Application Documentation

Flutter Mobile Application Documentation

Overview

The developed Flutter mobile application serves as a financial transaction platform, allowing users to securely register, authenticate, verify identities through integration with **Esignet**, deposit funds using multiple payment gateways (Visa Cards and Mobile Money), and send money to other registered users within the application ecosystem. The app consumes Django backend endpoints, incorporates React-based Esignet integration, and operates smoothly on Android emulators (e.g., Pixel 9 Pro XL API 35).

Application Flow

1. User Registration & Authentication

- **Registration**:
    - User initiates the app by creating an account.
    - User provides registration details through a registration form.
    - Upon successful registration, user is directed to the login page.
- **Login (Authentication)**:
    - User logs in using registered credentials (username/email and password).
    - Django backend authenticates credentials and allows access.

2. Home Page

- **Access**:

    - Accessible only after successful authentication.
    - Displays user's basic information and key functionalities.
- **Components Displayed**:
    - Username
    - Account balance
    - Transaction history
    - News/announcements section
    - **Two main buttons**:
        - **Send Money**
        - **Deposit Money**
- **Interaction Restriction**:
    - Initially, the user is restricted from interacting with any components.
    - Clicking on any component directs the user to the **verification page** for Esignet verification.

3. Identity Verification (Esignet Integration)

- **Verification Process**:
    - Upon clicking any interactive component for the first time, user is redirected to the verification page.
    - Verification involves integration with a **React application/plugin** implementing the Esignet verification button.
    - Upon successful verification, user receives full access to the Home page and all functionalities.
- **Verification Status Indication**:
    - Verified users are distinctly indicated with a **blue tick (✓)** icon beside their profiles to confirm verification status.

Functionalities

1. Depositing Money

- **Gateway Integration**:
    - Integrated payment gateway supporting:
        - **Visa Cards**
        - **Mobile Money**
- **Deposit Flow**:
    - User clicks **Deposit Money** button.
    - Selects preferred payment method (Visa or Mobile Money).
    - Enters deposit amount.
    - Completes transaction securely through the integrated gateway.
    - Upon success, deposited amount reflects in user's balance.
- **Deposit History**:
    - Users can view detailed deposit history.
    - Endpoint: `http://10.0.2.2:8000/api/v1/dpst/history/`

---

2. Sending Money (User-to-User Transaction)

- **Transaction Initiation**:
    - User selects **Send Money**.
    - The app displays a list of registered users within the application.
    - User selects recipient, retrieves email automatically.
    - User enters transaction amount and confirms the transfer.
- **Processing Transaction**:

    - Transaction request sent to Django backend (`transferUrl`).
    - Backend updates sender's and recipient's account balances accordingly.
- **Transaction History**:
    - Transaction details can be viewed anytime.
    - Endpoint: `http://10.0.2.2:8000/api/v1/trsf/history/`

API Endpoints (Django Integration)

Operation	Method	Endpoint URL	Purpose
Fetch Users	GET	`http://10.0.2.2:8000/api/v1/usr/`	Retrieve registered users list.
Transfer Money	POST	`http://10.0.2.2:8000/api/v1/trsf/`	Initiate money transfer.
Deposit History	GET	`http://10.0.2.2:8000/api/v1/dpst/history/`	Retrieve deposit history.
Transaction History	GET	`http://10.0.2.2:8000/api/v1/trsf/history/`	Retrieve transaction history details.

Security & Compliance

1. Authentication & Authorization

- Secure Django-based authentication ensures robust user credential verification.
- Sensitive user data and transactions protected through secured APIs.

2. Identity Verification

- Esignet integration ensures verified identities.

- React app/plugin implementation securely handles verification process.

3. Payment Gateway Security

- Compliant with security standards for payment processing (PCI DSS compliance recommended).
- Ensures encrypted transaction details for Visa and Mobile Money.

Technical Integration

Flutter Frontend

- Designed in Flutter for cross-platform mobile compatibility.
- Interacts seamlessly with Django RESTful APIs.

Django Backend

- Provides secure REST APIs for registration, authentication, deposits, transactions, and history retrieval.
- Stores and manages user data securely.

Esignet React Integration

- Implemented as an external React app/plugin.
- Verification seamlessly integrated into Flutter via WebView or appropriate embedding mechanism.

Visual Indicators & User Experience

Indicator	Meaning
Green Tick (✓) next to username	User successfully verified via Esignet.

Interactive components blocked	User verification pending.

Emulator Testing

Environment Details

- Android Emulator: **Pixel 9 Pro XL API 35**
- Endpoint Base URL (Emulator loopback): `http://10.0.2.2:8000/`

Code Reference (Constants & Configuration)

```
class ApiConstants {
  static const String baseUrl = 'http://10.0.2.2:8000/';
  static const String fetchUsersUrl = '${baseUrl}api/v1/usr/';
  static const String transferUrl = '${baseUrl}api/v1/trsf/';
  static const String depositHistoryUrl = '${baseUrl}api/v1/dpst/history/';
  static const String trsfHistoryUrl = '${baseUrl}api/v1/trsf/history/';
}
```

Future Recommendations

- Regular security audits for payment gateway integrations.
- UI/UX enhancements based on user feedback.
- Expanding supported payment methods and integrations.
- Robust error handling and comprehensive testing across multiple devices.

Documentation Compiled by:

- Developer: Vincent M. Banda Jr.
- Contact Email: vincentbanda010@gmail.com

Date: April 2025