CareAl Payment System Documentation

Overview

The CareAl Payment System is a comprehensive, multi-provider payment solution designed specifically for healthcare services in Africa. It supports multiple payment methods including Stripe, Mobile Money (MTN, Airtel), and Orange Money, with XAF currency support and regional pricing optimization.

Features

Regional Pricing

• Africa: Base pricing (cheapest rates)

• Europe: 2x Africa pricing

North America: 2.5x Africa pricing
 Other regions: 2x Africa pricing

Payment Methods

1. Stripe Integration

- Credit/Debit card payments
- Real-time validation
- PCI DSS compliant
- Global coverage

2. Mobile Money

- MTN Mobile Money
- Airtel Money
- Optimized for African markets
- Instant payments

3. Orange Money

- Francophone Africa coverage
- Secure mobile payments
- Wide acceptance

Currency Support

- Primary currency: XAF (Central African Franc)
- Automatic currency conversion
- Regional exchange rate optimization

Subscription Plans

Free Plan

- Price: Free forever
- Features:
 - 3 health documents storage
 - Basic chatbot access
 - Appointment booking
 - Basic health tracking

Basic Plan

• Price: 2,900 XAF/month (Africa)

- Features:
 - 25 health documents storage
 - Enhanced chatbot features
 - Priority appointment booking
 - Advanced health analytics
 - Email support

Premium Plan

• Price: 8,700 XAF/month (Africa)

- Features:
 - 100 health documents storage
 - AI-powered health insights
 - Telemedicine consultations
 - Family health management
 - Priority support
 - Custom health reports

Enterprise Plan

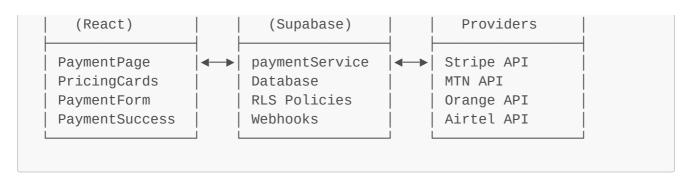
• Price: 29,000 XAF/month (Africa)

- Features:
 - Unlimited health documents
 - Advanced AI analytics
 - Multi-user management
 - API access
 - Custom integrations
 - 24/7 dedicated support
 - Compliance reporting

Technical Implementation

Architecture

Frontend	Backend	Payment	



Database Schema

Core Tables

- payment_transactions: Payment records and status
- user_subscriptions: Active user subscriptions
- user_payment_methods: Saved payment methods
- subscription_plans: Available plans and pricing
- regional_pricing: Region-specific pricing
- payment_webhooks: Webhook event handling

Security Features

1. Row Level Security (RLS)

- Users can only access their own payment data
- Automatic user isolation

2. Encryption

- 256-bit SSL encryption
- PCI DSS compliance for card data
- Secure token storage

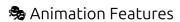
3. Validation

- Real-time form validation
- Phone number validation for mobile money
- · Card number validation with Luhn algorithm

UI/UX Features



- Smooth Animations: Framer Motion powered
- Physics Effects: Spring animations and micro-interactions
- Responsive Design: Mobile-first approach
- Dark Mode: Full dark mode support
- Accessibility: WCAG 2.1 compliant



- Staggered card animations
- Floating elements with physics
- · Progress indicators with smooth transitions
- Success celebrations with confetti
- Hover effects with spring physics

Mobile Optimization

- Touch-friendly interfaces
- Optimized for African mobile networks
- Progressive Web App (PWA) support
- Offline capability for form validation

Integration Guide

1. Setup Payment Service

```
import { paymentService } from '../lib/paymentService';

// Get available plans for user's region
const plans = paymentService.calculatePricing('africa');

// Get available payment methods
const methods = paymentService.getPaymentMethods('africa');
```

2. Process Payment

```
const paymentData = {
   plan_id: 'premium',
   payment_method: 'mtn_mobile_money',
   amount_xaf: 8700,
   user_id: user.id,
   region: 'africa',
   phone_number: '+237612345678'
};

const result = await paymentService.processPayment(paymentData);
```

3. Check Document Limits

```
const usage = await paymentService.canUploadDocument(userId);
console.log(`${usage.currentCount}/${usage.limit} documents used`);
```

Component Usage

PROI

PaymentPage Component

```
import { PaymentPage } from '../components/payment/PaymentPage';

<PaymentPage
  preSelectedPlan="premium"
  onClose={() => setShowModal(false)}
/>
```

Subscription Management

```
import SubscriptionPage from '../pages/SubscriptionPage';

// Full subscription management page
<Route path="/subscription" element={<SubscriptionPage />} />
```

Regional Considerations

Africa-Specific Features

- Mobile Money Priority: Mobile money options shown first
- Local Currency: XAF pricing prominently displayed
- **Network Optimization**: Optimized for slower connections
- Offline Support: Form validation works offline

Localization

- Languages: English and French support
- Cultural Adaptation: Payment flows adapted for African markets
- Regional Compliance: Meets local financial regulations

Testing

Demo Mode

- Visit /payment demo for interactive demonstration
- All payments are simulated (no real charges)
- Test all payment flows and animations

Test Data

```
// Test card numbers (Stripe test mode)
const testCards = {
  visa: '42424242424242',
  mastercard: '555555555554444',
```

```
declined: '40000000000000002'
};

// Test mobile money numbers
const testMobile = {
  mtn: '+237612345678',
  orange: '+237698765432'
};
```

Deployment

Environment Variables

```
# Stripe
VITE_STRIPE_PUBLISHABLE_KEY=pk_test_...
STRIPE_SECRET_KEY=sk_test_...

# Mobile Money APIs
MTN_API_KEY=your_mtn_api_key
ORANGE_API_KEY=your_orange_api_key
AIRTEL_API_KEY=your_airtel_api_key

# Database
SUPABASE_URL=your_supabase_url
SUPABASE_ANON_KEY=your_supabase_anon_key
```

Database Setup

- 1. Run the payment schema: database/payment_schema.sql
- 2. Configure RLS policies
- 3. Set up webhook endpoints

Monitoring

Key Metrics

- · Payment success rates by method
- Regional conversion rates
- Average transaction values
- User upgrade patterns

Error Handling

- Automatic retry logic
- Graceful degradation
- User-friendly error messages
- Comprehensive logging

Support

For Users

- In-app help system
- Email support for payment issues
- Phone support for premium users

For Developers

- Comprehensive API documentation
- Webhook testing tools
- Sandbox environment
- · Technical support

Roadmap

Upcoming Features

- Cryptocurrency Support: Bitcoin and stablecoins
- Bank Transfer: Direct bank account payments
- Installment Plans: Split payments over time
- Corporate Billing: Enterprise invoicing
- Multi-Currency: Support for more African currencies

Performance Improvements

- Caching: Redis for pricing data
- CDN: Global content delivery
- Optimization: Bundle size reduction
- Analytics: Advanced payment analytics

Quick Start

PROF

1. Install Dependencies

npm install framer-motion lucide-react

2. Setup Database

-- Run database/payment_schema.sql

3. Configure Environment

Add payment provider keys

4. Test Payment Flow

Visit /payment-demo

5. Go Live

Switch to production keys

+8/8+

For detailed implementation examples, see the /src/components/payment/ directory.