

# LidaPay Mobile App - MVP Software Requirements Specification

## Document Information

| Document Version | 1.0                        |
|------------------|----------------------------|
| Date             | January 8, 2026            |
| Author           | Development Team           |
| Project          | LidaPay Mobile Application |
| Target Platform  | Flutter (iOS & Android)    |

## 1. Introduction

### 1.1 Purpose

This document defines the functional and non-functional requirements for the LidaPay Mobile Application Minimum Viable Product (MVP). LidaPay is a mobile application designed for international and local airtime and internet data remittance, specifically targeting the Ghana market.

### 1.2 Scope

The MVP will focus on core functionalities that enable users to:

- Register and authenticate securely
- Purchase airtime for Ghanaian mobile networks
- Purchase data bundles for Ghanaian mobile networks
- Manage transactions and view history
- Handle payments securely

### 1.3 Target Audience

- **Primary:** Users in Ghana requiring airtime/data services

- **Secondary:** Ghanaians living abroad sending airtime/data to family/friends
- **Age Range:** 16+ years
- **Technical Proficiency:** Basic to intermediate mobile app users

## 1.4 Business Objectives

- Provide fast, reliable airtime and data remittance services
  - Support all major Ghanaian mobile networks
  - Ensure secure payment processing
  - Achieve 99.9% service availability
  - Scale to 10,000+ concurrent users
- 

# 2. System Overview

## 2.1 Application Architecture

- **Frontend:** Flutter 3.x with Dart null-safety
- **State Management:** Riverpod
- **Navigation:** go\_router
- **Networking:** Dio with Retrofit
- **Local Storage:** SharedPreferences
- **Backend Integration:** NestJS API at <https://api.advansistechnologies.com>

## 2.2 Supported Platforms

- **Android:** Minimum SDK 21 (Android 5.0) - Target SDK 34
- **iOS:** Minimum iOS 12.0 - Target iOS 17.0

## 2.3 External Integrations

- **Reloadly Global API:** International airtime purchases
- **Prymo API:** Ghana local airtime purchases
- **ExpressPay Ghana:** Payment gateway processing

## 2.4 Architecture Design (MVP)

### 2.4.1 Logical Architecture (Layered)

The MVP follows a layered architecture aligned with the project structure:

**Presentation Layer (Flutter UI)**

- Screens / Widgets (Features: Auth, Dashboard, Airtime/Data, Transaction)
- State Management (Riverpod Providers / Notifiers)

**Domain Layer (Business Rules)**

- Entities (User, Transaction, Purchase)
- Use-cases (Login, Purchase Airtime, Purchase Data, Verify Payment, Fetch)

**Data Layer (Integration)**

- Repositories (AuthRepository, AirtimeRepository, DataRepository, PaymentRepository)
- Remote Datasources (Retrofit/Dio API Client)
- Local Storage (SharedPreferences for tokens and basic preferences)

**2.4.2 System Context (MVP)****2.4.3 Deployment View (MVP)**

- **Mobile App:** Flutter app running on iOS/Android
- **API:** NestJS backend exposed over HTTPS
- **External Providers:** Reloadly/Prymo/ExpressPay accessed by the backend
- **Storage:**
  - Mobile: SharedPreferences (tokens, preferences)
  - Backend: Database and provider integrations (out of scope for mobile MVP SRS)

**2.4.4 Key Sequence Flows (MVP)**

## A. Login & Token Refresh

User -> Mobile App: Enter credentials  
Mobile App -> Backend: POST /api/v1/users/login  
Backend -> Mobile App: access\_token + refresh\_token + user  
Mobile App: Store tokens securely (local encrypted storage strategy recomm  
Mobile App -> Backend (later): POST /api/v1/auth/refresh (when access toke

## B. Airtime/Data Purchase (Payment + Fulfillment)

User -> Mobile App: Select recipient, product, amount  
Mobile App -> Backend: POST /api/v1/payment/expresspay/initiate  
Backend -> Mobile App: Payment URL / payment reference  
Mobile App -> ExpressPay (WebView/Browser): User completes payment  
Mobile App -> Backend: POST /api/v1/payment/expresspay/verify  
Backend -> Provider (Reloadly/Prymo): Fulfill airtime/data  
Backend -> Mobile App: Transaction result + receipt

# 3. Functional Requirements

## 3.1 User Authentication (Priority: High)

### 3.1.1 User Registration

- **Requirement:** Users must be able to create a new account
- **Input Fields:**
  - First Name (required, max 50 chars)
  - Last Name (required, max 50 chars)
  - Email (required, valid email format)
  - Phone Number (required, Ghana format: +233XXXXXXXXX)
  - Password (required, min 8 chars, 1 uppercase, 1 number, 1 special char)
  - Country (required, default: Ghana)
  - Account Type (required: User/Merchant/Agent)
- **Validation:** Real-time field validation with error messages
- **API Endpoint:** POST /api/v1/users/register
- **Success Response:** User object with profile data
- **Error Handling:** Duplicate email/phone detection with appropriate messages

### 3.1.2 User Login

- **Requirement:** Registered users must be able to authenticate
- **Input Methods:**
  - Email/Username + Password
  - Phone Number + Password
- **API Endpoint:** POST /api/v1/users/login
- **Response:** Access token, refresh token, user object
- **Session Management:** Persistent authentication with token refresh
- **Remember Me:** Option to stay logged in (30 days)

### 3.1.3 Password Recovery

- **Requirement:** Users must be able to reset forgotten passwords
- **Flow:** Email → Security Question → New Password
- **API Endpoint:** POST /api/v1/users/forgot-password
- **Security:** Rate limiting to prevent abuse

## 3.2 Dashboard (Priority: High)

### 3.2.1 Balance Display

- **Requirement:** Display user's wallet balance prominently
- **Features:**
  - Real-time balance updates
  - Multiple currency support (USD, EUR, GHS)
  - Last updated timestamp
  - Refresh functionality
- **API Endpoint:** GET /api/wallet/balance
- **Update Frequency:** Every 30 seconds or on pull-to-refresh

### 3.2.2 Quick Actions

- **Requirement:** Quick access to main functions
- **Actions Available:**
  - Send Airtime
  - Buy Data
  - View Transactions
  - More Options
- **UI:** Grid layout with icons and labels

### 3.2.3 Recent Transactions

- **Requirement:** Display last 5 transactions

- **Information Shown:**
  - Transaction type (airtime/data)
  - Recipient/Network
  - Amount
  - Status (completed/pending/failed)
  - Date/time
- **Interaction:** Tap to view full details

### 3.3 Airtime Purchase (Priority: High)

#### 3.3.1 Network Selection

- **Requirement:** Select target mobile network
- **Supported Networks:**
  - MTN Ghana
  - Vodafone Ghana
  - AirtelTigo Ghana
  - Glo Ghana
- **UI:** Grid with network logos and names

#### 3.3.2 Recipient Input

- **Requirement:** Enter recipient phone number
- **Features:**
  - Phone number validation (Ghana format)
  - Contact picker integration
  - Recent recipients list
  - Save recipient option

#### 3.3.3 Amount Selection

- **Requirement:** Enter or select airtime amount
- **Features:**
  - Manual amount input
  - Preset amounts (GHS 1, 5, 10, 20, 50)
  - Real-time currency conversion
  - Fee calculation display
- **Validation:** Minimum GHS 1, Maximum GHS 1000

#### 3.3.4 Payment Processing

- **Requirement:** Secure payment processing
- **Payment Methods:**

- ExpressPay Ghana (primary)
- Mobile Money (future)
- Bank Cards (future)
- **Flow:**
  1. Payment method selection
  2. Payment initiation
  3. Redirect to payment provider
  4. Payment verification
  5. Transaction completion

### 3.3.5 Transaction Confirmation

- **Requirement:** Provide transaction receipt
- **Receipt Details:**
  - Transaction ID
  - Recipient number
  - Amount paid
  - Network operator
  - Timestamp
  - Status
- **Actions:**
  - Share via WhatsApp/SMS/Email
  - Save to device
  - View in transaction history

## 3.4 Data Bundle Purchase (Priority: High)

### 3.4.1 Bundle Selection

- **Requirement:** Select data bundle by network
- **Bundle Types:**
  - Daily bundles
  - Weekly bundles
  - Monthly bundles
  - Special promotional bundles
- **Information Display:**
  - Data volume (GB/MB)
  - Validity period
  - Price
  - Network operator

### 3.4.2 Purchase Flow

- **Requirement:** Similar to airtime purchase flow
- **Additional Features:**
  - Data balance estimation
  - Bundle comparison
  - Auto-renewal options (future)

## 3.5 Transaction Management (Priority: Medium)

### 3.5.1 Transaction History

- **Requirement:** View complete transaction history
- **Features:**
  - Filter by type (airtime/data)
  - Filter by status (completed/pending/failed)
  - Date range selection
  - Search functionality
  - Pagination (20 items per page)
- **API Endpoint:** GET /api/v1/transactions

### 3.5.2 Transaction Details

- **Requirement:** View detailed transaction information
- **Details Include:**
  - Full transaction receipt
  - Payment method used
  - Processing time
  - Error messages (if failed)
  - Support contact information

## 3.6 User Profile (Priority: Medium)

### 3.6.1 Profile Management

- **Requirement:** View and edit user profile
- **Editable Fields:**
  - First Name
  - Last Name
  - Email
  - Phone Number
  - Profile Picture
- **Read-only Fields:**
  - User ID



- Account Type
- Registration Date

### 3.6.2 Settings

- **Requirement:** Application preferences
  - **Settings Categories:**
    - Account Settings
    - Payment Methods
    - Notifications
    - Security
    - Language (English, Twi, Ewe, Ga, Hausa)
    - Theme (Light/Dark/System)
- 

## 4. Non-Functional Requirements

### 4.1 Performance Requirements

#### 4.1.1 Response Times

- **API Calls:** < 3 seconds for 95% of requests
- **App Launch:** < 3 seconds cold start, < 1 second warm start
- **Screen Transitions:** < 500ms
- **Payment Processing:** < 30 seconds end-to-end

#### 4.1.2 Concurrent Users

- **Target:** 10,000 concurrent users
- **Peak Load:** 50,000 concurrent users during promotions
- **Load Testing:** Must handle 1000 TPS (transactions per second)

### 4.2 Security Requirements

#### 4.2.1 Data Protection

- **Encryption:** AES-256 for sensitive data at rest
- **Transmission:** TLS 1.3 for all API communications
- **Authentication:** JWT tokens with 15-minute expiry
- **Password Security:** bcrypt hashing with salt

#### 4.2.2 Payment Security

- **PCI DSS:** Compliance for payment processing
- **Tokenization:** Payment method tokenization
- **3D Secure:** Support for 3D Secure authentication
- **Fraud Detection:** Basic fraud detection rules

#### 4.2.3 Privacy

- **GDPR Compliance:** User data handling per GDPR
- **Data Minimization:** Collect only necessary data
- **Consent Management:** Explicit consent for data processing
- **Right to Deletion:** Account and data deletion capability

### 4.3 Availability Requirements

#### 4.3.1 Uptime

- **Target:** 99.9% uptime (8.76 hours downtime/month)
- **Maintenance Window:** 2 hours weekly (Sundays 2-4 AM GMT)
- **Disaster Recovery:** 4-hour RTO (Recovery Time Objective)

#### 4.3.2 Error Handling

- **Graceful Degradation:** App remains functional during partial outages
- **Offline Mode:** Basic functionality available offline
- **Error Messages:** User-friendly error messages with recovery options
- **Crash Reporting:** Automatic crash reporting and analytics

### 4.4 Usability Requirements

#### 4.4.1 User Experience

- **Learning Curve:** New users complete first transaction in < 5 minutes
- **Task Success Rate:** > 95% for core tasks
- **User Satisfaction:** > 4.5/5 rating in app stores
- **Accessibility:** WCAG 2.1 AA compliance

#### 4.4.2 Design Standards

- **Consistency:** Material Design 3 compliance
- **Responsive Design:** Optimized for various screen sizes
- **Multi-language:** Support for 5 languages
- **Theme Support:** Light, dark, and system themes

## 4.5 Compatibility Requirements

### 4.5.1 Platform Support

- **Android:** 5.0 (API 21) and above
- **iOS:** 12.0 and above
- **Screen Sizes:** 4" to 7" devices
- **Orientations:** Portrait primary, landscape secondary

### 4.5.2 Network Conditions

- **Connectivity:** 2G, 3G, 4G, 5G, WiFi
  - **Offline Support:** Queue transactions when offline
  - **Data Usage:** < 1MB per typical transaction flow
- 

## 5. User Interface Requirements

### 5.1 Design System

#### 5.1.1 Color Palette

- **Primary:** #00D47E (Vibrant mint green)
- **Primary Dark:** #00B870
- **Secondary:** #2563EB (Blue accent)
- **Background:** #F8FAFC (Light), #0B1F1A (Dark)
- **Text:** #0F172A (Light), #F8FAFC (Dark)
- **Success:** #00D47E
- **Error:** #EF4444
- **Warning:** #FACC15

#### 5.1.2 Typography

- **Font Family:** Inter (Google Fonts)
- **Display Sizes:** 40px, 32px, 28px
- **Headline Sizes:** 24px, 20px, 18px
- **Body Sizes:** 16px, 14px, 12px
- **Label Sizes:** 14px, 12px, 11px

#### 5.1.3 Spacing System

- **Base Unit:** 8px grid

- **Scale:** 4px, 8px, 16px, 24px, 32px, 48px, 64px

### 5.1.4 Component Standards

- **Border Radius:** 16px (default), 24px (cards), 8px (small elements)
- **Elevation:** Flat design with subtle shadows
- **Buttons:** 56px minimum height, 16px border radius
- **Input Fields:** 16px border radius, filled style

## 5.2 Screen Requirements

### 5.2.1 Navigation Structure

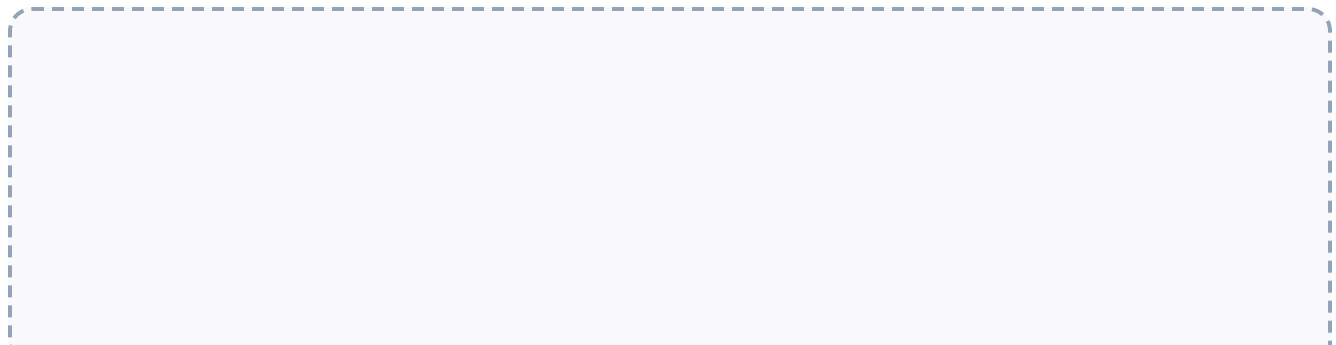
```
/login - Authentication  
/otp - OTP Verification  
/dashboard - Home Screen  
/airtime - Airtime Purchase  
/data - Data Purchase  
/transactions - Transaction History  
/statistics - Spending Analytics  
/settings - App Settings  
/profile - User Profile
```

### 5.2.2 Bottom Navigation

- 5 tabs: Home, Airtime, Data, History, More
- Fixed position, always visible
- Icons with labels
- Active state: Primary green color

## 5.3 UI Placeholders (Insert Screenshots/Mockups)

The sections below reserve space for interface screenshots in the PDF. Replace each placeholder with an exported screenshot/mockup when available.



## UI Placeholder: Login Screen

## UI Placeholder: Registration Screen

## UI Placeholder: OTP Verification Screen

**UI Placeholder: Dashboard (Balance + Quick Actions + Recent Transactions)**

**UI Placeholder: Airtime/Data Menu Screen**

**UI Placeholder: Select Recipient (Contacts + Recent)**

**UI Placeholder: Enter Amount (Presets + Conversion + Fees)**

**UI Placeholder: Confirm Transaction (Summary + Payment Method)**

**UI Placeholder: ExpressPay Payment (WebView/Deep Link Callback)**

**UI Placeholder: Transaction Receipt (Share/Save)**

**UI Placeholder: Transaction History (Filters + List)**



**UI Placeholder: Transaction Detail Screen**

**UI Placeholder: Settings / Profile**

## 6. API Requirements

### 6.1 Base Configuration

- **Base URL:** `https://api.advansistechnologies.com`
- **API Version:** `/api/v1`
- **Authentication:** Bearer token (JWT)
- **Content Type:** `application/json`
- **Timeout:** 15 seconds connection, 30 seconds read

### 6.2 Required Endpoints

#### 6.2.1 Authentication

- POST `/api/v1/users/register` - User registration
- POST `/api/v1/users/login` - User login
- POST `/api/v1/auth/refresh` - Token refresh
- POST `/api/v1/users/change-password` - Password change
- GET `/api/v1/users/profile` - Get profile
- PUT `/api/v1/users/profile` - Update profile

#### 6.2.2 Wallet

- GET `/api/v1/wallet/balance` - Get wallet balance

#### 6.2.3 Airtime

- POST `/api/v1/airtime/reloadly` - International airtime
- POST `/api/v1/airtime/prymo` - Ghana local airtime

#### 6.2.4 Data

- POST `/api/v1/data/reloadly` - International data
- POST `/api/v1/data/prymo` - Ghana local data

#### 6.2.5 Payment

- POST `/api/v1/payment/expresspay/initiate` - Initiate payment
- POST `/api/v1/payment/expresspay/verify` - Verify payment
- GET `/api/v1/payment/expresspay/methods` - Get payment methods

#### 6.2.6 Transactions

- GET /api/v1/transactions - List transactions
- GET /api/v1/transactions/{id} - Transaction details

## 6.3 Error Handling

- **Standard HTTP Status Codes:** 200, 201, 400, 401, 403, 404, 500
- **Error Response Format:**

```
{
  "error": {
    "code": "ERROR_CODE",
    "message": "Human readable message",
    "details": "Additional context"
  }
}
```

---

## 7. Testing Requirements

### 7.1 Testing Strategy

#### 7.1.1 Unit Testing

- **Coverage:** Minimum 80% code coverage
- **Tools:** Flutter test framework
- **Scope:** Business logic, utilities, models

#### 7.1.2 Integration Testing

- **API Integration:** Test all API endpoints
- **Payment Flow:** End-to-end payment testing
- **Authentication Flow:** Complete auth journey testing

#### 7.1.3 UI Testing

- **Widget Tests:** Critical UI components
- **Golden Tests:** Visual regression testing
- **Accessibility Tests:** Screen reader compatibility

#### 7.1.4 Performance Testing

- **Load Testing:** 1000+ concurrent users
- **Stress Testing:** Peak load scenarios
- **Memory Testing:** Memory leak detection

## 7.2 Test Environments

- **Development:** Local development environment
  - **Staging:** Production-like environment
  - **Production:** Live environment with monitoring
- 

# 8. Deployment Requirements

## 8.1 App Store Requirements

### 8.1.1 Google Play Store

- **Target SDK:** 34 (Android 14)
- **Content Rating:** Everyone
- **Category:** Finance
- **Privacy Policy:** Required and accessible
- **Target Audience:** Ghana (primary), International (secondary)

### 8.1.2 Apple App Store

- **iOS Target:** 17.0
- **App Category:** Finance
- **Age Rating:** 4+
- **App Privacy:** Complete privacy nutrition label

## 8.2 Build Requirements

### 8.2.1 Android

- **Build Type:** APK for testing, AAB for production
- **Signing:** Release signing with proper keystore
- **ProGuard:** Code obfuscation enabled
- **Size:** < 50MB APK size

### 8.2.2 iOS

- **Build Type:** IPA for distribution

- **Code Signing:** Proper certificates and provisioning
  - **App Thinning:** Enabled for device-specific optimization
  - **Size:** < 100MB IPA size
- 

## 9. Security Considerations

### 9.1 Data Protection

- **Local Storage:** Sensitive data encrypted
- **API Keys:** Stored securely, not hardcoded
- **User Data:** Minimal data collection
- **Logs:** No sensitive information in logs

### 9.2 Network Security

- **Certificate Pinning:** Prevent MITM attacks
- **Request Signing:** Critical requests signed
- **Rate Limiting:** API abuse prevention
- **Input Validation:** All inputs validated

### 9.3 Application Security

- **Root/Jailbreak Detection:** Basic detection
  - **Screen Recording:** Prevent sensitive screens
  - **Screenshot Prevention:** Payment screens
  - **App Backup:** Disabled for sensitive data
- 

## 10. Assumptions and Constraints

### 10.1 Assumptions

- Users have stable internet connection for transactions
- Ghanaian mobile networks support API integration
- Payment gateway (ExpressPay) remains operational
- Users understand basic mobile app usage
- Device hardware meets minimum requirements

### 10.2 Constraints

- MVP limited to Ghana market only

- Single payment gateway (ExpressPay) initially
- No offline transaction processing
- Limited to airtime and data services (no bill payments)
- No peer-to-peer transfers in MVP

## 10.3 Dependencies

- External API availability (Reloadly, Prymo, ExpressPay)
  - Flutter framework stability and updates
  - App store approval processes
  - Network operator cooperation
  - Payment gateway reliability
- 

# 11. Success Criteria

## 11.1 Technical Metrics

- **App Launch Time:** < 3 seconds
- **API Response Time:** < 3 seconds (95th percentile)
- **Crash Rate:** < 0.5%
- **API Success Rate:** > 99.5%
- **Test Coverage:** > 80%

## 11.2 Business Metrics

- **User Registration:** 1000+ users in first month
- **Transaction Success:** > 95% completion rate
- **User Retention:** 60% monthly retention
- **App Store Rating:** > 4.0 stars
- **Support Tickets:** < 5% of transactions

## 11.3 User Experience Metrics

- **First Transaction:** < 5 minutes from install
  - **Task Success Rate:** > 95% for core flows
  - **User Satisfaction:** > 4.2/5 rating
  - **Support Response:** < 2 hours average
  - **Feature Adoption:** > 70% for core features
-

## 12. Risk Assessment

### 12.1 Technical Risks

- **API Downtime:** Mitigate with retry logic and error handling
- **Payment Failures:** Implement robust error recovery
- **Performance Issues:** Optimize code and use caching
- **Security Breaches:** Regular security audits and updates

### 12.2 Business Risks

- **Regulatory Changes:** Monitor regulatory environment
- **Competition:** Differentiate with better UX and features
- **Market Adoption:** Aggressive marketing and user education
- **Payment Gateway Issues:** Backup payment provider options

### 12.3 Operational Risks

- **Scaling Issues:** Design for scalability from start
  - **User Support:** Implement comprehensive support system
  - **Data Loss:** Regular backups and disaster recovery
  - **App Store Rejection:** Follow guidelines strictly
- 

## 13. Future Enhancements (Post-MVP)

### 13.1 Feature Roadmap

- **Bill Payments:** Utilities, TV, internet bills
- **Peer-to-Peer Transfers:** User-to-user money transfers
- **Virtual Cards:** Physical and virtual debit cards
- **Investment Services:** Savings and investment products
- **Business Services:** Merchant services and bulk purchases

### 13.2 Technical Enhancements

- **Push Notifications:** Transaction alerts and promotions
  - **Biometric Authentication:** Fingerprint and face recognition
  - **Voice Commands:** Voice-activated transactions
  - **AI Assistant:** Smart recommendations and support
  - **Blockchain Integration:** Enhanced security and transparency
-

# 14. Approval Sign-off

| Role                 | Name | Signature | Date |
|----------------------|------|-----------|------|
| Product Owner        |      |           |      |
| Technical Lead       |      |           |      |
| QA Lead              |      |           |      |
| Business Stakeholder |      |           |      |

## Document History

| Version | Date            | Author           | Changes         |
|---------|-----------------|------------------|-----------------|
| 1.0     | January 8, 2026 | Development Team | Initial MVP SRS |

**This document serves as the authoritative source of requirements for the LidaPay Mobile Application MVP. All development, testing, and deployment activities should reference this document to ensure alignment with project objectives and quality standards.**