**Project Option**

**Project Option:** Open-ended Project

**Project Description**

**Project Name:** Personal Finance Manager

**Description:** The Personal Finance Manager is designed to help individuals efficiently manage their finances by tracking income, expenses, and savings goals. A key feature of this application is its ability to automatically calculate and manage tithing, which is 10% of the income paid per paycheck or income. The program will allow users to input their income sources and categorize their expenses, providing an overview of their financial health. It will generate detailed reports, helping users understand their spending habits.

The project will include features to set and track savings goals, ensuring users can plan for future expenses or investments. The program will use principles of object-oriented programming to create a robust and scalable system. By implementing classes for income, expenses, tithing, and savings goals, the application will demonstrate abstraction, encapsulation, inheritance, and polymorphism. For example, a base Transaction class can be inherited by Income and Expense classes, with specific methods overridden to handle different transaction types. This project aims to provide users with a comprehensive tool to manage their finances while adhering to their tithing commitments.

UML Class Diagram

A black screen with white text

Description automatically generated

**Class Descriptions**

1. **Personal Finance Manager**
   * **Attributes:** transactions (List<Transaction>), savingsGoals (List<SavingsGoal>), users (List<User>), reports (List<Report>)
   * **Behaviors:**
     + AddTransaction(transaction: Transaction): void
     + RemoveTransaction(transaction: Transaction): void
     + CalculateTithing(income: decimal): decimal
     + AddSavingsGoal(goal: SavingsGoal): void
     + RemoveSavingsGoal(goal: SavingsGoal): void
     + AddUser(user: User): void
     + GenerateReport(): Report
2. **Transaction**
   * **Attributes:** date (DateTime), amount (decimal), category (string), description (string)
   * **Constructor:** Transaction(date: DateTime, amount: decimal, category: string, description: string)
3. **Income** (inherits from Transaction)
   * **Attributes:** incomeSource (string)
   * **Constructor:** Income(date: DateTime, amount: decimal, category: string, description: string, incomeSource: string)
4. **Expense** (inherits from Transaction)
   * **Attributes:** paymentMethod (string)
   * **Constructor:** Expense(date: DateTime, amount: decimal, category: string, description: string, paymentMethod: string)
5. **Tithing**
   * **Attributes:** tithingAmount (decimal)
   * **Constructor:** Tithing()
6. **SavingsGoal**
   * **Attributes:** goalName (string), goalAmount (decimal), progress (decimal)
   * **Constructor:** SavingsGoal(goalName: string, goalAmount: decimal)
7. **Report**
   * **Attributes:** reportType (string), content (string)
   * **Constructor:** Report(reportType: string, content: string)
   * **Behaviors:**
     + GenerateReport(transactions: List<Transaction>, savingsGoals: List<SavingsGoal>): string
8. **User**
   * **Attributes:** userID (int), userName (string), email (string)
   * **Constructor:** User(userID: int, userName: string, email: string)
   * **Behaviors:**
     + GetUserDetails(): string

**Interaction Diagram**

The interaction among the elements of the design is as follows:

* The Personal Finance Manager class interacts with Transaction, Income, Expense, Tithing, SavingsGoal, User, and Report classes.
* Users can add or remove transactions (income or expenses) to the Personal Finance Manager.
* The CalculateTithing method calculates the tithing amount based on the total income.
* Users can set and track savings goals using the AddSavingsGoal and RemoveSavingsGoal methods.
* The Transaction class serves as a base class for Income and Expense, demonstrating inheritance.
* The Tithing class manages tithing calculations separately, ensuring encapsulation and abstraction.
* The Report class generates detailed financial reports based on transactions and savings goals.
* The User class manages user information and details.