**Banking System Documentation**

**Overview:**

This document describes a simple banking system implemented in C++. The system allows users to create accounts, deposit and withdraw money, manage loans, and view transaction records. The system is designed to be user-friendly and provides a menu-driven interface for interaction.

**Classes:**

1. **Transaction**

**Purpose:**

* Represents a financial transaction associated with an account.

**Attributes:**

* string type: The type of transaction (e.g., Deposit, Withdraw, Loan).
* int amount: The amount involved in the transaction.

**Constructor:**

* Transaction(string t, int a): Initializes a transaction with a type and amount.

1. **Account**

**Purpose:**

* Represents a bank account.

**Attributes:**

* string accountnumber: Unique identifier for the account.
* string name: Name of the account holder.
* double balance: Current balance of the account.
* double loanAmount: Amount of loan taken by the account holder.
* double loanInterestRate: Interest rate applicable to the loan.
* double loanBalance: Remaining balance of the loan.
* Account\* next: Pointer to the next account in the linked list.
* vector<Transaction> t: List of transactions associated with the account.

**Methods:**

* void addTransaction(string type, int amount): Adds a transaction to the account.
* void displayTransactions(): Displays all transactions for the account.

1. **Bank**

**Purpose:**

* Manages multiple accounts and provides banking functionalities.

**Attributes:**

* Account\* start: Pointer to the first account in the linked list.
* Account\* current: Pointer used for traversing the linked list of accounts.

**Methods:**

* void createaccount(string accnum, string accname, double initialdeposit): Creates a new account.
* void deposit(string accnum, int amount): Deposits money into an account.
* void withdraw(string accnum, int amount): Withdraws money from an account.
* void displayAccount(string accnum): Displays details of a specific account.
* void deleteAccount(string accnum): Deletes an account.
* void displayAllAccounts(): Displays all accounts in the bank.
* void searchAccount(string accnum): Searches for an account by its number.
* void addInterest(string accnum, double interestRate): Adds interest to an account.
* void requestLoan(string accnum, int amount, double interestRate): Requests a loan for an account.
* void repayLoan(string accnum, int amount): Repays a loan for an account.
* void displayLoanInfo(string accnum): Displays loan information for an account.

**Main Functionality:**

The main function provides a menu-driven interface for users to interact with the banking system. The following options are available:

* Create Account: Allows the user to create a new account by providing an account number, account holder's name, and an initial deposit.
* Deposit: Enables the user to deposit money into an existing account.
* Withdraw: Allows the user to withdraw money from an existing account.
* Display Account: Displays the details and transaction history of a specific account.
* Delete Account: Deletes an existing account.
* Display All Accounts: Lists all accounts in the bank.
* Search Account by Number: Searches for an account using the account number.
* Add Interest: Adds interest to the balance of a specific account.
* Request Loan: Allows the user to request a loan for a specific account.
* Repay Loan: Enables the user to repay a portion of the loan.
* Display Loan Info: Displays loan details for a specific account.
* Exit: Exits the banking system.

**Input Validation:**

The system includes basic input validation:

* Account holder names must contain only letters and spaces.
* Deposit and withdrawal amounts must be positive integers.
* Loan repayment amounts must also be positive integers.

**Conclusion:**

This banking system provides a foundational structure for managing bank accounts and transactions. It can be further enhanced with additional features such as user authentication, data persistence, and more complex financial operations.

**CODE:**





















