App Architecture – CashConnect

1. Overview

CashConnect is a **desktop-based transaction management application** developed in **simple Python**, designed with a clear separation between **presentation (GUI)**, **application logic**, and **data management**. The architecture follows a **modular and layered approach** to ensure maintainability, scalability, and ease of deployment.

The application is optimized for **PC/Laptop full-screen dashboards**, supports **secure PIN-based authentication**, and opens each major activity in a **new full-sized window/tab** to maintain a professional user experience.

---

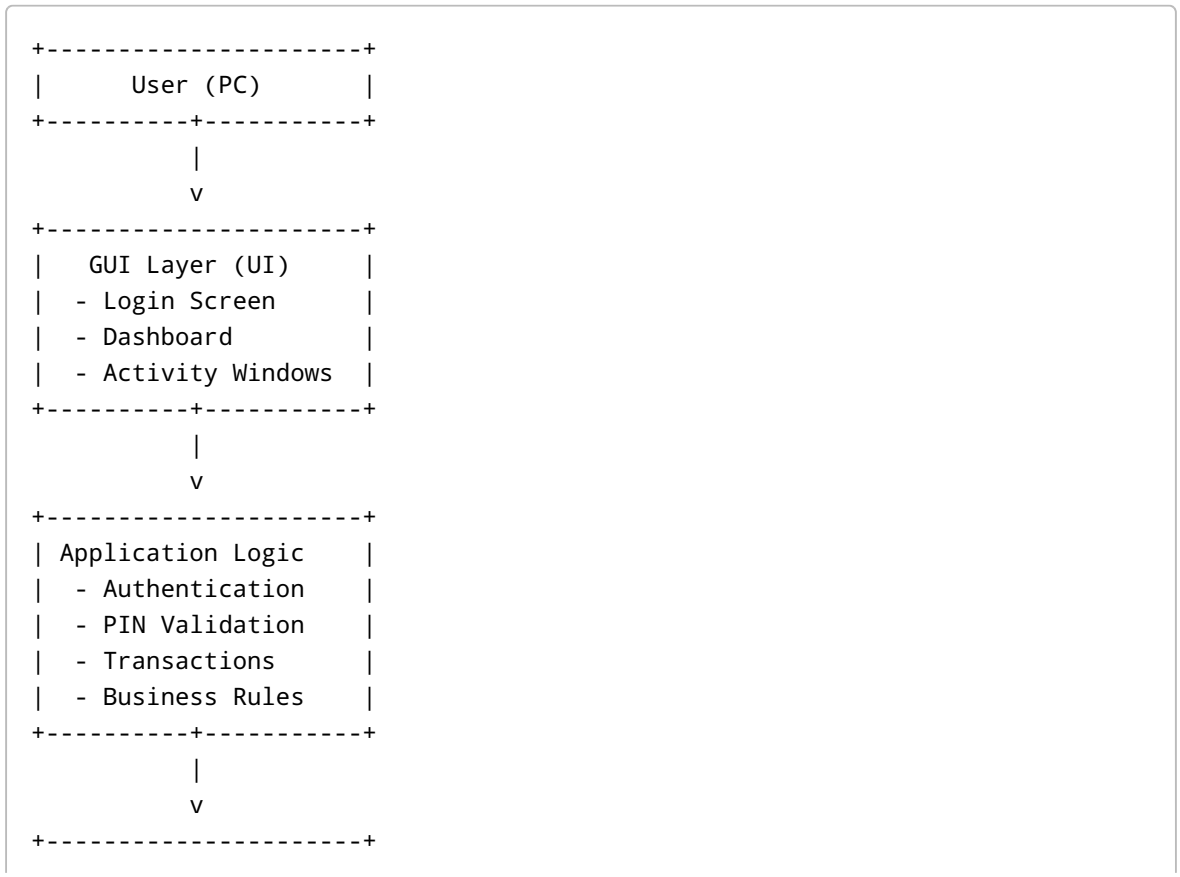
2. Architectural Style

Architecture Pattern: - Modular Monolithic Architecture - Layered Design (UI → Logic → Data)

Why this architecture? - Easy for beginners and students to understand - Clean separation of concerns - Ideal for academic and portfolio projects - Simple to extend (future APIs, database, or cloud support)

---

3. High-Level System Components



```
|   Data Layer         |
|  - Local Storage     |
|  - User Data         |
|  - Transaction Logs  |
+----------------------+
```

---

4. Layer-wise Breakdown

4.1 Presentation Layer (GUI)

Responsibilities: - Handles user interaction - Displays data in full-screen windows - Opens each activity in a new tab/window

Key Features: - Responsive full-screen dashboard (PC/Laptop aspect ratio) - Separate windows for: - Send Money - Receive Money - Transaction History - Profile & Settings

Technologies: - Tkinter / CustomTkinter (or equivalent simple Python GUI library)

4.2 Authentication & Security Layer

Responsibilities: - User login and validation - Secure PIN storage and verification

Security Flow: 1. User enters username 2. User enters PIN (stored securely) 3. PIN is validated before dashboard access

Security Measures: - PIN stored in encrypted/hashed format - No plain-text PIN storage

---

4.3 Application Logic Layer

Responsibilities: - Business rules execution - Transaction validation - Navigation control between windows

Core Modules: - User Management - Transaction Processing - Balance Calculation - Activity Routing (new window handling)

4.4 Data Layer

Responsibilities: - Persistent storage of data - Read/write user and transaction records

Data Storage Options: - Local JSON files (initial version) - CSV files for transaction history

Stored Data Includes: - Username - Encrypted PIN - Account balance - Transaction logs (date, amount, type)

---

5. Navigation & Window Management

- Dashboard launches in **full-screen mode**
- Each activity opens in a **new full-sized window**
- Main dashboard remains accessible for navigation
- Clean exit and session handling

---

6. Scalability & Future Enhancements

This architecture supports easy upgrades such as: - Database integration (SQLite / PostgreSQL) - REST API support - Mobile or Web frontend - Role-based authentication - Cloud deployment

---

7. Deployment Architecture (GitHub Ready)

```
CashConnect/
│
├── src/
│   ├── gui/
│   ├── auth/
│   ├── logic/
│   ├── data/
│   └── main.py
│
├── assets/
├── docs/
│   └── architecture.md
├── requirements.txt
├── README.md
└── LICENSE
```

---

8. Key Architectural Advantages

- Clean separation of responsibilities
- Beginner-friendly Python structure
- Professional UI workflow
- GitHub-ready organization
- Easy academic evaluation and demo

Document Owner: Project Manager / Technical Writer **Project:** CashConnect – Transaction Application