Enterprise Wellness SaaS Platform

Product Requirements Document (PRD) & System Design Document

Version: 1.0
Prepared For: Enterprise Product, Engineering, Security & Compliance Teams
Document Type: Enterprise SaaS Architecture & Product Specification
Deployment Model: Multi-Tenant SaaS

---

1. PLATFORM OVERVIEW

1.1 Executive Summary

The Wellness SaaS Platform is a multi-tenant enterprise wellness ecosystem designed for organizations to provide holistic wellness services to employees, members, or customers through verified experts including therapists, coaches, nutritionists, fitness trainers, and wellness consultants.

The platform supports:

- Enterprise onboarding
- Multi-organization tenancy
- Secure user management
- Expert discovery and assignment
- Wellness program orchestration
- Session booking and engagement
- Compliance-ready data governance
- Enterprise-grade security and auditability

The platform is intended to operate as a scalable Software-as-a-Service (SaaS) solution with strict tenant isolation, enterprise identity integrations, configurable role-based permissions, and compliance-oriented lifecycle management.

1.2 Target Users

Primary Users

Platform Admins

SaaS owners and operational teams managing the global platform.

Organization Admins

HR teams, wellness managers, and enterprise administrators managing employee wellness programs.

Experts

Certified professionals providing wellness services.

End Users

Employees or members consuming wellness services.

1.3 Multi-Tenant SaaS Overview

The platform is architected as a secure multi-tenant SaaS application where:

- Multiple organizations share the same application infrastructure.
- Each organization has logically isolated data.
- Access is restricted using tenant-aware authorization.
- All business entities are scoped by organization_id.
- Cross-tenant access is prevented through backend enforcement.

The architecture supports:

- Shared infrastructure
- Horizontal scaling
- Tenant-level customization
- Enterprise onboarding
- Future white-label support

1.4 Key Business Goals

Business Objectives

- Enable enterprises to launch wellness programs rapidly.
- Centralize employee wellness operations.
- Provide secure expert-user interactions.
- Support enterprise-grade compliance.
- Enable scalable SaaS monetization.
- Reduce operational overhead through automation.

Technical Objectives

- High availability architecture
- Secure multi-tenancy

- Extensible RBAC framework
- Scalable onboarding system
- Immutable auditability
- Compliance-ready data governance

1.5 Core Modules

Administrative Modules

- Platform Administration
- Organization Management
- Tenant Configuration
- Billing & Subscription
- Audit & Compliance Center

User Management Modules

- User Directory
- Invitation Management
- Access Control
- Expert Verification
- Role Management

Wellness Modules

- Expert Discovery
- Appointment Scheduling
- Wellness Sessions
- Progress Tracking
- Messaging & Communication

Compliance Modules

- Consent Management
- Privacy Center
- Data Export
- Deletion Management
- Audit Logging

---

2. USER ROLES & PERMISSIONS

2.1 Role Hierarchy

System hierarchy:

1. Platform Admin
2. Organization Admin
3. Expert
4. End User

Hierarchy Principles

- Higher roles inherit visibility, not unrestricted access.
- Tenant boundaries override hierarchy.
- Platform Admins operate across tenants.
- Organization Admins operate only within assigned organizations.
- Experts access only assigned or authorized records.
- End Users access only their own data.

2.2 Platform Admin

Permissions

- Create organizations
- Suspend organizations
- View global analytics
- Manage subscriptions
- Configure platform settings
- Review audit logs
- Force revoke sessions
- Approve enterprise onboarding
- Manage compliance settings
- Manage feature flags

Restrictions

- Cannot impersonate users without explicit audit trail.
- Cannot access confidential session content without privileged workflow.
- Restricted from modifying immutable audit logs.

Access Boundaries

- Global access across all tenants.
- Access controlled through privileged administrative policies.

Example Use Cases

- Suspend malicious tenant
- Investigate security incident
- Enable enterprise feature
- Review compliance export request

2.3 Organization Admin

Permissions

- Invite users
- Manage organization users
- Assign experts
- Configure organization wellness programs
- View organizational analytics
- Manage organization policies
- Initiate organization deletion request
- Export organization reports

Restrictions

- Cannot access other organizations.
- Cannot elevate self to Platform Admin.
- Cannot bypass compliance retention.

Access Boundaries

- Restricted to assigned organization_id.

Example Use Cases

- Bulk onboard employees
- Assign therapists to teams
- Suspend employee access
- Export wellness participation report

2.4 Experts

Permissions

- Manage availability
- Access assigned sessions
- View assigned users
- Submit wellness notes
- Conduct sessions
- Send approved communications

Restrictions

- Cannot access organization analytics.
- Cannot access users outside assignments.

- Cannot export bulk data.

Access Boundaries

- Limited to assigned organizations and sessions.

Example Use Cases

- Conduct therapy session
- Submit wellness recommendations
- Manage appointments

2.5 End Users

Permissions

- Book sessions
- View personal wellness records
- Manage profile
- Request data export
- Submit deletion request
- Communicate with assigned experts

Restrictions

- Cannot access other users.
- Cannot view expert internal notes.
- Cannot access organization administration.

Access Boundaries

- Self-only access.

Example Use Cases

- Schedule coaching session
- Track wellness goals
- Download personal data archive

2.6 Permission Matrix

Capability	Platform Admin	Organization Admin	Expert	End User
Create Organization	Yes	No	No	No
Manage Tenant Settings	Yes	Limited	No	No

Capability	Platform Admin	Organization Admin	Expert	End User
Invite Users	Yes	Yes	No	No
Assign Experts	Yes	Yes	No	No
Access Wellness Records	Limited	Limited	Assigned Only	Self Only
Export Data	Yes	Scoped	No	Self Only
View Audit Logs	Yes	Scoped	No	No
Delete Accounts	Yes	Scoped	Self	Self
Manage Roles	Yes	Scoped	No	No
Configure SSO	Yes	Yes	No	No

---

3. MULTI-TENANT ARCHITECTURE

3.1 Architecture Overview

The platform uses a logical multi-tenant architecture with tenant-aware authorization.

Each organization represents a tenant.

Core isolation mechanisms:

- organization_id scoping
- Row-level authorization
- Tenant-aware APIs
- Tenant-scoped caching
- Tenant-scoped background jobs

3.2 Tenant Isolation Model

Shared Database with Logical Isolation

Recommended model:

- Shared PostgreSQL cluster
- Shared schema
- Tenant-scoped rows
- organization_id attached to tenant-owned entities

Benefits:

- Lower operational cost
- Easier scaling
- Simplified migrations
- Centralized analytics

Risks mitigated by:

- Strict backend authorization
- Row-level security
- Tenant validation middleware

3.3 Alternative Isolation Strategies

Option A: Shared DB / Shared Schema

Pros:

- Cost efficient
- Easy maintenance

Cons:

- Requires strong authorization discipline

Option B: Shared DB / Separate Schema

Pros:

- Stronger isolation

Cons:

- Operational complexity

Option C: Dedicated Database per Tenant

Pros:

- Maximum isolation

Cons:

- High infrastructure overhead

3.4 Recommended Enterprise Strategy

Recommended Hybrid Model:

- Shared DB for SMB and mid-market tenants
- Dedicated database option for enterprise customers

3.5 Row-Level Security (RLS)

PostgreSQL Row-Level Security policies enforce:

- organization_id match
- ownership validation
- scoped access rules

Example Policy Logic:

- Users can only access rows where organization_id matches active tenant context.
- Experts can only access assigned records.
- Organization admins can access organization-owned records.

3.6 Data Ownership Model

Entity	Owner
Organization	Platform
Users	Organization
Expert Profiles	Platform + Expert
Wellness Records	Organization + User
Audit Logs	Platform
Messages	Shared contextual ownership

3.7 Cross-Tenant Prevention Mechanisms

Required Controls

- Tenant-aware ORM filters
- organization_id validation middleware
- Signed tenant context tokens
- Separate Redis namespaces
- Tenant-scoped queues
- Audit monitoring for anomalous access

Security Safeguards

- Prevent direct object reference attacks
- Validate ownership before query execution

- Enforce tenant context at service layer
- Block cross-tenant joins without authorization

---

4. AUTHENTICATION & AUTHORIZATION

4.1 Authentication Methods

Supported authentication methods:

- Email/password
- Google OAuth SSO
- Microsoft OAuth SSO
- Magic link login
- Enterprise SAML SSO (future-ready)

4.2 Email/Password Authentication

Requirements

- BCrypt/Argon2 password hashing
- Password complexity validation
- Breached password detection
- Password rotation support

Security Controls

- Login throttling
- CAPTCHA on abuse detection
- IP anomaly detection
- Device fingerprinting

4.3 OAuth SSO

Google SSO

Supported for:

- End users
- Organization admins
- Experts

Microsoft SSO

Supported for enterprise organizations.

Features:

- Azure AD integration
- Domain-based auto provisioning
- Enterprise identity federation

4.4 Magic Link Authentication

Flow

1. User enters email.
2. Signed token generated.
3. Email link delivered.
4. Token validated.
5. Session established.

Security

- Short expiration window
- Single-use tokens
- Device/IP validation

4.5 SAML Support

Future enterprise support includes:

- Okta
- Azure AD
- OneLogin
- Ping Identity

Capabilities:

- Just-in-time provisioning
- SCIM synchronization
- Enterprise identity federation

4.6 JWT & Session Management

Token Strategy

Access Token:

- Short-lived
- JWT-based
- Signed with rotating keys

Refresh Token:

- Stored securely
- Revocable
- Device-bound

4.7 MFA Support

Supported MFA methods:

- TOTP authenticator apps
- Email OTP
- SMS OTP (optional)

Required for:

- Platform admins
- Enterprise admins
- Sensitive actions

4.8 Session Revocation

Capabilities

- Revoke all sessions
- Device-level revocation
- Forced logout
- Suspicious login invalidation

4.9 Invite Token Architecture

Invite tokens:

- Cryptographically signed
- Single-use

- Expiring
- Tenant-bound
- Role-bound

Invite metadata:

- organization_id
- invited_role
- expiration_timestamp
- invited_email

4.10 Authorization Model

RBAC

Role-based permissions define high-level access.

ABAC

Attribute-based access validates:

- organization ownership
- resource ownership
- assignment relationships
- verification status

Ownership Validation

All requests validate:

- tenant context
- actor permissions
- resource ownership
- lifecycle status

---

5. COMPLETE USER ONBOARDING FLOWS

5A. Organization Creation Flow

Workflow

1. Platform Admin creates organization.
2. Organization record enters PENDING_SETUP.
3. Primary admin invitation generated.

4. Domain configuration initiated.
5. Subscription configured.
6. Organization activated.

Validation Checks

- Domain uniqueness
- Subscription eligibility
- Email validation
- Tenant slug uniqueness

Notifications

- Welcome email
- Setup instructions
- Billing activation notification

Failure Handling

- Retry provisioning
- Suspend incomplete onboarding
- Expire inactive setup requests

5B. Organization Admin Invitation Flow

Workflow

1. Platform Admin creates invitation.
2. Secure token generated.
3. Email delivered.
4. Admin accepts invitation.
5. MFA setup required.
6. Account activated.

Status Transitions

PENDING → ACCEPTED → ACTIVE

Expired Invitations

- Auto-expire after configurable duration.
- Allow resend.
- Audit all reissues.

5C. User Invitation Flow

Workflow

1. Organization Admin invites user.
2. Invitation stored.
3. Email notification delivered.
4. User accepts.
5. Account created.
6. Organization membership linked.

Security Checks

- Email domain validation
- Duplicate membership prevention
- Invite throttling

5D. Bulk User Import (CSV)

Flow

1. Admin uploads CSV.
2. File scanned.
3. Schema validated.
4. Duplicate detection executed.
5. Preview generated.
6. Import job queued.
7. Invitations dispatched.

Failure Handling

- Partial rollback support
- Row-level error reporting
- Retry support

5E. Direct User Creation

Used for:

- HR-managed onboarding
- Enterprise migration
- Manual provisioning

Security Controls

- Mandatory temporary password rotation
- Email verification
- Admin attribution logging

5F. Domain Auto-Join

Flow

1. Organization configures verified domain.
2. User signs up with matching email.
3. Auto-association rule triggered.
4. Organization membership assigned.

Security

- Domain verification required
- DNS validation
- Approval mode optional

5G. Expert Onboarding

Flow

1. Expert registers.
2. Profile created.
3. Credentials uploaded.
4. Verification initiated.
5. Compliance review conducted.
6. Expert activated.

Required Documents

- Government ID
- Certifications
- Licenses
- Tax information

5H. Expert Verification Flow

Verification Stages

- Pending Review
- Under Verification
- Approved
- Rejected
- Suspended

Validation

- Credential authenticity
- Background review
- Duplicate profile detection
- Compliance eligibility

---

6. USER LIFECYCLE STATE MACHINE

6.1 User States

States

- INVITED
- PENDING_ACTIVATION
- ACTIVE
- SUSPENDED
- DEACTIVATED
- DELETION_REQUESTED
- SOFT_DELETED
- PURGED

6.2 Allowed User Transitions

Current State	Allowed Transition
INVITED	PENDING_ACTIVATION
PENDING_ACTIVATION	ACTIVE
ACTIVE	SUSPENDED
ACTIVE	DELETION_REQUESTED
SUSPENDED	ACTIVE
DELETION_REQUESTED	SOFT_DELETED

Current State	Allowed Transition
SOFT_DELETED	PURGED

6.3 Invalid User Transitions

Examples:

- PURGED → ACTIVE
- INVITED → PURGED
- DEACTIVATED → ACTIVE without approval

6.4 Recovery States

Supported Recovery

- Restore suspended accounts
- Restore soft-deleted users within retention window
- Reinstate deactivated experts after review

6.5 Expert Lifecycle States

- PENDING_VERIFICATION
- VERIFIED
- SUSPENDED
- DEACTIVATED
- ANONYMIZED

6.6 Organization Lifecycle States

- ACTIVE
- PAYMENT_DUE
- SUSPENDED
- ORG_PENDING_DELETION
- PURGED

---

7. ACCOUNT DELETION & DATA RETENTION SYSTEM

7.1 Deletion Architecture Principles

Deletion architecture must:

- Preserve audit integrity
- Prevent accidental destruction
- Support regulatory compliance
- Enable recovery windows
- Protect analytical continuity

7.2 Soft Deletion

Soft deletion flags records as inactive.

Fields:

- deleted_at
- deleted_by
- deletion_reason
- retention_expiry

Benefits:

- Recovery support
- Audit preservation
- Legal hold support

7.3 Hard Deletion

Hard deletion occurs only after:

- Retention expiration
- Compliance review
- Dependency cleanup
- Backup retention validation

7.4 Data Anonymization

Anonymization replaces identifying data while preserving analytics.

Example:

- Replace names with pseudonymous IDs
- Remove email addresses
- Retain aggregated wellness metrics

7.5 Why Immediate Hard Deletion Is Dangerous

Immediate deletion risks:

- Regulatory violations
- Audit destruction
- Fraud investigation obstruction
- Accidental loss
- Broken relational integrity

Recommended:

- Grace period
- Reversible deletion window
- Staged purge pipeline

7A. End User Deletion Request

Flow

1. User submits deletion request.
2. MFA verification required.
3. Request enters review queue.
4. Grace period initiated.
5. Data export offered.
6. Soft deletion executed.
7. Purge scheduled.

Recovery Window

30–90 days configurable.

7B. Organization Admin Deleting User

Flow

1. Admin initiates deletion.
2. Ownership validated.
3. User notified.
4. Retention policy applied.
5. Access revoked.
6. Records anonymized.

7C. Expert Deletion Request

Additional Requirements

- Session dependency validation
- Billing settlement validation
- Regulatory retention review

7D. Organization Deletion Request

Flow

1. Organization admin requests deletion.
2. Platform review initiated.
3. Billing closure validated.
4. Legal retention review conducted.
5. Data export package generated.
6. Organization suspended.
7. Purge scheduled.

7E. Platform Admin Forced Deletion

Used for:

- Fraud
- Abuse
- Legal requirements
- Security incidents

Requirements

- Immutable audit trail
- Multi-admin approval
- Incident reference ID

7.6 Analytics Preservation Strategy

Analytics retained using:

- Aggregated metrics
- Anonymized identifiers
- Non-identifiable event summaries

PII must be removed before retention.

---

8. PRIVACY, COMPLIANCE & CONSENT

8.1 Compliance Targets

The platform should support:

- GDPR
- India DPDP Act
- HIPAA-like wellness privacy controls

8.2 Consent Tracking

Consent records include:

- consent_type
- granted_at
- revoked_at
- policy_version
- user_agent
- IP address

8.3 Privacy Center

Features:

- Download my data

- Manage consents
- Request deletion
- View active sessions
- Revoke device access

8.4 Data Export

Export formats:

- JSON
- CSV
- PDF summary

Exports must:

- Be encrypted
- Expire automatically
- Require re-authentication

8.5 Legal Retention

Examples:

- Audit logs retained 7 years
- Billing records retained per tax law
- Wellness notes retained per contractual obligations

8.6 Data Minimization

Principles:

- Collect only necessary data
- Avoid excessive health information
- Limit sensitive data exposure

---

9. AUDIT LOGGING SYSTEM

9.1 Audit Logging Principles

Audit logs must be:

- Immutable

- Tamper-evident
- Timestamped
- Actor-attributed
- Tenant-aware

9.2 Security-Sensitive Events

Examples:

- USER_DELETED
- ROLE_CHANGED
- LOGIN_FAILED
- MFA_DISABLED
- DATA_EXPORTED
- SESSION_VIEWED
- PERMISSION_ESCALATION_ATTEMPT
- ORG_SUSPENDED

9.3 Audit Event Structure

Fields:

- event_id
- actor_id
- actor_role
- organization_id
- event_type
- resource_type
- resource_id
- timestamp
- IP_address
- device_metadata
- before_state
- after_state

9.4 Why Audit Logs Matter

Critical for:

- Compliance
- Forensics

- Security investigations
- Insider threat monitoring
- Enterprise trust

---

10. DATABASE DESIGN

10.1 Database Overview

Recommended Database:

- PostgreSQL

Supporting Services:

- Redis
- Object storage
- Search index (optional)

10.2 Core Tables

organizations

Purpose:

- Tenant management

Important Fields:

- id
- name
- slug
- status
- subscription_plan
- settings_json
- created_at

Relationships:

- One-to-many users
- One-to-many experts

Security Concerns:

- Tenant isolation

- Billing metadata protection

---

users

Purpose:

- User identity management

Important Fields:

- id
- organization_id
- email
- password_hash
- status
- MFA_enabled
- deleted_at

Relationships:

- Many-to-many roles
- One-to-many appointments

Security Concerns:

- PII protection
- Authentication security

---

roles

Purpose:

- RBAC role definitions

---

permissions

Purpose:

- Granular permission mapping

---

user_organizations

Purpose:

- Multi-org memberships

---

experts

Purpose:

- Expert profiles and verification

Important Fields:

- verification_status
- license_metadata
- specialization

---

expert_assignments

Purpose:

- Expert-to-user relationships

---

appointments

Purpose:

- Scheduling and session lifecycle

---

sessions

Purpose:

- Wellness interactions

Security Concerns:

- Sensitive notes
- Access restrictions

---

wellness_records

Purpose:

- Wellness tracking and assessments

---

messages

Purpose:

- Secure communication

Security Concerns:

- Encryption
- Retention policies

---

audit_logs

Purpose:

- Immutable event tracking

---

deletion_requests

Purpose:

- Lifecycle governance

---

consents

Purpose:

- Legal consent tracking

10.3 Soft Delete Implementation

Pattern:

- deleted_at timestamp

- filtered queries
- restoration support

10.4 Indexing Strategy

Recommended Indexes:

- organization_id
- email
- status
- created_at
- composite tenant indexes

Example:

(organization_id, status)

10.5 Multi-Tenant Querying Strategy

All tenant-owned queries must:

- Include organization_id filter
- Validate access before execution
- Prevent unrestricted scans

---

11. API DESIGN

11.1 REST API Standards

Principles:

- RESTful conventions
- Versioned APIs
- JSON responses
- Consistent error structures

Base Path:

/api/v1

11.2 Authentication APIs

POST /auth/login
POST /auth/logout
POST /auth/refresh
POST /auth/magic-link
POST /auth/password-reset
POST /auth/mfa/verify

11.3 Invitation APIs

POST /organizations/{id}/invite
GET /invitations/{token}
POST /invitations/accept

11.4 Organization APIs

POST /organizations
GET /organizations/{id}
PATCH /organizations/{id}
DELETE /organizations/{id}

11.5 Expert APIs

POST /experts
GET /experts/{id}
POST /experts/{id}/verify
POST /expert-assignments

11.6 Deletion APIs

POST /deletion-requests
GET /deletion-requests/{id}
POST /deletion-requests/{id}/approve

11.7 Audit APIs

GET /audit-logs
GET /audit-logs/export

11.8 Consent APIs

POST /consents
PATCH /consents/{id}/revoke

11.9 Authorization Middleware

Every request validates:

- Authentication
- Tenant scope
- RBAC permissions
- Resource ownership

11.10 Error Handling

Standardized errors:

- AUTHENTICATION_FAILED
- ACCESS_DENIED
- TENANT_MISMATCH
- RESOURCE_NOT_FOUND
- VALIDATION_ERROR

11.11 Rate Limiting

Rate limits by:

- IP address
- User ID
- Organization
- Endpoint category

---

12. SECURITY ARCHITECTURE

12.1 Encryption

At Rest

- AES-256 encryption

- Encrypted database volumes
- Encrypted object storage

In Transit

- TLS 1.2+
- HSTS enforcement
- Secure cookie configuration

12.2 Secret Management

Recommended:

- AWS Secrets Manager
- HashiCorp Vault
- Kubernetes secrets

Never store:

- Secrets in code
- Plaintext API keys

12.3 Role Escalation Prevention

Controls:

- Explicit permission validation
- Immutable admin audit logs
- Approval workflows

12.4 Secure File Uploads

Requirements:

- Malware scanning
- File type validation
- Signed upload URLs
- Storage isolation

12.5 Session Security

Features:

- Device management

- Session revocation
- Suspicious login detection
- IP intelligence

12.6 Application Security

Protections:

- CSRF protection
- XSS sanitization
- SQL injection prevention
- CSP headers
- Secure cookies

12.7 Zero Trust Principles

- Never trust internal traffic
- Validate every request
- Enforce identity everywhere
- Continuous authorization

12.8 Least Privilege Principle

Users receive:

- Minimal required access
- Time-limited elevation
- Scoped permissions

---

13. NOTIFICATION SYSTEM

13.1 Notification Channels

Supported channels:

- Email
- Push notifications
- In-app notifications
- SMS (optional)

13.2 Notification Events

Examples:

- Invitation received
- Account activated
- Session reminder
- Expert assigned
- Organization suspended
- Deletion request initiated

13.3 Notification Architecture

Recommended:

- Queue-driven delivery
- Retry mechanisms
- Delivery tracking
- Provider abstraction layer

---

14. SCALABILITY & INFRASTRUCTURE

14.1 Recommended Stack

Backend

Recommended:

- Node.js + NestJS OR
- Python + Django

Database

- PostgreSQL

Cache

- Redis

Queue System

- BullMQ OR
- RabbitMQ

Storage

- AWS S3

Containerization

- Docker
- Kubernetes

14.2 Deployment Architecture

Recommended Architecture

- API Gateway
- Load Balancer
- Stateless application services
- Dedicated worker services
- Background processing queues
- Managed database cluster

14.3 Monitoring & Observability

Tools:

- Prometheus
- Grafana
- Datadog
- ELK stack
- OpenTelemetry

14.4 Backup Strategy

Requirements:

- Point-in-time recovery
- Daily snapshots
- Cross-region replication
- Backup encryption

14.5 Scalability Strategy

Capabilities:

- Horizontal scaling
- Read replicas
- Queue partitioning
- CDN acceleration

---

15. ENTERPRISE FEATURES & FUTURE IMPROVEMENTS

Planned Enterprise Features

Identity & Provisioning

- SCIM provisioning
- Azure AD sync
- Okta integration
- SAML SSO

Platform Features

- White-labeling
- Custom branding
- Multi-region deployments
- Enterprise reporting

AI & Analytics

- AI wellness assistant
- Predictive analytics
- Wellness risk scoring
- Intelligent recommendations

Advanced Administration

- Custom roles
- Workflow automation
- Policy engine
- Advanced audit exports

---

16. COMPLETE END-TO-END USER JOURNEYS

16.1 New Organization Onboarding Journey

Sequence:

1. Platform admin creates organization.
2. Tenant provisioned.
3. Organization admin invited.
4. SSO configured.
5. Users imported.
6. Experts assigned.
7. Wellness programs activated.
8. Reporting enabled.

16.2 User Invitation to Activation Journey

Sequence:

1. User invited.
2. Email received.
3. Invite accepted.
4. Password/MFA configured.
5. Consent accepted.
6. Profile completed.
7. Access granted.

16.3 Expert Assignment Journey

Sequence:

1. Expert verified.
2. Organization assigns expert.
3. Availability configured.
4. Users notified.
5. Sessions booked.
6. Records maintained.

16.4 Session Booking Journey

Sequence:

1. User selects expert.
2. Availability queried.
3. Session booked.
4. Notifications sent.
5. Session conducted.
6. Notes stored.
7. Follow-up reminders triggered.

16.5 Account Deletion Journey

Sequence:

1. User initiates deletion.
2. MFA verification completed.
3. Grace period begins.
4. Export option presented.
5. Account suspended.
6. Soft deletion executed.
7. Purge queued.

16.6 Organization Offboarding Journey

Sequence:

1. Organization requests closure.
2. Billing finalized.
3. Data export generated.
4. Tenant suspended.
5. Retention countdown initiated.
6. Anonymization performed.
7. Final purge completed.

---

17. BEST PRACTICES & RECOMMENDATIONS

17.1 Industry Best Practices

- Use immutable audit logging.
- Separate authentication from authorization.
- Use defense-in-depth security.
- Automate compliance reporting.
- Adopt infrastructure as code.

17.2 Common Mistakes to Avoid

- Missing tenant validation
- Over-permissioned roles
- Immediate hard deletion
- Shared admin credentials
- Unencrypted backups

17.3 Security Recommendations

- Enforce MFA for admins.
- Use signed URLs for uploads.
- Rotate secrets regularly.
- Monitor anomalous activity.
- Enable centralized logging.

17.4 Compliance Recommendations

- Implement data minimization.
- Track policy version consent.
- Maintain retention schedules.
- Enable export and deletion rights.

17.5 UX Recommendations

- Progressive onboarding
- Clear deletion messaging
- Transparent consent flows
- Session reminders
- Guided activation journeys

17.6 SaaS Scaling Recommendations

- Design stateless services.
- Use asynchronous workflows.
- Separate read/write workloads.
- Adopt queue-based processing.
- Plan for regional expansion.

---

18. FINAL SYSTEM SUMMARY

18.1 Recommended Architecture Summary

Recommended architecture:

- Multi-tenant SaaS
- PostgreSQL + Redis
- NestJS/Django backend
- Kubernetes deployment
- Queue-driven asynchronous processing
- Zero trust security model

18.2 Recommended Deletion Policy

Recommended approach:

- Soft deletion first
- Grace periods mandatory
- Delayed hard purge
- Audit retention preserved
- Analytics anonymized

18.3 Recommended Onboarding Strategy

- Invite-based onboarding
- Enterprise SSO support
- CSV bulk imports
- Domain auto-join
- Guided activation workflows

18.4 Recommended Security Strategy

- MFA for privileged roles
- RBAC + ABAC authorization
- Tenant-aware middleware
- Immutable audit logs
- Encryption everywhere

18.5 Recommended Operational Workflows

Operational priorities:

- Automated provisioning
- Centralized observability
- Compliance automation
- Incident response readiness
- Backup validation

TEXTUAL SYSTEM DESIGN DIAGRAMS

High-Level System Architecture

[Client Apps] ↓ [API Gateway / Load Balancer] ↓ [Authentication Service] ↓ [Application Services] ├── User Service ├── Organization Service ├── Expert Service ├── Session Service ├── Messaging Service ├── Audit Service ├── Compliance Service └── Notification Service ↓ [Redis Cache] [Queue Workers] [PostgreSQL Cluster] [AWS S3 Storage] [Monitoring & Logging Stack]

Multi-Tenant Access Flow

User Request ↓ JWT Validation ↓ Tenant Context Resolution ↓ RBAC Validation ↓ ABAC Ownership Validation ↓ organization_id Query Scoping ↓ Data Retrieval

---

Deletion Lifecycle Pipeline

Deletion Requested ↓ Verification & Approval ↓ Grace Period ↓ Soft Delete ↓ Anonymization ↓ Retention Countdown ↓ Hard Purge

---

CONCLUSION

This document defines a production-grade enterprise wellness SaaS architecture designed for scalability, security, compliance, and operational resilience.

The proposed system emphasizes:

- Secure multi-tenancy
- Enterprise identity integration
- Compliance-first lifecycle management
- Auditability and governance
- Flexible onboarding
- Scalable infrastructure
- Future enterprise extensibility

The architecture is suitable for:

- Enterprise wellness platforms
- Corporate healthcare ecosystems
- Coaching and therapy SaaS products
- Global multi-tenant B2B SaaS deployments

End of Document.