Software Requirements Specification (SRS)

Digital E-Book Subscription & Author Marketplace Platform

Prepared by: Business Analyst (generated by assistant)

Version: 1.1

Date: 2025-12-06

Table of Contents

1. Introduction

1.1 Purpose

This SRS defines the requirements for a Digital E-Book Subscription & Author Marketplace Platform. It is intended for backend and frontend developers, QA, DevOps, and stakeholders.

1.2 Scope

The platform will support user registration, subscription plans, one-time purchases, an author dashboard for content management and analytics, admin workflows, and a secure online reader. Payment processing, storage, and basic analytics are included.

1.3 Definitions and Acronyms

JWT: JSON Web Token.
API: Application Programming Interface.
S3: Object storage service.
MVP: Minimum Viable Product.

2. System Features Summary

High-level modules:

• Authentication & Authorization (JWT with refresh tokens).

• Subscription management (plans, billing, renewals).

• One-time e-book purchases with receipts and order history.

• Author dashboard: upload, manage, analytics, earnings.

• Admin panel: approve authors/books, manage plans and users.

• Secure online reader with signed URLs and reading analytics.

• Background jobs (cron or queue) for renewals, payouts, and analytics aggregation.

3. User Roles & Permissions

Roles and allowed actions:

Role	Capabilities / Constraints
Guest	Browse catalog, search, view book metadata, register/login.
Registered User	Purchase books, subscribe to plans, read/persist reading progress, view order history.
Author	Apply/Manage author profile, upload books, set price, view sales & analytics, request withdraw
Admin	Manage users, approve/reject authors and books, create plans, view system analytics, manag

4. Functional Requirements (detailed)

4.1 Authentication & User Management

FR-A1: Registration with email verification (optional).

FR-A2: Login returns short-lived access token (15m) and refresh token (30d).

FR-A3: Password reset flow via secure one-time token.

FR-A4: Role-based access control middleware for API routes.

FR-A5: Session invalidation on password change.

4.2 E-Book Management

FR-B1: Authors can upload book metadata and PDF (max 50MB). Files stored in S3 (or Spaces).

FR-B2: Admin approves book before it is public.

FR-B3: Book statuses: draft, pending_review, published, rejected, unlisted.

FR-B4: Versioning: support minor metadata updates without invalidating purchase access.

FR-B5: Cover images and sample (first 10 pages) attachments.

4.3 Subscription System

FR-S1: Admin creates plans (monthly/yearly) with trial option and limits.

FR-S2: Integrate payment gateway (Stripe recommended; Paymob optionally for Egyptian market).

FR-S3: Subscription purchase activates access immediately and creates billing records.

FR-S4: Auto-renewal using saved payment method; system retries failed payments up to 3 times with exponential backoff.

FR-S5: Cancellation: user can cancel to stop auto-renew while keeping access until end_date.

4.4 One-time Purchase Flow

FR-P1: Cart & Checkout for single or multiple e-books (MVP can be single-book checkout).

FR-P2: Instant access after successful payment; store order and payment metadata.

FR-P3: Refund policy: Admin/manual refunds (automated refunds optional, need payment provider support).

FR-P4: Email receipts and downloadable invoice PDF.

4.5 Online Reader & Analytics

FR-R1: Reader uses signed short-lived URLs to stream PDFs; direct URLs hidden.

FR-R2: Track reading events: open, close, pages_read, time_spent, last_page.

FR-R3: Limit concurrent device sessions if required (configurable).

FR-R4: Allow bookmarking and resume position per user per book.

4.6 Author Dashboard

FR-AU1: Author application form with verification fields (ID, bio).

FR-AU2: Dashboard lists books, sales, top books, views, conversion rates.

FR-AU3: CSV export for sales reports (date range).

FR-AU4: Manage book pricing and promotions (discount codes optional).

FR-AU5: Earnings calculation: platform fee (configurable %) and author net revenue.

5. Non-functional Requirements

Performance

• 95th percentile API response time < 500ms under normal load. • Reader streaming should start within 1s for first page.

Scalability

• Use stateless backend; session via JWT. Scale via load balancer and multiple instances. • Store files in S3-compatible storage; use CDN for reader content.

Security

• Use bcrypt (or argon2) for password hashing.

• JWT with rotating refresh tokens and revocation list.

• Rate limiting per IP and per user (e.g., 100 req/min default).

• Input validation and sanitization for all endpoints.

• Encrypt sensitive data at rest if required (PCI concerns).

Reliability & Availability

• Daily backups for DB; transactional logs retained 30 days. • Health checks and alerting for critical services.

Compliance

• GDPR considerations: ability to delete user data, export data on request. • PCI: do not store raw card data; use external payment gateway's tokenization.

6. Data Model (detailed schema)

Primary tables and key columns. Data types are relational (Postgres).

Table	Columns (summary)
users	id (uuid), name, email (unique), password_hash, role, is_author(bool), created_at, last_login
authors	id, user_id (fk users.id), bio, national_id, status(enum), payout_account, created_at
books	id, author_id, title, slug, description, category_id, price_cents (int), currency, cover_image_url
plans	id, name, interval(enum monthly/yearly), price_cents, trial_days, max_books_per_period (null
subscriptions	id, user_id, plan_id, start_date, end_date, status, auto_renew(bool), payment_provider_subsc
orders	id, user_id, amount_cents, currency, status, payment_provider_charge_id, created_at
order_items	id, order_id, book_id, price_cents, created_at
reading_progress	id, user_id, book_id, last_page, progress_percent, time_spent_seconds, updated_at
payments	id, user_id, provider, provider_id, amount_cents, status, metadata, created_at
author_earnings	id, author_id, book_id, gross_cents, platform_fee_cents, net_cents, settled(bool), created_at

7. API Endpoints (detailed)

Selected endpoints, method, brief request/response structure.

POST /auth/register

```
Request: { name, email, password } Response: { user: {id, name, email}, accessToken, refreshToken }
```

POST /auth/login

```
Request: { email, password } Response: { accessToken, refreshToken }
```

GET /books

```
Query params: ?q=&category;=&author;=&page;=&limit;= Response: { data: [books], meta: {page, total} }
```

GET /books/:id

```
Response: { book: metadata, isPurchased: bool, isSubscribed: bool }
```

POST /author/books

```
Auth: Author token Request (multipart): metadata + pdf file Response: { bookId, status:
'pending_review' }
```

POST /subscriptions/subscribe

```
Request: { plan_id, payment_method_id } Response: { subscription_id, status }
```

POST /checkout

```
Request: { book_id, payment_method_id } Response: { order_id, status }
```

GET /reader/:book_id

```
Auth: User Response: signed_url (short-lived) or streaming endpoint
```

8. Security & Compliance

Detailed security controls and compliance notes.

• Do not store cardholder data; use PCI-compliant gateways.

• Maintain audit logs for admin actions for at least 90 days.

• Protect file uploads: virus-scan on upload (ClamAV or third-party).

• Signed URL expiry times: reader URLs expire in <= 2 minutes; download URLs expire in <= 24 hours and require additional checks.

9. Operational Requirements & Deployment

Deployment recommendations:

• Deploy in containers (Docker) behind a load balancer.

• Use managed Postgres (e.g., AWS RDS) and managed Redis for queues and caching.

• Use CI/CD for automated tests and deploy pipelines (GitHub Actions).

• Monitoring: Prometheus + Grafana or hosted monitoring. Alerts for payment failures and cron job errors.

10. Acceptance Criteria & High-level Test Cases

AC1: A registered user can subscribe to a monthly plan and immediately access all published books.

AC2: An author can upload a PDF and see it in 'pending' status; admin can approve and it becomes visible to users.

AC3: A successful one-time purchase grants permanent access to that book for the user.

AC4: Reader streaming uses signed URLs and does not expose raw S3 keys.

AC5: Failed subscription renewal retries and emails the user after final failure.

11. Project Plan & Milestones (suggested)

Sprint 0 (1 week): Requirements finalization, repo setup, CI, infra design. Sprint 1 (2 weeks): Auth, User model, basic books listing, DB migrations. Sprint 2 (2 weeks): Author flow (apply, upload), Admin approval, file storage integration. Sprint 3 (2 weeks): Checkout and payments (one-time), orders, receipts. Sprint 4 (2 weeks): Subscription flows, billing, renewal cron and retry logic. Sprint 5 (2 weeks): Online reader, reading analytics, bookmarks. Sprint 6 (1 week): Polish, tests, deployment pipeline, basic monitoring.

12. Appendices

Glossary

Include terms like JWT, S3, CDN, CDN signed URLs, MVC, API, MVP.

Contacts

Project Owner: TBD. Dev Team: TBD. Admin: TBD.