**GROUP 74**

**Nkwa Real Estate Ltd - Expenditure Management System**

**Project Report**

1. **Project Overview**

Nkwa Real Estate Ltd required an offline-first, command-line-based Expenditure Management System built with Java and core data structures. This system simulates accountant workflows by tracking expenditures, managing categories and bank accounts, and performing financial analysis without relying on external libraries or databases.

1. **Functional Requirement Fulfilment**
2. **Expenditure Records**

* Expenditures are uniquely stored using a custom MyMap (which is similar to a HashMap).
* Each record includes code, amount, date, phase, category, account, and receipt.

1. **Category Management**

* Categories are stored in a custom MyCategorySet (set-based structure) to ensure uniqueness.
* Search and listing of categories are supported.

1. **Bank Account Ledger**

* Bank accounts are mapped using MyMap and include account ID, name, balance, and linked expenditure codes.
* Expenditures affect account balances.

1. **Search and Sort**

* Users can search by date range, category, cost, and bank account.
* Sorting is done by category (alphabetically) and date (chronologically) using basic comparison-based logic.

1. **Invoice/Receipt Handling**

* Each expenditure is linked to a receipt using a text note.
* A stack or queue is used to simulate a review process.

1. **Bank Tracker**

* Expenditures deduct from bank balances.
* A MinHeapBankAlert is used to flag accounts below a user-defined threshold.

1. **Financial Analysis**

* **Burn Rate Tracking**: Monthly totals computed using MyMap.
* **Profit Forecast**: Month-over-month expenditure trends.
* **Material Impact**: Total/average spending on key materials. Estimated affordability calculated based on material cost.

1. **Data Structure Justification**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Structure Used** | **Justification** |
| Expenditures | MyMap<String, Expenditure> | Fast lookup by code |
| Categories | MySet<String> | It prevents duplicates |
| Bank Accounts | MyMap<String, BankAccount> | Easy mapping from ID to data |
| Receipt Review | Stack / Queue | Simulates upload/review flow |
| Low Balance Alert | Min Heap | Efficient retrieval of smallest balances |
| Sorting/Search | Arrays + Manual Algorithms | Full control and offline compliance |

1. **Sorting & Searching Techniques**

* **Sort by Category**: Bubble sort on expenditure array by getCategory().
* **Sort by Date**: Same bubble sort logic using getDate().
* **Search Filters**: Iterative traversal with comparison.

1. **Complexity Analysis**

|  |  |  |
| --- | --- | --- |
| **Operation** | **Best Case (Ω)** | **Worst Case (O)** |
| Search by Code (Map) | Ω(1) | O(n) (custom implementation) |
| Add to Set | Ω(1) | O(n) (to check duplicates) |
| Sorting | Ω(n) | O(n^2) (bubble sort) |
| Min Heap Insert | Ω(log n) | O(log n) |
| Expenditure Save to File | Ω(1) | O(1) |