

TravelCam - Product Requirement Document (PRD)

Developed by GEP Protech — Pioneers of the Next Tech Frontiers

Document Version: 1.0.0

Status: Approved Enterprise Architecture Specification

Date: August 26, 2026

Target Platform: Multi-Platform Enterprise Travel Solution (Laravel 13, Blade, React Native, React JS, MySQL/PSQL)

1. Executive Summary & Vision

TravelCam is an enterprise-grade, multi-tenant travel agency management and passenger ticketing platform engineered by GEP Protech. It is designed to streamline transport agency operations across multiple regions and branch terminals. The platform provides complete operational autonomy to travel agencies while offering passengers a seamless, real-time booking experience.

Core Objectives

- **1. Multi-Tenant Autonomy:** Isolated tenant data (agency_id), custom roles, fleet management, and localized branch pricing.
- **2. Dynamic Scope-Based Management:** Allow managers to oversee specific scopes (AGENCY, REGION, BRANCH) with granular action permissions without account duplication.
- **3. Branch-to-Branch Routing:** Support consolidated trip scheduling where a single bus picks up passengers across multiple intermediate branch stops with transparent departure timetables.

- **4. Station Physical Ticket Collection:** Enable passengers (both authenticated online users and walk-in counter guests) to search, select seats, book online, and receive digital booking reference codes. Passengers present their booking code or name at the agency terminal counter prior to departure to collect their official physical hard-copy paper ticket for bus boarding.
- **5. Polymorphic Infrastructure:** Utilize dynamic polymorphic engines for centralized notifications, enterprise activity audit logs, document attachments, financial ledgers, and service reviews.

2. Project Engineering Team & Developer Roles (GEP Protech)

The development, architecture, deployment, and quality assurance of the TravelCam platform are executed by the specialized engineering team at GEP Protech.

2.1 Key Project Personnel Roster

- Engr Ngulefac Terence
- Mr Elvis Ejah Malle
- Engr Maidugu Jean
- Engr Salemo Allen
- Engr Mac Junior
- Engr T Praise
- Engr Perly Piply
- Engr Ngali Mohamed Tasha
- Engr Dinga Favor
- Engr Jordan

2.2 Developer Roles & Assigned Personnel Matrix

Developer Role / Title	Assigned Personnel	Key Responsibilities & Module Focus	Primary Technology Stack
Lead Solution Architect / Project Manager	Engr Ngulefac Terence Mr Elvis Ejah Malle	System architecture design, database entity modeling, API security standards, multi-tenant scoping policies, milestone tracking, and stakeholder reporting.	Laravel Architecture, DB Modeling, System Design
Backend Engineers	Engr Maidugu Jean (Lead) Engr Salemo Allen	Core RESTful API development, Eloquent	Laravel (PHP 8.x), MySQL / PSQL, REST APIs,

	Engr Mac Junior Engr Ngulefac Terence Engr Jordan Engr T Praise Engr Nagali Mohamed	tenant scoping, dynamic scope-based RBAC engine, polymorphic service engines, and payment gateway integration.	Sanctum
Mobile App Developer (iOS & Android)	Engr Ngulefac Terence (Lead) Engr Mac Junior Engr Maidugu Jean	Cross-platform mobile app development, interactive bus seat matrix component, digital booking reference wallet, receipt generation, and push notifications.	React Native, Redux Toolkit / Context, Mobile SDKs
Frontend Web Developer	Engr Mac Junior (Lead) Engr Salemo Allen Engr Maidugu Jean Engr Perly Piply Engr Dinga Favor Engr T Praise	Admin panel interface implementation (Phase 1) and passenger single page application (Phase 4), station POS counter lookup, hard-copy ticket printing, and boarding manifests.	Laravel Blade, React JS, CSS3, JavaScript
Payment & Integrations Engineer	Engr Ngulefac Terence (Lead) Engr Maidugu Jean Engr Mac Junior	Direct integration of local mobile money APIs (CamPay & Fapshi), automated webhook callbacks, transaction ledger reconciliation, and payment security.	REST Webhooks, CamPay SDK, Fapshi API, TLS 1.3
QA & DevOps Engineer	Engr Ngulefac Terence (Lead) Engr Maidugu Jean	Cloud server provisioning on Hostinger, Nginx web server configuration, CI/CD pipeline automation, automated testing, app store deployment (Google Play & Apple App Store).	Hostinger VPS, Nginx, Redis, Git, App Store Connect

3. Stakeholders & Actor Profiles

Actor Role	Scope Level	Primary Responsibilities	Authentication Guard
System SuperAdmin	Platform	Global system overseer; approves new agency tenants, manages subscription tiers, audits platform health, and maintains global permission catalog.	auth:system_admin
Agency Manager	Agency	Agency owner/executive; manages agency structure (Regions & Branches), creates custom agency roles, registers fleet buses, and sets global agency policies.	auth:staff (AGENCY)
Regional Manager	Region	Oversees a cluster of branch terminals within a geographic region; monitors regional trip schedules and regional staff performance.	auth:staff (REGION)
Branch Manager	Branch	Manages local terminal operations, branch staff assignments, daily counter cash accounts, and local branch departure timetables.	auth:staff (BRANCH)
Branch Staff	Branch	Counter Cashiers selling walk-in tickets, verifying online passenger booking codes/names, and printing physical hard-copy tickets	auth:staff (BRANCH)

		for bus boarding.	
Registered Passenger	Customer	Authenticated customer with account history, saved profile, digital booking reference wallet, and service review capabilities.	auth:web / auth:api
Guest Passenger	Guest	Unauthenticated online or walk-in passenger booking tickets without creating an account (stored directly on Booking).	Guest (None)

4. Technical Architecture & Developer Tech Stack

This section outlines the technical stack, infrastructure, and architectural choices mandated by GEP Protech for the engineering team developing the TravelCam platform.

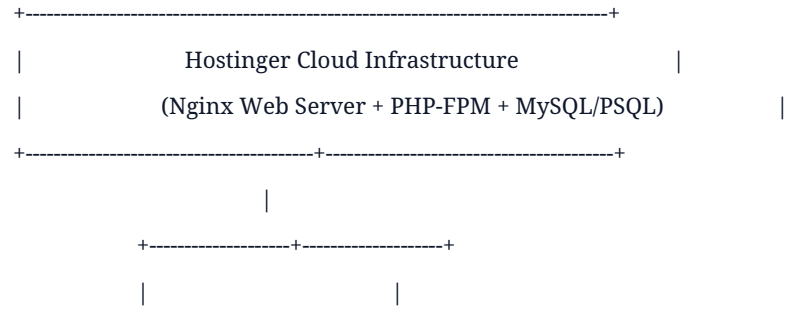
4.1 Technology Stack Matrix

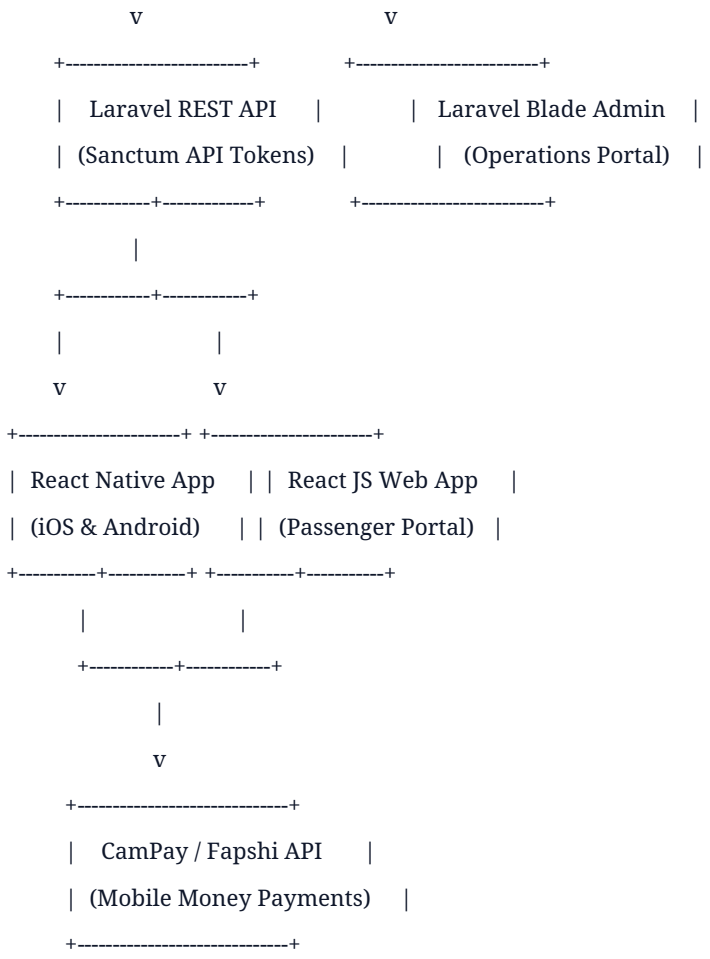
Component	Technology / Framework	Details & Justification
Mobile Application	React Native	Cross-platform native mobile application for Android and iOS. Delivers real-time trip booking, interactive seat selection, digital wallet, and booking reference delivery.
Admin Panel Frontend	Laravel 13 Blade	Server-side rendered administration portal integrated with Laravel backend. Used by SuperAdmins, Agency Managers, Branch Managers, and Counter Cashiers for booking lookup and ticket printing.
Full Web Application	React JS	Modern Single Page Application (SPA) for web passengers, delivering identical user experience and feature parity with the mobile app for browser

		booking.
Backend & REST API	Laravel 13 (PHP 8.4)	Modular RESTful API and core business logic engine, utilizing Laravel Sanctum for API token authentication and Eloquent ORM with global tenant scopes.
Database Engine	MySQL / PSQL	Relational database storage with ACID compliance, JSON column support, and spatial capabilities. Uses database indexes and FK constraints for tenant isolation.
Mobile Money & Payments	CamPay / Fapshi	Integrated local mobile money payment gateways (MTN Mobile Money, Orange Money) alongside card processing for automated payment verification.
Cloud Hosting	Hostinger	VPS / Dedicated Cloud Hosting environment running Nginx web server, PHP-FPM, Redis caching, and automated SSL certificate management.

4.2 Developer Integration Architecture

The system architecture follows a decoupled API-first design pattern deployed on Hostinger Cloud Infrastructure. Mobile clients (React Native) and Web Single Page Applications (React JS) communicate via secure REST APIs with Laravel Sanctum token authentication, while the internal operations admin portal renders directly through Laravel Blade.





5. Functional Requirements (FRs)

5.1 Module 1: Multi-Tenant Governance & Administration

- **FR-1.1 Agency Onboarding:** System SuperAdmin can review, approve, suspend, or reactivate agency tenant accounts.
- **FR-1.2 Tenant Data Isolation:** All tenant data (Branches, Routes, Staff, Fleet, Bookings) must be isolated per agency_id using global Eloquent database scopes.
- **FR-1.3 Subscription Management:** System Admin can assign subscription plans (e.g., maximum allowed branches, buses, or monthly booking volume).
- **FR-1.4 Global Feature Catalog:** Platform master permissions (Permission) are defined globally by SuperAdmins and mapped to backend policy gates.

5.2 Module 2: Unified Staff & Dynamic Scope-Based RBAC

- **FR-2.1 Unified Staff Table:** Managers and operational staff are consolidated into a single Staff table with employee_type (STAFF or MGR).
- **FR-2.2 Scope Assignments:** Staff members are dynamically assigned management scopes via StaffScopeAssignment referencing scopeable_type and scopeable_id.
- **FR-2.3 Tenant Role Definition:** Agencies can create custom job titles (Role) attached to their agency_id and map them to permissions via RolePermission.
- **FR-2.4 Multi-Role Support:** Operational staff can hold multiple active roles simultaneously (e.g. Cashier + Terminal Clerk) via StaffRole.

5.2.1 Non-Manager Operational Staff Management (Cashiers, Drivers, Dispatchers, Clerks)

Non-manager staff (such as Counter Cashiers, Bus Drivers, Conductors, Gate Dispatchers, and Terminal Clerks) operate within a streamlined, action-oriented workflow:

- **Home Terminal Binding:** Non-manager staff are directly assigned to their home terminal station via branch_id on the Staff table.
- **Operational Roles & Permissions:** Agencies define operational roles (e.g. 'Station Cashier', 'Bus Driver', 'Terminal Clerk') linked to global action permissions (booking.create, ticket.print). Staff members receive one or more roles via the StaffRole junction.
- **Trip Crew Assignments:** When scheduling a Trip, managers assign specific staff members to operational crew roles (driver_id -> Staff.id, conductor_id -> Staff.id).
- **Transaction Accountability:** Every walk-in ticket sale (Booking.staff_id) and counter payment (BranchCashAccount) records the exact staff member who processed the transaction.

5.3 Module 3: Fleet Management & Branch-to-Branch Routing

- **FR-3.1 Vehicle Registry:** Track bus registration numbers, total seat capacity, bus model, amenities (Wi-Fi, AC, VIP), and operational status.
- **FR-3.2 Bus Maintenance:** Log maintenance work orders, repair costs, servicing dates, and service status (BusMaintenance).
- **FR-3.3 Branch-to-Branch Routes:** Routes are explicitly defined between origin and destination branch terminals (Route & RouteBranchStop).
- **FR-3.4 Multi-Branch Pickup Scheduling:** Schedule trips (Trip & TripStop) with sequential pickup/dropoff branch stops, arrival times, and stop pricing.

5.4 Module 4: Booking & Station Ticket Issuance Engine

- **FR-4.1 Trip Search:** Passengers can search trips by selecting exact origin branch, destination branch, travel date, and time.
- **FR-4.2 Real-Time Seat Matrix:** Interactive bus seat layout display (Seat) allowing passengers or cashiers to pick specific available seats.
- **FR-4.3 Guest & Authenticated Booking:** Bookings support both registered users (user_id) and unauthenticated guest passengers (passenger_name, passenger_email, passenger_phone).
- **FR-4.4 Mobile Money & Counter Payments:** Process payments seamlessly via CamPay or Fapshi APIs (MTN Mobile Money, Orange Money) as well as counter cash payments.
- **FR-4.5 Digital Booking Reference:** Upon successful booking, the platform generates a unique Booking Reference Code (e.g. TC-892A4) stored in the passenger's account wallet and sent via SMS/Email.
- **FR-4.6 Onsite Counter Verification & Ticket Printing:** Passengers present their digital booking reference code or passenger name/phone number to the Counter Cashier at the agency terminal prior to departure. The cashier looks up the booking in the POS, verifies payment status, and prints the official physical hard-copy ticket.
- **FR-4.7 Walk-In Counter Sales:** Passengers visiting a physical terminal counter without an advance booking can buy tickets directly from an on-duty Counter Cashier, who collects payment and prints their physical hard-copy ticket.
- **FR-4.8 Cashier Shift Accounting:** Every counter transaction stores booking_channel, booked_by_staff_id, and pickup_branch_id. Cash collected is tied to the Cashier's daily shift account (BranchCashAccount) for shift reconciliation.

5.5 Module 5: Terminal Operations & Boarding Procedures

- **FR-5.1 Deferred Luggage Module:** Luggage weighing, fee calculations, and luggage tagging features are deferred for future releases. Passengers carry standard personal luggage directly onto the vehicle without terminal weighing.
- **FR-5.2 Hard-Copy Ticket Inspection:** Dispatchers and Bus Conductors inspect physical hard-copy tickets presented by passengers at the boarding gate/bus door to confirm seat assignment, departure terminal, and passenger name.
- **FR-5.3 Boarding Manifest:** Terminal staff update the real-time passenger boarding manifest as physical hard-copy tickets are verified, updating trip status (DEPARTED, IN_TRANSIT, COMPLETED).
- **FR-5.4 Daily Cash Reconciliation:** Branch Managers balance counter cash register collections with BranchCashAccount ledger entries at the end of each shift.

5.6 Module 6: Polymorphic Shared Engines

- **FR-6.1 Polymorphic Notifications:** Single Notification entity dispatching alerts to SystemAdmin, Staff, or User via notifiable_type + notifiable_id.
- **FR-6.2 Polymorphic Audit Log:** Log administrative and operational actions in ActivityLog linking causer (actor) and subject (entity modified).
- **FR-6.3 Polymorphic Attachments:** Store receipts, national ID copies, maintenance invoices in Attachment (attachable_type + attachable_id).
- **FR-6.4 Polymorphic Financial Ledger:** Record double-entry financial transactions in LedgerTransaction (transactionable_type + transactionable_id).
- **FR-6.5 Polymorphic Reviews:** Allow passengers to review trips, buses, or agency services via Review (reviewable_type + reviewable_id).

6. Non-Functional Requirements (NFRs)

6.1 Performance & Scalability

- **API Latency Target:** Trip search and availability requests respond in under 200ms across mobile and web clients.
- **Seat Lock Concurrency:** 5-minute atomic database locks during seat selection to prevent double-booking across concurrent cashier and passenger sessions.
- **Database Indexing:** Composite indexes on multi-tenant columns (agency_id, status), (trip_id, seat_number), and polymorphic columns (notifiable_type, notifiable_id).

6.2 Security & Data Isolation

- **Multi-Tenant Scoping:** Strict tenant isolation enforcing agency_id global scopes on all Eloquent models to prevent cross-tenant data leaks.
- **RBAC Policy Enforcement:** Every staff action verified against dynamic StaffScopeAssignment gates prior to execution.
- **Encryption & Communication:** Passwords hashed with bcrypt; API tokens authenticated via Laravel Sanctum; TLS 1.3 enforced across all web/mobile endpoints.

6.3 Availability & Reliability

- **Uptime Target:** 99.9% application uptime hosted on Hostinger Cloud Infrastructure with Nginx and PHP-FPM process management.
- **Automated Backups:** Daily automated database snapshots with point-in-time recovery up to 30 days.

7. Phased Deliverables & Implementation Roadmap

The development and deployment of TravelCam by GEP Protech will follow a strict 4-phase sequential delivery roadmap:

Phase 1: Admin Panel Web App (Laravel & Blade) — Primary Priority

The primary objective of Phase 1 is to establish the database foundation and operational administration portal:

- **Core Database Foundation:** Database schema migrations (MySQL/PSQL) with multi-tenant agency_id scoping.
- **Management Portals:** System SuperAdmin, Agency Manager, Regional Manager, and Branch Manager administrative dashboards.
- **Station Counter POS Interface:** Station Counter Cashier POS interface for booking lookup by reference code or passenger name, cash/mobile money collections, and hard-copy paper ticket printing.
- **Boarding & Crew Tools:** Dispatcher boarding manifest tool for passenger check-in and crew assignment.
- **Target Output:** Fully functional backend REST API and operational Admin Panel ready for agency deployment.

Phase 2: Mobile Application (React Native for Android & iOS) — Secondary Priority

Phase 2 focuses on delivering the native passenger mobile application for iOS and Android devices:

- **Cross-Platform Application:** Cross-platform mobile app built with React Native for Android and iOS.
- **Passenger Features:** Real-time trip search, interactive bus seat matrix selection, guest and user account management, and digital booking reference wallet.
- **Mobile Money Integration:** Direct integration with CamPay and Fapshi payment APIs for instant mobile money payment processing.
- **Push Notifications:** Instant trip departure alerts, booking confirmations, and schedule updates.

Phase 3: Landing Page & App Redirect Website — Tertiary Priority

Phase 3 establishes the public web presence and app store redirection website:

- **Public Web Presence:** Modern marketing website presenting TravelCam features, partner travel agencies, and route networks.

- **App Store Redirects:** Smart download badges and QR links directing visitors to download the official mobile apps on the Google Play Store and Apple App Store.

Phase 4: Full Passenger Web Application (React JS) — Final Phase

Phase 4 extends online passenger booking capabilities to web browsers with a full SPA frontend:

- **Web App SPA:** Built with React JS SPA, bringing complete feature parity with the React Native mobile app to desktop and mobile web browsers.
- **Web Passenger Portal:** Allows desktop users to search trips, select seats, complete CamPay/Fapshi mobile money payments, and view digital booking references directly on the web.

8. Master 27-Entity Domain Model Summary

- 1. SystemAdmin (Platform Governance)
- 2. Agency (Tenant Root)
- 3. Region (Organizational Hierarchy)
- 4. Branch (Station Terminal Base)
- 5. Staff (Unified Employee Model)
- 6. StaffScopeAssignment (Dynamic Scope Scoping)
- 7. Role (Tenant Custom Role)
- 8. Permission (Global Feature Capability)
- 9. RolePermission (Junction Table)
- 10. StaffRole (Multi-Role Junction)
- 11. Bus (Fleet Vehicle)
- 12. BusMaintenance (Work Orders & Servicing)
- 13. Seat (Bus Layout Matrix)
- 14. Route (Branch-to-Branch Route)
- 15. RouteBranchStop (Intermediate Branch Order)
- 16. Trip (Scheduled Departure)
- 17. TripStop (Timetable Pickup/Dropoff)
- 18. User (Registered Passenger)
- 19. Booking (Guest/Auth Reservation & Digital Code)
- 20. Ticket (Physical Hard-Copy Issued Ticket)
- 21. Luggage (Deferred / Simplified baggage entry)
- 22. BranchCashAccount (Counter Cash Shift)
- 23. Notification (Polymorphic Alert Engine)
- 24. ActivityLog (Polymorphic Audit Trail)

- 25. Attachment (Polymorphic Documents)
- 26. LedgerTransaction (Double-Entry Accounting)
- 27. Review (Polymorphic Passenger Rating)