

Semester Project: Money Management System

Iteration 2 – Class Diagram

System Description

The Money Management System is designed to represent a person's finances over the course of a month. At the center is the budget, which brings together all other parts of the system. The budget keeps track of totals, compares actual activity against limits, and produces a complete picture of financial health. To do this, it must organize and use several different kinds of information: day-to-day activity, grouped spending, money owed, and savings goals.

Transactions form the basic records of financial activity. Each transaction represents either income or an expense and includes details such as an amount and a description. Transactions are connected to the budget so they can contribute to totals, but they are also organized into categories. Categories make it possible to track spending by type, and each one has a spending limit. The system needs to compare the sum of a category's transactions with its limit in order to show whether the budget is being followed.

The system also needs to track long-term items that are not just single transactions. Debts represent obligations that must be reduced over time. Each debt stores a total balance and records payments that decrease it, making it possible to show the remaining amount owed. Sinking funds represent savings targets. Each fund has a goal amount and tracks contributions, allowing the system to measure progress toward that goal. Debts and funds can be managed separately, but the budget must also be able to combine them into summaries that show their overall effect on finances.

The budget acts as the link between these different elements. It gathers information from transactions, categories, debts, and funds, and uses that information to provide totals, remaining balances, summaries, and progress updates. Each element has its own purpose, but the system depends on the connections between them to work as a whole.

Your task is to represent this structure in a UML class diagram. The diagram should show the main components of the system, the information they contain, and the relationships that connect them.

Deliverables

You are responsible for producing a **Class Diagram** that represents the complete system.

- Each class should include attributes and methods that make sense for its role in managing financial data (pull from the project description above).
- Relationships between classes must be shown clearly, with correct useage of UML shapes.
- Multiplicity (e.g., one-to-many, many-to-many) must be included on relationships where it applies.
- The diagram should be complete, professional, and easy to read. A diagram submitted without a background or as an unrendered file type (drawio or html) will not be graded.

Submission

You do not need to reclone your directory to sumbit your work. The directory you cloned for the first iteration remains linked to the GitHub repo constantly. Submit your class diagram by adding it to the

directory you cloned from Git. Then follow the commands below to push it to the remote repository.

```
git add "class diagram file"  
git commit -m "Solution for Iteration 2"  
git push
```