📄 PRODUCT REQUIREMENTS DOCUMENT (PRD)

Project Title: Data Bundle Sales Web Application
Prepared by: LucienTech (Developer)
Client Role: System Owner (Super Admin)
Date: April 28, 2025

---

1. Project Overview

The Data Bundle Web Application enables vendors to sell mobile data bundles to customers online. The platform supports a hierarchical vendor system where vendors can create sub-vendors (pending admin approval). Customers purchase data via vendor-specific links and pay using mobile money. All orders are fulfilled manually by the super admin. Vendors earn profits by adding margins to base prices set by the admin and can withdraw their earnings automatically.

---

2. User Roles & Permissions

2.1 Super Admin

- Approves/rejects vendor and sub-vendor applications
- Manages all mobile network providers and bundle offerings
- Manually completes customer orders
- Views total vendor performance, sales, and withdrawal history
- Approves vendor withdrawals automatically
- Has access to analytics and platform settings

2.2 Vendor

- Registers with personal and MoMo info
- Sets prices for bundles (≥ base price)
- Shares personal vendor link for customer ordering
- Creates sub-vendors (admin approval required)
- Views sub-vendor performance
- Withdraws profits to MoMo
- Tracks orders and total profit

2.3 Customer

- Does not need to register
- Places orders as guest using vendor link
- Pays via mobile money (instant payment confirmation)
- Sees order status updates

---

3. Functional Requirements

3.1 Admin Panel

- ✅ Create, update, and delete **Network Providers** and **Data Bundles**
- ✅ View and manage all **Vendor** and **Sub-vendor** accounts
- ✅ Approve or ban vendors
- ✅ Receive **dashboard notifications** for new orders
- ✅ Mark orders as “Completed”
- ✅ Auto-approve vendor **withdrawal requests**

- ✅ View, search, and export withdrawal history (CSV/PDF)
- ✅ Analytics: total vendors, sales, withdrawals, top vendors

3.2 Vendor Dashboard

- ✅ Signup with full profile (Full Name, Email, Phone, Password, MoMo Number)
- ✅ Login/logout securely
- ✅ Edit profile (except email)
- ✅ Set custom prices ≥ base price
- ✅ View profit (set price - base price) × quantity sold
- ✅ Create sub-vendors
- ✅ View sub-vendor performance
- ✅ View customer orders placed through their link
- ✅ Request withdrawal (auto-approved if MoMo matches)

3.3 Customer Flow

- ✅ Guest order form (no registration)
- ✅ Provide: recipient number, network, data package
- ✅ Pay via MoMo (trigger payment confirmation)
- ✅ See in-app confirmation: "Your order is being processed"
- ✅ Track order status (Pending → Completed)

---

4. Business Logic & Rules

- Vendors cannot set a price lower than the base price
- Profit = vendor’s price - base price
- Vendors can withdraw **any amount** of profit
- MoMo withdrawal number must match profile
- Sub-vendor creation must be approved by admin
- All vendor links are unique (e.g. `vendorapp.com/v/{username}`)

---

5. Technical Stack

Layer	Tool / Tech
Frontend	React.js + TailwindCSS
Backend	Node.js + Express.js
Database	PostgreSQL via Supabase
Auth	JWT (Admin/Vendor), session-based for customers
Hosting	Frontend: Vercel, Backend: Render or Railway
Payment	Flutterwave or Paystack (for MoMo confirmation)

---

6. Core APIs (Suggestions)

Endpoint	Method	Description
`/api/admin/networks`	CRUD	Manage networks and bundles
`/api/vendors`	POST	Vendor signup
`/api/vendors/login`	POST	Vendor login
`/api/vendors/price/:bundleId`	PATCH	Set or update price

`/api/orders`	POST	Place order (customer)
`/api/admin/orders`	GET/PATCH	View and mark orders complete
`/api/withdrawals`	POST	Vendor withdrawal request
`/api/admin/withdrawals`	GET	View withdrawal history

7. Database Tables (Simplified)

- **users** (id, name, email, phone, role, momo_number, password_hashed)
- **vendors** (id, user_id, parent_vendor_id, approved)
- **networks** (id, name)
- **bundles** (id, network_id, size, base_price)
- **vendor_prices** (vendor_id, bundle_id, custom_price)
- **orders** (id, vendor_id, customer_phone, bundle_id, status, paid, created_at)
- **withdrawals** (id, vendor_id, amount, momo_number, status, created_at)

8. Notifications

User	Event	Type
Admin	New order placed	Dashboard
Vendor	Withdrawal completed	In-app only
Customer	Order placed	In-app message
Customer	Order marked complete	Status updated UI

9. Security

- Passwords hashed using bcrypt
- JWT-based authentication for all users
- MoMo number re-confirmation before withdrawal
- No 2FA for now (planned for future)

---

10. Branding

- Style: Minimal, clean
- Fonts: Inter or Poppins
- Color palette: Suggest neutral with one bold primary (e.g., #2A2D34 + #FF6B35)

---

11. Development Timeline

Week	Dates	Phase	Goals & Deliverables
Week 1	Apr 29 – May 5	📌 Planning & Architecture	- Finalize PRD- Create wireframes/UI mockups- Define DB schema and API structure
Week 2	May 6 – May 12	🎨 Frontend (Customer Side)	- Build landing/order page- Implement guest checkout- MoMo payment flow (mocked)
Week 3	May 13 – May 19	🔧 Backend Core (Auth & Orders)	- Setup Node/Express API- JWT auth (vendor & admin)- Customer order logic
Week 4	May 20 – May 26	💻 Vendor Dashboard	- Vendor signup/login- Price management UI- Order history- Profit tracking

Week 5	May 27 – Jun 2	🛠️ Admin Dashboard (Part 1)	- Manage vendors- Approve/reject sub-vendors- View customer orders
Week 6	Jun 3 – Jun 9	🛠️ Admin Dashboard (Part 2)	- Create/edit/delete bundles- View analytics (sales, vendors)- Withdrawals logic
Week 7	Jun 10 – Jun 16	🧪 Integration & Testing	- Connect frontend to backend- Test vendor & admin flows- Fix bugs, optimize
Week 8	Jun 17 – Jun 23	🚀 Launch & Deployment	- Deploy to production- Final UI polish- Go live, start onboarding vendors

12. Optional Future Features

- SMS/email alerts
- Order fulfillment automation
- Referral bonuses for vendors
- In-app chat support for all users