# MULTI-TENANT SAAS UPGRADE – FEATURE & REQUIREMENT SPECIFICATION

## 1. PURPOSE & SCOPE

* Convert the existing single-tenant service platform into a shared, multi-tenant SaaS that lets independent service-provider businesses (“tenants”) run booking, workforce, and customer-engagement operations on one infrastructure while maintaining data isolation and custom branding.

## 2. TENANT MODEL & USER ROLES

| Role | Scope | Responsibilities |
| --- | --- | --- |
| Platform Super-Admin | Global | Manage tenants, subscription plans, billing, global configuration, system health, security, and compliance. |
| Service Provider Admin | Tenant only | Configure services, pricing, availability, branding, staff, and customer settings; view tenant analytics; handle tenant-level billing and payouts. |
| Provider Staff (Team Member) | Tenant only | Accept/perform jobs, manage schedule and profile, in-app communication, training, rewards. |
| End Customer | Tenant only | Discover services, book and manage appointments, communicate with staff, pay and tip. |

## 3. CORE FEATURE MODULES (Tenant-Scoped)

### 3.1 Booking Portal

* Custom domain and whitelabel branding per tenant
* Service catalogue per market or city
* Zip-code lookup, quote engine, real-time availability
* Discount codes, prepaid vouchers, dynamic pricing
* Appointment scheduler with deposit / checkout
* Cross-channel availability sync and blocked-dates view
* Analytics tags for marketing attribution

### 3.2 Provider Staff App

* Account creation with SSO (tenant or social)
* Full job lifecycle (accept → perform → close) with geo clock-in
* In-shift chat (staff ↔ customer, staff ↔ staff)
* Points, badges, earnings goals, milestones, in-app rewards
* Training modules, surveys, help chat
* Ride-sharing and “Go Live” on-demand mode
* Profile, payout management, availability, auto-scheduling

### 3.3 Customer App

* Account sign-in, referral codes, loyalty points
* Book and manage appointments; on-demand help and to-do lists
* Team verification, tipping, reviews, favorites
* Chat, dynamic FAQs, local resources, video submissions, surveys

### 3.4 Tenant Admin Dashboard

* Live appointment feed with color-coded pipeline and task queue
* Create or update bookings; blocked-time management
* Staff roster with activity logs and payout controls
* Custom reports; pricing, services, discounts per market
* Notification center and help-ticket routing

### 3.5 APIs & Webhooks

* Tenant-scoped REST / GraphQL endpoints
* Automatic billing and payroll
* Email, SMS, and push notification services
* Event webhooks for bookings, payments, staff status

## 4. SAAS PLATFORM LAYER (Cross-Tenant Services)

* Tenant onboarding and provisioning – self-serve flow spins up isolated schemas and storage with default configurations.
* Subscription and usage billing – per-tenant plans, metered events (bookings, staff seats, SMS) through Stripe.
* Whitelabel and theming – logo, colors, domain, app-store listing variants.
* Identity and access management – unified authentication (OAuth/SAML), RBAC, SCIM sync.
* Data isolation and security – row-level tenancy filters, encryption at rest and in transit, GDPR / CCPA tooling.
* Analytics and reporting – tenant dashboards plus platform-wide BI; export to CSV or JSON.
* Integration hub – Zapier connectors, webhooks, marketplace for add-ons (accounting, CRM, etc.).
* Support center – in-app ticketing routed to tenant admins and/or platform support.

## 5. NON-FUNCTIONAL REQUIREMENTS

* Scalability – horizontal auto-scaling (Kubernetes); multi-region readiness; CDN for assets.
* Performance – 95th-percentile response under 300 ms for core API calls; real-time WebSocket for chat and notifications.
* Reliability – 99.9 % uptime SLA; automated backups; blue/green deployments; feature flags.
* Security – SOC 2 Type II readiness; OWASP coverage; audit logs; MFA enforcement.
* Compliance – PCI-DSS for payments; HIPAA-ready data design toggle; GDPR data-subject-request APIs.

## 6. TECHNICAL ARCHITECTURE (High-Level)

* Micro-services with event bus (Kafka).
* PostgreSQL with row-level security per tenant; Redis cache.
* Object storage (S3-compatible) with isolated buckets.
* Authentication – AWS Cognito or Auth0 with custom tenant claims.
* CI/CD – GitHub Actions → Argo CD → Kubernetes.
* Observability – OpenTelemetry, Prometheus, Grafana.

## 7. MVP ROADMAP (Weeks)

* Tenant Provisioning & Auth — Weeks 1-3
* Booking Portal (Web) & Core APIs — Weeks 2-6
* Staff & Customer Apps (Beta) — Weeks 5-9
* Tenant Admin Dashboard & Billing — Weeks 7-11
* Platform Super-Admin Console & Analytics — Weeks 10-12

## 8. FUTURE ENHANCEMENTS

* AI-driven scheduling and pricing recommendations
* Marketplace for third-party service modules (cleaning, lawn-care, handyman)
* Franchise mode: parent tenant with sub-tenants
* Advanced workforce optimization (route planning, load balancing)
* IoT integrations (smart locks, vehicle telematics)
* Community features: volunteer exchange and loyalty tiers

UI Design Inspo - <https://www.figma.com/design/BFGLbUSyJK54DFPDX8d2h1/Evez2.0-Light-Theme--copy?m=auto&t=rra0EnJCn3T3yJTa-1>

— End of Specification —