.UI Menu

Start()

Calls if 1: AccountLookupWorkflow.Execute

Calls if 2: DepositWorkflow.Execute

.UI ConsoleIO

DisplayAccountDetails()

.UI Program

Calls Menu.Start()

.UI/Workflows/AccountLookupWorkflow

Void Execute():

Readline() account to look up

Creates AccountManager Manager (calls create() from AccountManagerFactory)

Calls Manager.LookupAccount(accountNumber) (AccountLookupResponse response)

Calls ConsoleIO.DisplayAccountDetails(response.Account)

.UI/Workflows/DepositWorkflow

Void Execute():

Reads Account number & deposit amt

Creates AccountManager(calls create() from AccountManagerFactory)

Calls accountManager.Deposit(accountNumber,amount) (AccountDepositResponse response)

Prints out account info including new balance

Static .BLL/AccountManagerFactory

Static Create()

Returns an AccountManager(after reading app.config) – currently FreeAccountTestRepository

.BLL/AccountManager (IAccountRepository via constructor)

LookupAccount(string acctnumber)

Calls LoadAccount(accountnumber) (=response.Account)

Returns AccountLookupResponse

Deposit(string acctnumber, decimal amount)

Calls LoadAccount (accountnumber)(=response.Account)

Calls DepositRulesFactory.Create(response.Account.Type)

Calls Deposit(response.Account, amount)

Calls SaveAccount

Returns AccountDepositResponse

.BLL/DepositRules/DepositRulesFactory

Static Create(AccountType):

Return new FreeAccountDepositRule(as an IDeposit)

.BLL/DepositRules/FreeAccountDepositRule : IDeposit

Deposit(Account,decimal):

Creates new AccountDepositResponse

Validates it meets free deposit constraints & sets balance and account properties

Returns AccountDepositResponse

.Data/FreeAccountTestRepository : IAccountRepository

Defines private Account w/data

LoadAccount(string acctnumber)

Returns Account

Void SaveAccount (Account account)

Sets \_account to account

Models/Responses/AccountLookupResponse : Response

Property Account account

From Response(success, message)

Models/Responses/Response

Properties bool success, string message

Models/Responses/AccountDepositResponse : Response

Properties:

Account Account (hold new balance)

Decimal Amount (amount that was deposited)

Decimal OldBalance

From Response(success, message)

IDeposit

AccountDepositResponse Deposit(Account, amount)

IAccountRepository

Account LoadAccount(acctnumber);

Void SaveAccount(Account account);

Models/Account

Properties:

String Name

String AccountNumber

Decimal Balance

AccountType type

Models/AccountType – enum

Free = 1

Basic

Premium