Bank System Design Document

This document outlines the design of a bank system, focusing on key objects and their relationships.

System Overview

The bank system manages customer accounts, assets, and bank operations. It provides functionalities for creating and managing accounts, tracking assets, and performing transactions.
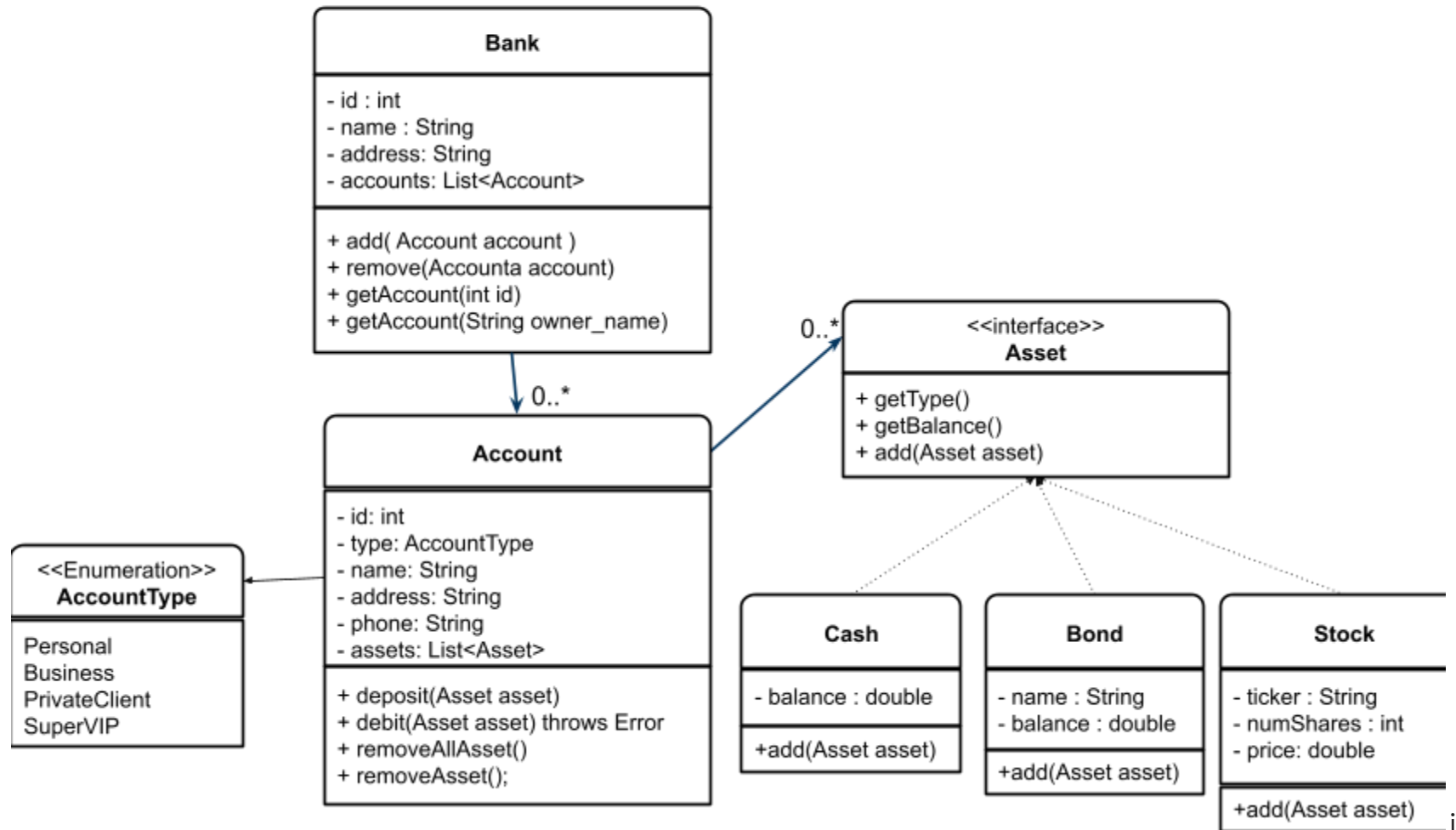
Object Model

The following objects are central to the system:

- **Bank:** Represents a financial institution.
- **Account:** Represents a customer's account within the bank.
- **Assets:** Represents the financial assets associated with an account.

UML Diagram

Aiwen's feet

Bank

- id : int
- name : String
- address: String
- accounts: List<Account>

+ add(Account account)
+ remove(Accounta account)
+ getAccount(int id)
+ getAccount(String owner_name)

0..*

0..*

<<interface>>
Asset

+ getType()
+ getBalance()
+ add(Asset asset)

Account

- id: int
- type: AccountType
- name: String
- address: String
- phone: String
- assets: List<Asset>

+ deposit(Asset asset)
+ debit(Asset asset) throws Error
+ removeAllAsset()
+ removeAsset();

<<Enumeration>>
AccountType

Personal
Business
PrivateClient
SuperVIP

Cash

- balance : double

+add(Asset asset)

Bond

- name : String
- balance : double

+add(Asset asset)

Stock

- ticker : String
- numShares : int
- price: double

+add(Asset asset)

like klein

Class Definitions

Bank

Attribute	Type	Description
id	int	Unique identifier for the bank.
name	String	Name of the bank.
location	String	Location of the bank.
totalAssets	double	Total assets held by the bank.
accounts	List<Account>	All the accounts in this bank

Account

Attribute	Type	Description
id	int	Unique identifier for the account.
type	Enumerator	Type of account (e.g., checking, savings).
assets	List<Asset>	Assets owned by this account

Assets

Attribute	Type	Description
assetType	Enumerate	Type of asset (e.g., cash, stocks, bonds, gold).

Attribute	Type	Description
value	Double	Current value of the asset.

Relationships
- A **Bank** has multiple **Accounts**.
- An **Account** has multiple **Assets**.

Future Considerations
- Implementation of transaction management.
- Integration with external systems.
- Security considerations.