**Personal Budget Management System**

**1. Purpose of the Assignment**

The main objective of this assignment is to design and develop a **Personal Budget Management System** to help users effectively track and manage their personal finances. In today's world, controlling expenses and saving money is a crucial factor that enables individuals to achieve their long-term financial goals. This system will help users gain a better understanding of their financial situation, enabling them to make smarter financial decisions. Specifically, the goals of this assignment include the following:

* **Income and Expense Management:** The system allows users to enter their income and expenses into specific categories (e.g., food, entertainment, bills, savings). This way, users can easily track how much they have spent and how much is left in their monthly budget.
* **Financial Goal Setting:** Users can set financial goals (e.g., saving a specific amount of money in 6 months) and track their progress over time. The system will help users see clearly whether they are on track or need to adjust their budget.
* **Ease of Use and Access:** The system is designed with a user-friendly interface that is easy to use and can be accessed across different platforms. This will allow users to track and update their budget anytime and anywhere.

**2. How to Apply it in Real-Life Scenarios**

In real-life situations, the **Personal Budget Management System** can be used by individuals to track their finances on a monthly or yearly basis. It can be applied to help users:

* **Track Monthly Income, Expenses, and Savings:** Users can enter their income and expenses into the system to monitor their financial situation on a monthly basis. The system will calculate and display the remaining balance after each expense.
* **Set Goals to Reduce Debt or Save for Specific Purposes (e.g., Vacation, Retirement):** Users can set specific saving goals for large future expenses or debt repayments and track their progress toward achieving these goals.
* **Analyze Spending Habits and Identify Areas to Cut Costs:** The system can analyze spending patterns by category (e.g., food, entertainment, bills) to identify areas where users can save, helping them adjust their budget effectively.
* **Generate Reports or Graphs Showing Financial Status and Trends Over Time:** The system can generate detailed financial reports or graphs that show users the trends in their income, expenses, and savings over time.

**3. Tasks to Complete**

In this project, the tasks to be completed include:

1. **Design and Develop a User Dashboard for Entering and Viewing Budget Data:** The user interface should be designed in a way that allows easy data entry and budget viewing, enabling users to input income, expenses, and monitor their financial status.
2. **Implement a System to Track Income and Expenses with Categorization (e.g., Food, Entertainment, Bills):** The system should allow users to categorize their expenses and income into different groups for easier tracking and management of their finances.
3. **Implement the Ability to Set and Track Saving Goals:** Users should be able to create saving goals for specific purposes and track their progress toward achieving those goals, such as saving for a vacation or retirement.
4. **Generate Reports and Provide Visual Insights into Spending Patterns:** The system should create reports and visual representations (such as graphs or charts) to give users a clear overview of their spending habits over time.

**4. Future Development**

In the future, the system could include:

* **Integration with Online Banking APIs for Real-Time Data Entry:** The system could be connected to users’ bank accounts via APIs to automatically import income and expense data in real-time, making financial tracking more seamless and accurate.
* **Advanced Machine Learning Algorithms to Predict Future Spending Patterns Based on Historical Data:** By using machine learning, the system could analyze users’ past spending behavior and predict future trends, helping them adjust their budgets proactively.
* **Mobile Application Version for Easy Access and Updates on the Go:** A mobile app version of the system would allow users to access and update their budgets anytime and anywhere, making it more convenient for users who are constantly on the move.
* **Budget Sharing Feature for Families or Households to Track Shared Finances:** The system could allow multiple users (e.g., family members) to share and manage a common budget, enabling better collaboration and tracking of joint expenses.
* **Support for Multiple Currencies for Global Users:** The system could support various currencies to accommodate users from different countries, allowing them to track their finances in their local currency.

**5. Implementation**

* **Technology Used:** The system is implemented using **JavaFX** for the user interface. JavaFX provides a rich set of UI controls, layout managers, and APIs that allow for the development of modern, dynamic, and interactive desktop applications. It is especially effective for building applications that require real-time updates and visually engaging elements, such as a personal budget management system.
* **Algorithm:**
  + **Binary Search by Amount:** **Binary Search** is used to efficiently search for transactions by amount within the Binary Tree. When the user performs a search based on the amount, the system traverses the Binary Tree to find transactions with matching amounts. This search is much faster than a linear search because Binary Search works on a sorted structure, reducing the search time to O(log n).
  + **Sorting Algorithms:**
    - **Bubble Sort:** This algorithm is used to sort transactions by their amount in ascending and descending order. The algorithm iteratively compares and swaps adjacent items until the list is sorted.
    - **Selection Sort:** This algorithm is used to sort transactions by date, from the newest to the oldest or vice versa. It selects the minimum or maximum element from the unsorted portion and places it at the correct position in the sorted part.
* **Recommended Data Structure:**
  + **Binary Tree:** A **Binary Tree** is used to store the transactions in a sorted order, based on their amount. Each node in the tree contains a transaction, and the left and right children represent transactions with lesser and greater amounts, respectively. This structure allows for fast searching, insertion, and deletion of transactions, with a time complexity of O(log n) for each operation, making it highly efficient for managing large datasets.

**6. Architecture and Folder Structure**

├── src/

│ ├── org/

│ │ ├── example/

│ │ │ ├── transactionmanager/

│ │ │ │ ├── Controller/

│ │ │ │ │ └── DisplayController.java

│ │ │ │ │ └── DashboardController.java

│ │ │ │ │ └── AddController.java

│ │ │ │ ├── Entity/

│ │ │ │ │ └── Transaction.java

│ │ │ │ ├── Model/

│ │ │ │ │ └── BinaryTree.java

│ │ │ │ │ └── Node.java

│ │ │ │ ├── Service/

│ │ │ │ │ └── TransactionService.java

│ │ │ │ ├── Main.java

│ │ │ ├── resources/

│ │ │ │ ├── fxml/

│ │ │ │ │ └── add\_transaction.fxml

│ │ │ │ │ └── display.fxml

│ │ │ │ │ └── dashboard.fxml

│ │ │ │ ├── styles

│ │ │ │ │ └── add.css

│ │ │ │ │ └── dashboard.css

│ │ │ │ │ └── display.css

**Mô Tả Cấu Trúc**

1. **Controller**:
   * Các lớp trong thư mục này chịu trách nhiệm xử lý các sự kiện từ người dùng, điều khiển sự tương tác giữa giao diện và dữ liệu.
2. **Entity**:
   * Các lớp trong thư mục này định nghĩa các đối tượng dữ liệu của ứng dụng.
3. **Model**:
   * Thư mục này chứa các lớp xử lý logic dữ liệu, giúp lưu trữ.
4. **Service**:
   * Các lớp trong thư mục này cung cấp các phương thức nghiệp vụ giúp kết nối giữa Controller và Model.
5. **resources**:
   * **fxml**: Chứa các tệp FXML mô tả cấu trúc giao diện người dùng của ứng dụng.
   * **styles**: Chứa các tệp CSS giúp thay đổi phong cách giao diện của ứng dụng.
6. **Main.java**:
   * Lớp chính của ứng dụng, nơi điểm khởi động của JavaFX được thiết lập.