**🔄 1. Transaction Management Enhancements**

**🔧 Editable Transactions**

* **Problem**: Users cannot edit transaction details like amount, category, or account after creation.
* **Solution**:
  + Add an updateTransaction() method.
  + Internally handle:
    - Reversing the effect of the old transaction on the old account.
    - Applying the effect of the new transaction on the updated account.
* **Example Pseudocode**:
* oldAcc.balance += oldTxn.amount; // Undo old
* newAcc.balance -= newTxn.amount; // Apply new
* **Impact**: Maintains data integrity and consistent account balances during edits.

**🛠️ 2. Admin Functionality**

**👤 Admin Access & User Management**

* **Enhancement**:
  + Add methods:
  + List<User> getAllUsers();
  + void deleteUser(int userId);
  + Implement role-based access control:
  + enum Role { USER, ADMIN }
* **Impact**:
  + Empowers admins to manage users and maintain system integrity.
  + Essential for scalable, multi-user applications.

**🧾 3. User-Friendly Display of Transactions**

**📋 Display Category & Account Names (Not Just IDs)**

* **Enhancement**:
  + During transaction fetch:
  + String getCategoryNameById(String id);
  + String getAccountNameById(String id);
  + Update frontend/report:
  + [ ₹400 | FOOD | Wallet Account | 17-07-2025 ]
* **Impact**: Makes transaction views readable and intuitive for users.

**🔄 4. Recurring Transaction Support**

**🔁 Automate Repetitive Entries**

* **Enhancement**:
  + Update Transaction class:
  + boolean isRecurring;
  + String recurrencePeriod; // DAILY, MONTHLY, etc.
  + LocalDate nextOccurrence;
  + Controller method:
  + void processRecurringTransactions(LocalDate today);
* **Logic**:
  + On system trigger or start:
  + if (txn.isRecurring && txn.nextOccurrence <= today)
  + createCopyOfTransaction();
  + update nextOccurrence;
* **Impact**: Automatically tracks EMIs, salaries, subscriptions—reduces manual work.

**✅ 5. Data Validation and Input Sanitization**

**🔐 Ensure Valid & Safe Entries**

* **What to Validate**:
  + Negative values
  + Invalid dates
  + Invalid category/account references
* **Method**:
* boolean validateTransaction(Transaction t);
* **Impact**: Prevents logical/data errors and enhances robustness.

**🔍 6. Search, Filter & Sort Options**

**📊 Enhanced Transaction Viewing**

* **Features to Add**:
  + Filter by: date range, amount range, category, account, type
  + Sort by: newest, oldest, highest, lowest
* **Method**:
* List<Transaction> filterTransactions(FilterOptions opts);
* **Impact**: Enables customized views, aiding budgeting and analysis.

**📊 7. Dashboard Overview**

**📈 Summary at a Glance**

* **Show on Dashboard**:
  + Total net balance
  + Income vs Expense chart
  + Budget utilization
  + Top 3 spending categories
* **Impact**: Gives users immediate insight into their financial health.