

Bank System using Spring

Created by Stephen O Driscoll for submission to App Dev. Frameworks

Database Design:

- I've used three tables in my database
- The first is a customer table which has columns for customerId, firstName and lastName. This table stores basic customer info.
- The second is an account table which has columns for accountId, balance and overdraft. This table stores basic account info.
- The last table is called customerAccount has columns for customerId and accountId. This table is the link between customer and account. An account can be owned by any number of customers.

Beans used:

- dataSource: Our sql database, created using "schema.sql" and "data.sql"
- jdbcTemplate: Used to perform queries on the database
- accountDao: Performs queries on the account and customerAccount
- customerDao: Performs queries on the customer table
- accountService: Implements features by calling accountDao and processing results
- customerService: Implements features by calling customerDao and processing results

Functions created and their limitations:

1. Register:
Creates a new customer and tells them their id. This id is needed whenever customers wish to use the banking system in future. Customers can have the same first and last names.
2. Create account:
Creates an account for a specified customer with a specified overdraft. If the customerId provided doesn't exist, then an account won't be created and "Customer not found" is displayed.
3. Add a person to an account:
Adds a specified customer to a specified account. Verifies that the customer exists, then verifies that the account exists and adds the association. Prints "Customer is already an owner of that account" if association already exists.
4. View accounts:
Displays account information for a specified customer. Displays customerId, firstName and lastName as well as accountId, balance and overdraft for every account. Displays "Customer not found" if customer doesn't exist.
5. Make a withdrawal:
Takes money away from a specified account. Can display "Account not found", "Insufficient funds" or "Give user €..." if the account meets the required criteria. Doesn't take in customerId. If an account has a balance of €2000 and an overdraft of €2000 and wishes to withdraw €4000 this action is allowed. A user can have a negative balance if it's below their overdraft. This function doesn't verify a customer's identity.

6. Make a deposit:
Adds a specified amount of money to a specified account. Again, customerId isn't required thus a customer isn't verified. Can display "Account not found. Returning money" or "Deposited €...".
7. Transfer money:
Combination of withdraw and deposit. Verifies both accounts exist, then verifies account1 has enough money and performs the transfer. Can display "Customer account not found. Returning money", "Account to transfer to not found. Returning money", "Insufficient funds" or "Transferred €...".
8. Close an account:
Removes an account from the database. Doesn't verify customer identity. Deletes all occurrences of this account from the account table and customerAccount table. Can display "Couldn't find an account with id ..." or "Account ... closed".
9. Exit:
Simply exits the program.
10. Total Deposited:
Sums up the balance of each account. Displays the total amount of money currently held by the bank across all accounts.
11. Num accounts over €10,000:
Counts the number of accounts that have more than €10,000. If an account has exactly €10,000 it isn't counted.