

Xpense – An expenditure management system

About Group

Project Work 1 - Group 68

Members:

1. Fenteng Michael - 11210750
2. Samuel Akpah - 11125009
3. Ryan Brown- 11357610
4. Wisdom Nana-Abena Ogbonna - 11288300
5. Michel Kpodo - 11012578
6. Ransford Larbi - 10681368
7. Adams Emmanuel Paddy - 11264136

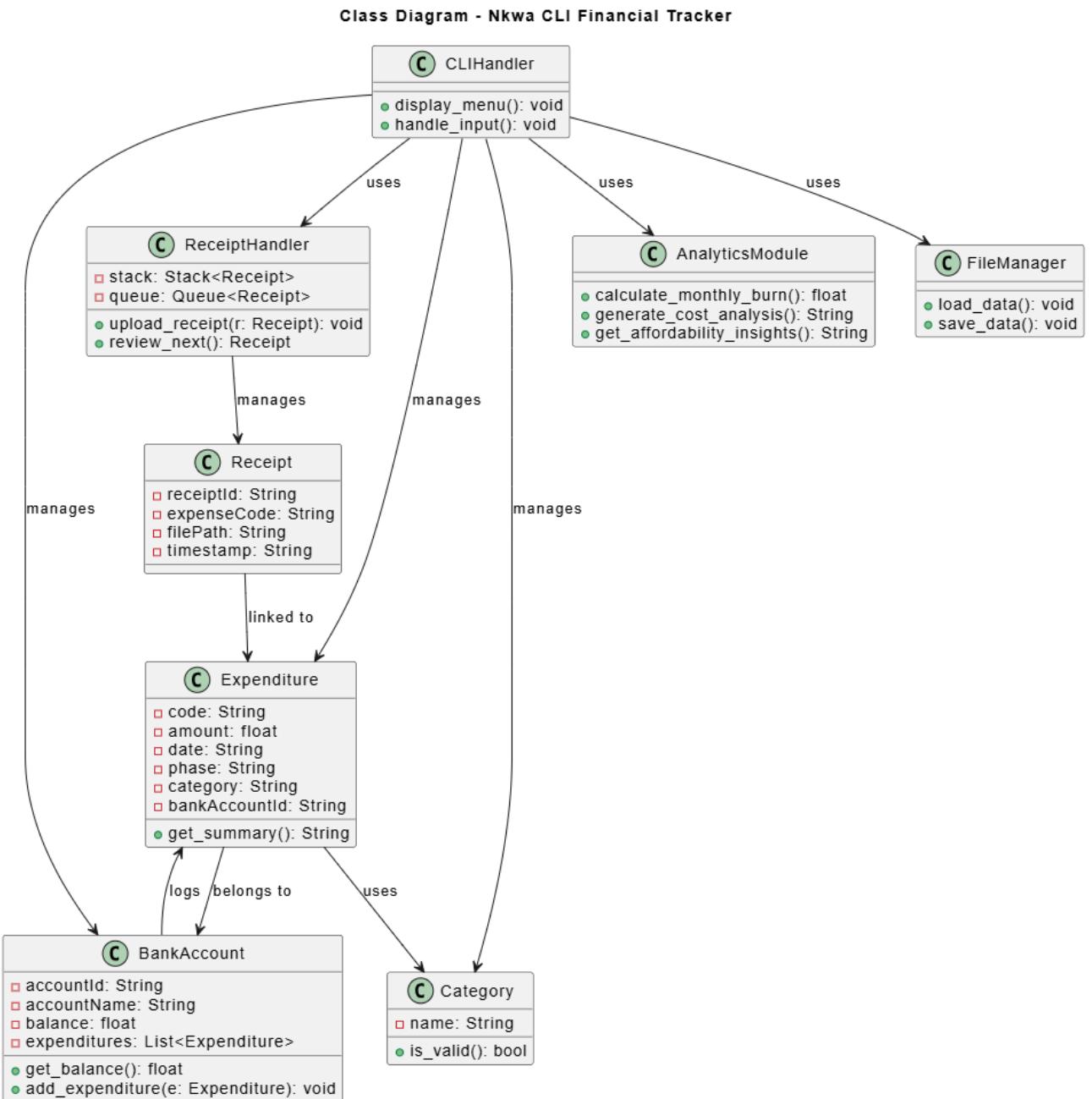
About Project

The goal of this project is to simulate a digital **expenditure tracking and financial management system** that mimics an accountant's workflow. It allows for:

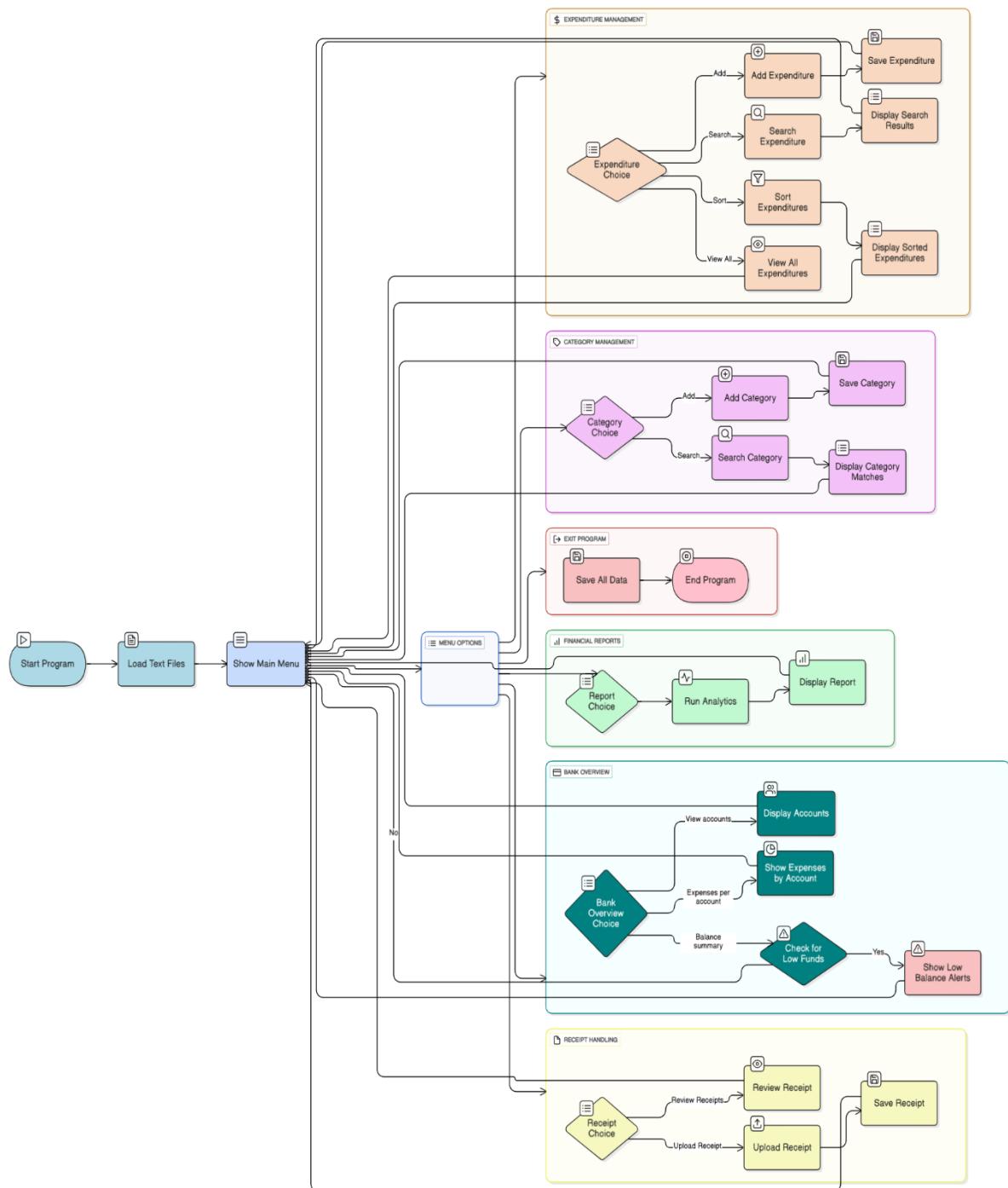
- Recording and categorizing expenditures
- Managing bank accounts and balances
- Generating analytics reports
- Raising alerts for critical financial events
- All while ensuring it works **offline-first** with no cloud or network dependencies.

This system helps users (usually accountants) manage their finances efficiently without the need for internet.

UML Diagram



FlowChart explaining how our system works



Classes and the various data structures used

1. CategoryManager

Purpose: Manages expenditure categories (add, search, update, delete, validate) and links expenditures to their respective categories.

Data Structures Used:

- SimpleSet: Custom set for storing unique categories
- SimpleMap: Maps category names to their expenditure list

These data structures were used because:

Workflow Simulated	Explanation
Avoiding duplicate categories	SimpleSet ensures uniqueness of categories , like an accountant organizing ledgers by distinct types.
Linking expenditures by type	SimpleMap provides a direct, fast lookup for category-to-expenditure association.
Efficient validation & updates	Enables real-time category validation without needing database access

The **worst-case (O)** and **best-case (Ω)** time complexities based on the **data structures used**

Operation	Big-O (Worst)	Omega (Best)	Notes
Add category	$O(1)$	$\Omega(1)$	Uses SimpleMap (likely a custom map with direct access or linear search)
Get category by name	$O(n)$	$\Omega(1)$	Depending on map implementation
List all categories	$O(n)$	$\Omega(n)$	Linear traversal

2. Category

Purpose: Represents a single expenditure category with metadata.

Data Fields: String id, String name, String description, String color

These data fields were used because:

Workflow Simulated	Explanation
Structured record-keeping	An accountant often tags and describes categories to keep financial reporting meaningful
Data validation before entry	isValid() ensures the object is complete, showing integrity

The **worst-case (O)** and **best-case (Ω)** time complexities based on the **data structures used**:

Operation	Big-O	Omega	Notes
Getters/Setters	$O(1)$	$\Omega(1)$	Simple field access
Object creation	$O(1)$	$\Omega(1)$	Just field assignment

3. BankLedger

Purpose: Central ledger for managing bank accounts, handling debit/credit transactions, and recording history.

Data Structures Used:

- HashMap: Stores all accounts by account ID
- List: Global transaction history
- HashMap: Per-account transaction tracking

These data structures were used because:

Workflow Simulated	Explanation
Ledger-style bookkeeping	HashMap allows fast access to any account, like an accountant accessing ledgers by ID
Efficient transaction tracking	Lists allow sequential transaction history, essential for audits and reporting
Modular logging	Every debit/credit is logged automatically — essential for reconciliation

The **worst-case (O)** and **best-case (Ω)** time complexities based on the **data structures used**:

Operation	Big-O (Worst)	Omega (Best)	Notes
Add new account	$O(1)$	$\Omega(1)$	HashMap insert
Debit/Credit account	$O(1)$	$\Omega(1)$	HashMap lookup and balance update
Get balance	$O(1)$	$\Omega(1)$	Fast lookup
Log expenditure	$O(1)$	$\Omega(1)$	HashMap + ArrayList append
Get transaction history	$O(k)$	$\Omega(k)$	k = number of transactions for that account
Get all transaction history	$O(n)$	$\Omega(n)$	n = total transactions
Get total balance (all accounts)	$O(m)$	$\Omega(m)$	m = number of accounts (sums all balances)

4. BankAccount

Purpose: Represents an individual bank account with balance, metadata, and expenditures.

Data Structures Used:

- BigDecimal balance: For precise financial arithmetic
- ArrayList: Tracks expenditures linked to this account

Workflow Simulated	Explanation
High-precision financial tracking	BigDecimal avoids floating-point errors — crucial in financial systems
Line-item tracking per account	Expenditures per account mirror ledger entries in accounting

The **worst-case (O)** and **best-case (Ω)** time complexities based on the **data structures used**:

Operation	Big-O	Omega	Notes
Debit/Credit	$O(1)$	$\Omega(1)$	Just balance arithmetic
Add expenditure	$O(1)$	$\Omega(1)$	ArrayList append
Get expenditures list	$O(1)$	$\Omega(1)$	Reference return
Get balance	$O(1)$	$\Omega(1)$	Simple read

5. AnalyticsModule

Purpose: Performs financial analysis — such as category breakdowns, burn rate, spending trends, and affordability.

Data Structures Used:

- TreeMap Chronologically sorted monthly spending
- PriorityQueue: Tracks top spending categories
- List: Used for sorting category spending totals

Workflow Simulated	Explanation
Trend and time-series analysis	TreeMap ensures months are sorted for chronological reporting
Budget prioritization	PriorityQueue allows ranking of highest-spending categories
Reporting on expense distribution	Accountant-style reporting by category is critical for financial review

The **worst-case (O)** and **best-case (Ω)** time complexities based on the **data structures used**:

Operation	Big-O	Omega	Notes
Calculate monthly burn	$O(n)$	$\Omega(n)$	Sums all expenditures
Generate cost/category analysis	$O(n \log n)$	$\Omega(n)$	TreeMap sort
Get affordability insights	$O(1)$	$\Omega(1)$	Simple comparison
Get spending trends	$O(n \log n)$	$\Omega(n)$	TreeMap insertion
Get top categories	$O(n \log n)$	$\Omega(n)$	Sort top-N

6. AlertSystem

Purpose: Manages alerts (e.g., low balance, overspending) using a custom MinHeap — no Java built-in collections used for heap.

Data Structures Used:

- MinHeap: Binary heap managing alerts by priority
- Alert : Stores alert message and priority

Workflow Simulated	Explanation
Alert prioritization	Urgent alerts surface first — like an accountant reacting to critical events
Budget breach detection	Detects if spending exceeds thresholds per category
Smart reminders	Alerts prompt user action (e.g., deposit funds, cut spending)

The **worst-case (O)** and **best-case (Ω)** time complexities based on the **data structures used**:

Operation	Big-O	Omega	Notes
Add alert	$O(\log n)$	$\Omega(1)$	Binary heap insert
Get next alert	$O(\log n)$	$\Omega(1)$	Binary heap removal
Check low funds / overspend	$O(\log n)$	$\Omega(1)$	Triggers alert
Display all alerts	$O(n \log n)$	$\Omega(n)$	Drains the entire heap

7. FileManager

Purpose: The FileManager is responsible for **all file input/output (I/O) operations** in the expenditure management system.

Data Structures & Utilities Used:

ArrayList: Stores loaded objects (e.g., Expenditure, BankAccount, Receipt) before returning to the system

BufferedReader / BufferedWriter: Efficient reading and writing of text files

Workflow Simulation	Explanation
Persistent record-keeping	Reads and writes structured plain-text files for expenditures, categories, accounts, and receipts

Workflow Simulation	Explanation
Data integrity and recovery	Backup and restore system ensures continuity in case of data loss
Journal-like audit trail	Writes structured entries (e.g., `code
Separation of modules with clean I/O	Each file type is read/written independently, reflecting how accountants work with separate books/files

The **worst-case (O)** and **best-case (Ω)** time complexities based on the **data structures used:**

Operation	Big-O	Omega	Notes
Load/save expenditures	$O(n)$	$\Omega(n)$	Reads/writes n lines
Load/save categories	$O(c)$	$\Omega(c)$	c = number of categories
Load/save bank accounts	$O(m)$	$\Omega(m)$	m = number of accounts
Load/save receipts	$O(r)$	$\Omega(r)$	r = number of receipts
Backup/Restore files	$O(f * s)$	$\Omega(f)$	f = number of files, s = size per file (linear copy)

8. XpenseSystem

Purpose of the Class: The XpenseSystem class serves as the core **orchestrator** of the entire offline-first expenditure management system. It integrates all sub-modules; including file handling, alerts, expenditures, accounts, categories, analytics, and search into a unified workflow. It handles initial data loading, expenditure validation, and data persistence, mimicking the responsibilities of an accountant managing multiple ledgers, budgets, receipts, and alerts.

A step-by-step navigation example of adding an expenditure to the system

Welcome to Xpense

```
=====
Welcome to Xpense - Project Financial Tracker
=====

== MAIN MENU ==
1. Add Expenditure
2. List Expenditures
3. View Expenditure Details
4. Add Bank Account
5. List Bank Accounts
6. Add Category
7. List Categories
8. View Alerts
9. Search & Sort
10. Generate Reports
11. Bank Overview
12. Receipt Management
13. Help & About
0. Exit
```

Choose option 1 to add expenditure

```
=====
== MAIN MENU ==
1. Add Expenditure
2. List Expenditures
3. View Expenditure Details
4. Add Bank Account
5. List Bank Accounts
6. Add Category
7. List Categories
8. View Alerts
9. Search & Sort
10. Generate Reports
11. Bank Overview
12. Receipt Management
13. Help & About
0. Exit

Select an option (0-13): 1
```

Add description, amount, category name, and ID

```
Select an option (0-13): 1

== ADD NEW EXPENDITURE ==
? Tip: Enter 'cancel' or '0' at any time to go back
Description (or 'cancel'/'0' to go back): Ghana Food
Amount (GHC): 49
Category name (or 'cancel'/'0' to go back): Food and Drinks
Phase (or 'cancel'/'0' to go back): Boys Boys
Bank Account ID (or 'cancel'/'0' to go back): UMB002
Expenditure added successfully with ID: EXP1003 and bank account debited.

Press Enter to continue...  
|
```

Expenditure added

```
Select an option (0-13): 2
==== ALL EXPENDITURES ====
ID | Description | Amount (Ghc) | Category | Date | Phase | Account
-----
EXP1001 | Gob3 for the... | Ghc 50.00 | Food and Drinks | 2025-07-16 | active | CAL001
EXP1002 | Ffds | Ghc 44.00 | dfs | 2025-07-16 | active | UMB002
EXP1003 | Ghana Food | Ghc 49.00 | Food and Drinks | 2025-07-16 | Boys Boys | UMB002
Press Enter to continue...
```

Attendance