, phụ huynh).
   * Quản lý thiết bị gửi GPS (device\_key, trạng thái kết nối).
3. Student & Parent Management Module
   * Quản lý thông tin học sinh, liên kết với phụ huynh.
   * Ghi nhận trạng thái đón/trả học sinh.
4. Bus & Driver Management Module
   * Quản lý thông tin xe buýt (biển số, số ghế, trạng thái).
   * Quản lý tài xế và phân công xe.
5. Route & Stop Management Module
   * Tạo, chỉnh sửa, xóa tuyến đường.
   * Quản lý danh sách điểm đón/trả.
6. Scheduling & Assignment Module
   * Tạo và cập nhật lịch trình theo ngày, tuần, tháng.
   * Phân công xe và tài xế cho từng chuyến.
   * Gán học sinh cho từng điểm đón.
7. Real-time Tracking Module
   * Tiếp nhận dữ liệu vị trí từ thiết bị GPS (qua Ingest API).
   * Cập nhật vị trí xe vào cache (Redis) và phát (broadcast) tới client qua WebSocket.
   * Ghi lại lịch sử hành trình xe.
8. Notification & Alert Module
   * Gửi thông báo thủ công từ quản trị đến tài xế hoặc phụ huynh.
   * Gửi thông báo tự động khi xe sắp đến điểm đón hoặc bị trễ.
   * Nhận cảnh báo từ tài xế và chuyển tiếp đến admin/phụ huynh.
9. Reporting & Analytics Module
   * Cung cấp báo cáo chi tiết: số chuyến, trạng thái học sinh, số lần trễ, sự cố.
   * Thống kê hiệu suất hoạt động của đội xe và tài xế.

**3. Data Layer (Database & Storage Modules)**

Mục tiêu: Quản lý, lưu trữ, và truy vấn dữ liệu hiệu quả, an toàn.

* Relational Database (PostgreSQL):
  + Lưu trữ dữ liệu có cấu trúc: học sinh, phụ huynh, xe, tuyến đường, lịch trình, thông báo.
* Time-series Database / Table (TimescaleDB):
  + Lưu dữ liệu vị trí GPS theo thời gian để phục vụ phân tích và truy xuất lịch sử hành trình.
* Cache & Realtime Layer (Redis):
  + Lưu vị trí xe hiện tại và hỗ trợ phát dữ liệu real-time.
  + Làm adapter pub/sub cho Socket.IO để mở rộng hệ thống.