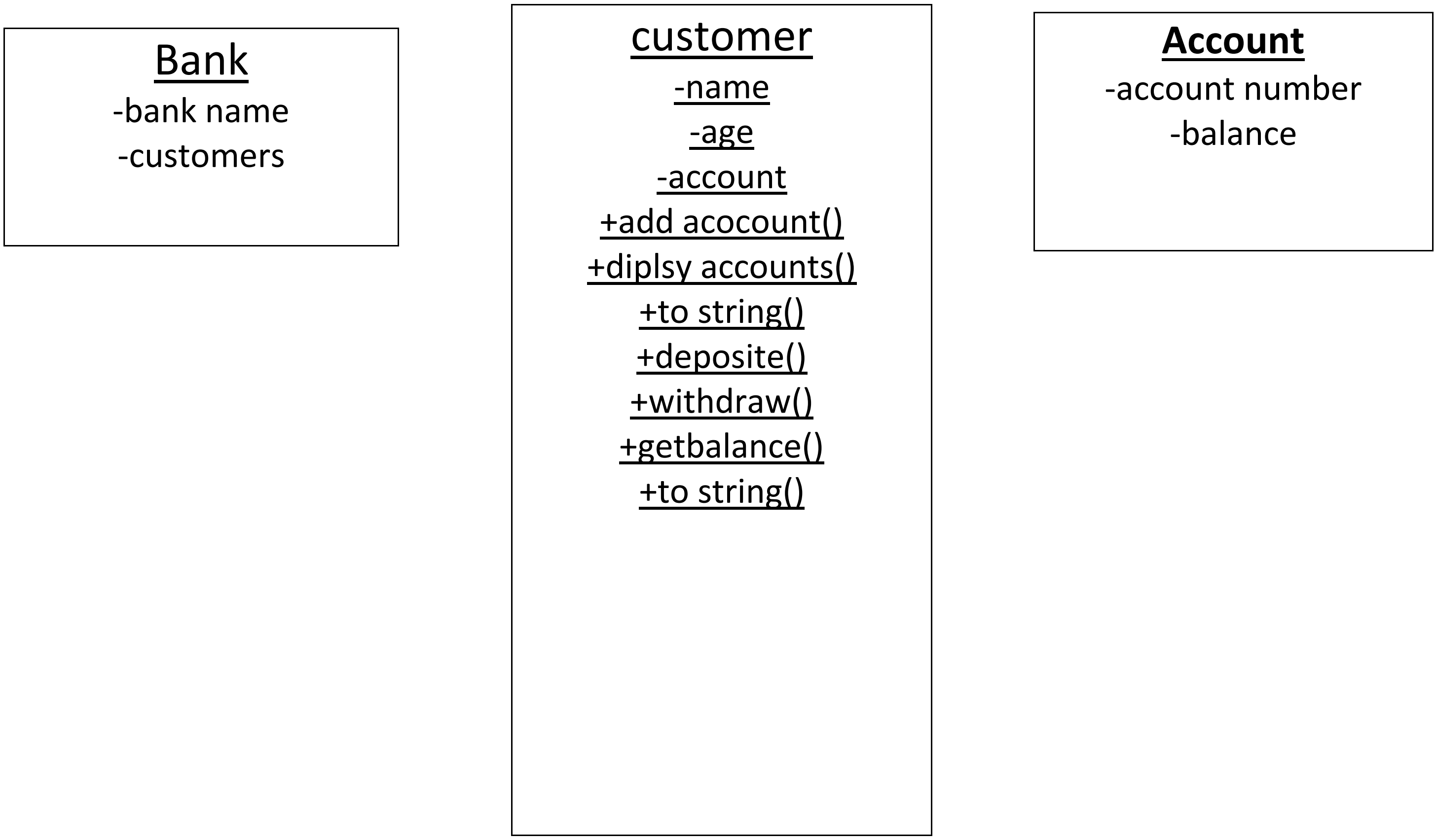

Bank

-bank name

-customers

customer

-name

-age

-account

+add acocount()

+diplsy accounts()

+to string()

+deposite()

+withdraw()

+getbalance()

+to string()

Account

-account number

-balance

The **Bank** class has a one-to-many relationship with the **Customer** class, as it can have multiple customers. The **Customer** class also has a one-to-many relationship with the **Account** class, as a customer can have multiple accounts.

The **Bank** class has attributes **bankName** and **customers**, representing the bank's name and the list of customers. It has methods **addCustomer()**, **displayCustomers()**, and **saveCustomersToFile()** for adding customers, displaying customer information, and saving customer details to a file, respectively.

The **Customer** class has attributes **name**, **age**, and **accounts**, representing the customer's name, age, and the list of accounts they own. It has methods **addAccount()**, **displayAccounts()**, and **toString()** for adding accounts, displaying account information, and generating a string representation of the customer, respectively.

The **Account** class has attributes **accountNumber** and **balance**, representing the account number and the current balance. It has methods **deposit()**, **withdraw()**, **getBalance()**, and **toString()** for depositing and withdrawing funds, retrieving the balance, and generating a string representation of the account, respectively.

The UML diagram provides an overview of the relationships and attributes among the classes in the Bank Management system.