# Software Requirements Specification (SRS)

Project: Hybrid Yoga Training Booking App (Android .apk & iOS .ipa)  
Prepared for: primal-balance  
Prepared by: (Development Team)  
Date: 13 October 2025

1. Introduction  
Purpose  
This SRS describes the functional and non-functional requirements for a hybrid mobile application (single codebase producing APK and IPA) that provides yoga training content, booking and subscription management. The app integrates with Momence (for events, videos, memberships, teachers, products) and Stripe (for payments). It supports a 7-day free trial, single-user and community subscription tiers, an end-user facing mobile UI, and a PHP-based admin panel for backend management.

Scope  
- Mobile application built with a hybrid framework (e.g., React Native / Flutter / Ionic) producing Android APK and iOS IPA.  
- Integrations: Momence API (content & scheduling) and Stripe (payments & subscriptions).  
- Subscription management with free trial, pricing, and renewal rules.  
- Booking module (view/search/book/cancel appointments).  
- Video library limited by user’s assigned program/membership.  
- User accounts with profile, subscription/credits view, bookings, and support.  
- Admin panel (PHP) for managing content, bookings, users, payments, and analytics.

Definitions, acronyms & abbreviations  
- MOM: Momence API (third-party content & scheduling)  
- STRIPE: Stripe payment platform  
- SKU: Stock Keeping Unit or internal product id for subscription  
- API: Application Programming Interface  
- JWT: JSON Web Token  
- CDN: Content Delivery Network

2. Overall description  
Product perspective  
The mobile app is a client of the platform's backend and of Momence and Stripe services. The backend provides user auth, connects to Momence to fetch events/videos/memberships/products/teachers, and uses Stripe to create customers, handle subscriptions and process payments. The admin panel (PHP) is an internal web application that controls content, user accounts, pricing, promotions and accesses reports.

User classes and characteristics  
- End User (Customer): Registers, claims free trial, subscribes, books sessions, watches assigned on-demand videos, views bookings, cancels/reschedules as per policy.  
- Admin: Manages events/products/memberships (via Momence and internal overrides), manages users, payments, refunds and analytics through PHP admin panel.  
- Support Staff: Smaller admin role to view bookings and assist users.

Operating environment  
- Hybrid mobile: Android (API level as per target market; minimum Android 8.0 recommended), iOS 13+.  
- Backend: REST API (Node.js / PHP / Python) served over HTTPS.  
- Database: Relational DB (MySQL/Postgres) + optional Redis for caching.  
- Admin panel: PHP web app hosted on secure server with HTTPS.

Design and implementation constraints  
- Payment card data must be handled per PCI rules — use Stripe SDKs; never store raw card data on own servers.  
- API keys must be kept secret (server-side) and not hard-coded in mobile apps.  
- Momence API token must be stored server-side; mobile app queries go via backend which proxies Momence endpoints.  
- App store rules: iOS in-app purchase (IAP) vs external subscriptions — since Stripe is used, ensure compliance with App Store policies (if selling digital content consumed in-app Apple may require IAP). Legal advice required; document currently assumes external Stripe subscription is acceptable for memberships/subscriptions that include digital content (but client should confirm App Store policy or use Apple IAP for iOS if required).

3. Functional requirements  
Account & Authentication  
FR-1: Users can register via email/password or social login (Google/Apple).  
FR-2: Email verification required on registration.  
FR-3: Login supports JWT-based sessions and refresh tokens.  
FR-4: Password reset via email.

Free Trial & Subscription Flow  
FR-5: New users receive a 7-day free trial automatically upon registration for subscription-eligible plans. Trial begins on first successful registration or on first activation of trial flag.  
FR-6: After trial expires user is prompted to subscribe to continue access.  
FR-7: Subscription pricing: Single-user £9.99/month; Community £19.99/month.  
FR-8: Subscription processing uses Stripe. Create Stripe Customer, attach Payment Method, and create Subscription.  
FR-9: App must display remaining trial days and next billing date.  
FR-10: Users can cancel subscription from the app; cancellation takes effect at period end unless configured otherwise.  
FR-11: Support downgrades/upgrades with proration per Stripe configuration.

Payments & Billing  
FR-12: Integrate Stripe using server-side secret key (do not store secret in client). Use Stripe Publishable key in client for tokenization.  
FR-13: Payment flow must support card entry via Stripe SDK/Elements and 3DS authentication where required.  
FR-14: Implement Stripe webhooks server-side for: invoice.paid, invoice.payment\_failed, customer.subscription.deleted, customer.subscription.updated, payment\_intent.succeeded.  
FR-15: Generate receipts and invoice views in the app.  
FR-16: Admin panel can issue refunds and view payment history.

Booking & Scheduling (Momence)  
FR-17: Users can search and view upcoming events/classes fetched from Momence's /Events endpoint.  
FR-18: Book a class: request is created via backend which communicates with Momence or stores booking locally depending on integration options.  
FR-19: App shows bookings list (upcoming & past), with ability to cancel or reschedule (subject to policy).  
FR-20: Booking confirmations are shown in-app and emailed.  
FR-21: Admin panel shows all bookings and can manage capacity, cancellations and manual bookings.

Video Library  
FR-22: Fetch on-demand videos from Momence /Videos endpoint; filter by user’s assigned program or membership.  
FR-23: Display video metadata, duration, teacher, and access controls.  
FR-24: Streaming handled via secure CDN / signed URLs (Momence provides links; cache minimal meta).  
FR-25: Optionally allow downloading for offline viewing (DRM and policy required). If implemented, downloads are encrypted and tied to user account.

Credits & Memberships  
FR-26: The account page shows subscription status, remaining credits (if credit model used), membership level (from Momence /Memberships).  
FR-27: Products retrieved from Momence /Products may be purchasable; purchasing can be handled via Stripe creating one-time payments or attaching to subscription.

Notifications  
FR-28: Push notifications for booking reminders (configurable: 24h, 1h reminders), payment failures, subscription renewals, admin messages.  
FR-29: In-app notification center showing history.

Admin Panel (PHP)  
FR-30: Authentication & roles: Admin, Support, Content Manager.  
FR-31: Manage users (view/edit/ban), bookings (view/cancel/change), and payments (view/refund/manual charge).  
FR-32: Manage content mapping: assign videos to programs, restrict access by membership.  
FR-33: View reports: revenue, active subscriptions, churn, booking counts by event/teacher.  
FR-34: Sync with Momence (pull events/videos/teachers/memberships/products) and force-refresh cache.

Search & Discovery  
FR-35: Landing page / Home screen provides curated programs, featured teachers, upcoming events and CTA to book or start trial.  
FR-36: Search for events, teachers and videos with filters (date, teacher, level, duration).

4. External interface requirements  
Momence API integration  
All Momence API calls should be proxied via the backend so the token remains secret. Use the following endpoints:  
- Events: GET https://api.momence.com/api/v1/Events?hostId=6663&token=eccb02ed77  
- Videos: GET https://api.momence.com/api/v1/Videos?hostId=6663&token=eccb02ed77  
- Memberships: GET https://api.momence.com/api/v1/Memberships?hostId=6663&token=eccb02ed77  
- Products: GET https://api.momence.com/api/v1/Products?hostId=6663&token=eccb02ed77  
- Teachers: GET https://api.momence.com/api/v1/Teachers?hostId=6663&token=eccb02ed77

Integration rules:  
- Backend must cache results (e.g., TTL 5–15 minutes) and respect any Momence pagination/limits.  
- Implement retry with exponential backoff on transient HTTP 5xx errors.  
- Validate and normalize data before storing or returning to clients.  
- Do not embed eccb02ed77 token in client builds; keep server-side only.

Stripe integration  
- Use the provided publishable key in client for tokenization: pk\_live\_51IVyVFEFQNX9VlqKXrQuYxBaR3vBsZeIbWg9Ev8jYg3zFjNc2m7ddh1UzSZUMl2Fx324hOZubIFT2WmahXgDu5a4002ewlVewn  
- Server must store Stripe secret key in environment variables and never commit to VCS.  
- Recommended flows:  
 - Create Customer: server-side stripe.customers.create() when user first adds payment method.  
 - Create Subscription: server-side stripe.subscriptions.create() with plan ids for single-user and community. Attach trial\_period\_days=7 if trial applied at creation or manage trial locally with metadata.  
 - Payment Method: use Stripe SDK on client to collect card details and return PaymentMethod id to server.  
 - Webhooks: Setup secure webhook endpoint that verifies signatures and processes events.

Backend APIs (examples)  
- POST /api/v1/auth/register — register user (returns JWT)  
- POST /api/v1/auth/login — login  
- GET /api/v1/momence/events — proxied events  
- GET /api/v1/momence/videos — proxied videos  
- POST /api/v1/bookings — create booking  
- GET /api/v1/bookings — list user bookings  
- POST /api/v1/stripe/create-subscription — create subscription server-side  
- POST /webhook/stripe — receive Stripe events  
- GET /api/v1/admin/reports — admin-only

5. Non-functional requirements  
Performance  
- API response times: 95th percentile < 500ms for core endpoints when not waiting on third-party calls.  
- Mobile app cold start < 3 seconds on modern devices.

Security  
- All traffic over HTTPS/TLS 1.2+.  
- Store secrets only on server environment variables or secret manager.  
- Use rate-limiting and IP blacklists to prevent abuse.  
- Comply with GDPR (data deletion on request) and applicable data protection laws.  
- PCI: Avoid storing card data; use Stripe Checkout/Elements and server-side tokenization.

Reliability & Availability  
- Backend uptime target: 99.5% monthly.  
- Webhook processing must be idempotent (use event ids to track processed events).

Scalability  
- System designed to scale horizontally: stateless app servers + shared DB + CDN for media.

Maintainability  
- Codebase follows modular architecture, documented API specs (OpenAPI/Swagger) and unit/integration tests for critical flows.

Accessibility  
- App UI should meet WCAG AA basics: contrast, scalable fonts, accessible labels for screen readers.

6. UI / UX Requirements  
Landing (Home) Screen  
- Brilliant, modern UI with hero banner and CTA: "Start 7-day free trial"  
- Sections: Featured programs, Upcoming live classes, Recommended videos, Teachers, Testimonials.

Tab Bar (primary navigation)  
- Home | Bookings | Video Library | Account

Booking Screen  
- Calendar/ list view of events, filter controls, ability to book in 2-3 taps.  
- Booking confirmation modal showing event details, refund/cancellation policy, and add to calendar.

Video Library  
- Organized by Program → Module → Video. Player supports resume playback and bookmarks.

Account Screen  
- Profile, subscription status, remaining trial days, payment method, invoices, logout.

Admin Panel (PHP)  
- Dashboard cards: Active Subscribers, Revenue (30d/90d), Upcoming Events, Pending Bookings, System Health.  
- CRUD UI for sync with Momence and manual overrides.

7. Data model (high level)  
- User: id, name, email, hashed\_password, role, stripe\_customer\_id, trial\_start, trial\_end, membership\_id, created\_at  
- Subscription: id, user\_id, stripe\_subscription\_id, plan\_type, status, current\_period\_end, canceled\_at  
- Booking: id, user\_id, momence\_event\_id, status, booked\_at, canceled\_at  
- Video: id, momence\_video\_id, title, url, duration, teacher\_id, assigned\_programs  
- Payment: id, user\_id, stripe\_payment\_intent\_id, amount, currency, status, created\_at

8. Error handling & edge cases  
- Third party failures: show friendly messages and queue background retries.  
- Stripe payment failures: inform user, notify admin and retry logic for subscription renewal notifications.  
- Conflicting bookings (capacity full): perform server-side capacity check and return clear error.

9. Logging & Monitoring  
- Aggregate logs with correlation IDs. Monitor API latency, errors, webhook delivery failures.  
- Alerts for payment webhook failures, elevated chargebacks, or sync failures with Momence.

10. Legal / Compliance  
- Terms of Service & Privacy Policy must be accepted on registration.  
- Clear refund policy for subscriptions and bookings.  
- If using Stripe Connect or handling VAT, ensure correct tax treatment and compliance.

11. Deployment & DevOps  
- CI/CD pipeline to build Android and iOS artifacts, run tests and publish to Play Store / App Store (subject to App Store review).  
- Separate environments: dev, staging, production with separate Momence tokens and Stripe keys.

12. API & Integration Appendix (sample requests)  
Sample Momence request (server-side proxy)  
GET https://api.momence.com/api/v1/Events?hostId=6663&token=eccb02ed77

Sample Stripe flow (server-side pseudocode)  
// On server  
const customer = await stripe.customers.create({ email: user.email });  
// Client uses Stripe SDK to get payment\_method\_id, sends to server  
const subscription = await stripe.subscriptions.create({  
 customer: customer.id,  
 items: [{ price: PRICE\_ID\_SINGLE\_OR\_COMMUNITY }],  
 trial\_period\_days: 7, // if using Stripe trial  
});

13. Acceptance criteria  
- User can register and claim 7-day trial.  
- User can complete subscription payment via Stripe and access restricted videos.  
- User can successfully book an event fetched from Momence and see it in Booking tab.  
- Admin can view & manage bookings and refund payments.  
- Webhooks correctly change subscription/payment status in DB.

14. Risks & Mitigations  
- App Store policies vs external payments: Seek legal/product guidance and consider using Apple IAP if required.  
- Secret leakage: Use server-side proxies, secret managers, and code reviews.  
- Third-party downtime: Cache data and provide degraded UX with helpful messages.

15. Next steps (recommended)  
1. Confirm mobile framework (React Native / Flutter / Ionic).  
2. Prepare Stripe product & price IDs and test keys.  
3. Confirm App Store payment compliance for iOS.  
4. Build minimal viable backend endpoints & webhook handling.  
5. Implement PHP admin panel scaffold and integrate Momence sync.

Security reminder: The publishable Stripe key and Momence token you provided should be treated carefully. Publishable keys (pk\_...) are safe to use in mobile clients but do not expose secret keys. Momence token eccb02ed77 must not be embedded in client apps; always proxy via server.