# Specification

This is a specification for a bank account module for a larger customer facing BankApp.

### *Overview*

Bank account information is held on the bank mainframe which serves as the system of record. This module, referred to in this spec a bank account class, serves as an in memory working copy of a bank account for the BankApp, which will be referred to here as the account object.

Once the BankApp session is saved the in memory updates to the account are written to the system of record.

### *Data Model*

The account object contains the following data:

1. **The Account Number.** This is a 8 digit number that is the unique identifier for the actual bank account.
2. **Status**: A one digit code that indicates the actual status of the bank account.
   1. 0 – Normal
   2. 1 – Closed – the account is not longer active. No operations are allowed
   3. 2 – Suspended – the account is currently suspended. Only queries may be done on the account object
3. **Balance**. The current balance of the account
4. **Available Balance:** The funds that are available for withdrawal.
5. **Transaction Limit:** The maximum amount that can be processed in a single withdrawal
6. **Session Limit:** The maximum total of all withdrawals that can be made during a single session of the BankApp

### *Constraints*

1. The balance of the account must never be negative. This app is intended to work only with accounts that are NOT overdrawn.
2. A withdrawal should be rejected if it would result in an overdraft
3. The available balance must always be less than or equal to the balance.
4. Amounts deposited or withdrawn must be greater than 0

5. Deposits and withdrawals can only be executed on accounts with a Normal status
6. Queries can only be executed on accounts that have either a normal or suspended status.
7. The transaction limit must always be less than or equal to the session limit. These limits can never be negative.

### *Interface*

The account object is defined by the following interface methods

1. *getData(account_number)* – Returns the data from the mainframe system of record as described above for the account corresponding to the data model described.
2. writeDataa(account_number, account_data) – writes the updated data to the account in the system of record.  Returns a true if the write was successful.
3. QueryBalance() - returns the balance of the current account object
4. queryAvailBalance() - returns the available balance for the current account object
5. deposit(amount) – increases the balance by the specified amount.
6. withdraw(amount) – decreases the balance and available balance by the specified amount.

Deposits and withdrawals return a true if the transactions executed sucessfully, false otherwise.

### *Operations*

1. The session and transaction limits cannot be modified during a sessions
2. Each deposit or withdraw attempted transaction must be recorded in a  transaction log The log entry contains a list of the account data, the operation attempted, the amount and whether it transaction succeeded.
3.  There may be only one account object active during a session for any given account number.
4. The account object is destroyed after the BankApp terminates.