To further expand on the initial outline provided, we can add more classes and define their behaviors and attributes. Let's enhance the program with additional classes for better organization and functionality:

1. UserAccount:

* Attributes: Username, Password, Email, AccountBalance
* Methods: Login, Logout, ChangePassword, UpdateEmail, CheckBalance, Deposit, Withdraw

1. TransactionCategory:

* Attributes: Name, BudgetedAmount, ActualSpending
* Methods: SetBudgetedAmount, RecordSpending, CalculateRemainingBudget

1. TransactionManager:

* Attributes: List<Transaction>
* Methods: AddTransaction, RemoveTransaction, GetTransactionsInRange, CalculateTotalIncome, CalculateTotalExpenses

1. BudgetManager:

* Attributes: List<Budget>, TotalBudgetedAmount
* Methods: AddBudget, RemoveBudget, UpdateBudgetedAmount, CalculateTotalBudgetedAmount, CheckCategoryBudgetStatus

1. FinancialGoalManager:

* Attributes: List<FinancialGoal>
* Methods: AddGoal, RemoveGoal, UpdateGoal, TrackProgress, CheckCompletionStatus

1. ReportGenerator:

* Attributes: Account, TransactionManager, BudgetManager, FinancialGoalManager
* Methods: GenerateTransactionReport, GenerateBudgetReport, GenerateGoalProgressReport, GenerateSummaryReport

1. UserInterface:

* Methods: DisplayMainMenu, DisplayLoginScreen, DisplayDashboard, DisplayBudgetMenu, DisplayGoalMenu, DisplayReportsMenu
* Here's a class diagram representing the relationships between these classes:

+------------------+ +------------------+

| UserAccount | | TransactionCategory |

+------------------+ +------------------+

| - Username | | - Name |

| - Password | | - BudgetedAmount |

| - Email | | - ActualSpending |

| - AccountBalance | +------------------+

+------------------+ |

| |

| |

v v

+----------+ +------------------+

| Account | | TransactionManager |

+----------+ +------------------+

| - transactions | | - transactions |

+----------+ +------------------+

\ |

\ |

v v

+-------------+ +--------------+

| BudgetManager| | FinancialGoalManager |

+-------------+ +--------------+

| - budgets | - goals |

+-------------+ +--------------+

\ /

\ /

\ /

\ /

v v

+------------+

| ReportGenerator|

+------------+

| - account |

| - transactionManager |

| - budgetManager |

| - goalManager |

+------------+

As for the program's flow, it would involve the following steps:

The user logs in through the UserInterface.

After successful login, the UserInterface displays the main menu.

The user can navigate to different functionalities such as managing transactions, budgets, financial goals, or generating reports through the main menu.

The user interacts with various classes (e.g., TransactionManager, BudgetManager, FinancialGoalManager) to perform specific actions such as adding transactions, setting budgets, tracking goals, etc.

When necessary, the ReportGenerator class is invoked to generate reports based on the user's financial data.

The program continues to run until the user chooses to log out.

This flow ensures that the user can efficiently manage their personal finances by utilizing the provided functionalities and generating insightful reports.