**User Requirement Specification**

Retail Bank Management

February 2016

Author: CTO APPhonics

Version: 1.1

Table of Contents

[1 Purpose of the document 3](#_Toc443425914)

[2 Business Overview 3](#_Toc443425915)

[3 Stakeholders 3](#_Toc443425916)

[4 Scope 4](#_Toc443425917)

[5 Requirement Statements 5](#_Toc443425918)

[5.1 For New account executive/New customer 5](#_Toc443425919)

[5.1.1 Login Screen: <common function for any employee role> 5](#_Toc443425920)

[5.1.2 Option of API call or database table for login and password validation 5](#_Toc443425921)

[5.1.3 Create Customer Screen: 5](#_Toc443425922)

[5.1.4 Update Customer Screen: 5](#_Toc443425923)

[5.1.5 Delete Customer Screen: 6](#_Toc443425924)

[5.1.6 Delete Account Screen: 7](#_Toc443425925)

[5.1.7 View Customer Status Screen: 7](#_Toc443425926)

[5.1.8 View Account Status Screen: 7](#_Toc443425927)

[5.1.9 Create database table to store entries for customer and account status 8](#_Toc443425928)

[5.2 For Cashier/Teller 8](#_Toc443425929)

[5.2.1 Get Accounts and Account Details Screen: 8](#_Toc443425930)

[5.2.2 Deposit Screen: 8](#_Toc443425931)

[5.2.3 Withdraw Screen: 9](#_Toc443425932)

[5.2.4 Transfer Screen: 9](#_Toc443425933)

[5.2.5 Get Statement Screen: 10](#_Toc443425934)

[6 Appendix 10](#_Toc443425935)

[6.1 Links to API documentation: 10](#_Toc443425936)

[6.2 Validations 11](#_Toc443425937)

# Purpose of the document

This document intends to specify and layout the functional requirements of a new software system required for ***Retail Bank employees as well as bank customers*** for the IT team to design and deliver web based User Interface.

# Business Overview

*Retail bank management* is one of the most widely used application management functions in banking and financial industry. This document specifies requirements, in a way, where in terms of both bank employees and bank customers, there will be distinct set of functions. These should be made available through well-defined User Interface.

**<ABC> Bank** is a retail bank which serves individual customers, to manage their savings and current accounts. Key requirements of the system is the ability to create customer, open customer account, update and delete customer, manage cashier transactions such as deposit, withdraw and transfer. Also, to be able to view customer and account information and get account transaction information.

Most of the above mentioned functions are made available as web APIs from retail bank’s back end legacy system.

# Stakeholders

Bank Employees

* New account/New customer executive
* Cashier/Teller

# Scope

The System needs to be able to handle primary functions that can be grouped into two set of activities, with detailed requirement of each group explained in the requirement section.

New account executive/New customer

* Create, Update, Delete Customer
* Create and Delete Account
* View Customer and Account Status

Cashier/Teller

* Manage deposit, withdraw and transfer
* Get Customer and Account details
* Get Customer-Account Transactions/Get Statement

# Requirement Statements

## For New account executive/New customer

### Login Screen: <common function for any employee role>

To allow only bank employee to be able to login to bank web site

Login can be alphabetical or both alphanumeric with maximum n characters

Password can be alphabetical or both alphanumeric with maximum n characters

### Option of API call or database table for login and password validation

#### Option1- API call

* + **API Call:** https://apphonics.tcs.com/public/login/1.0
  + From login screen, when user clicks submit button, invoke APPhonics login API
  + If successful, next screen can have options to any of the functions listed in “Scope” section

#### Option2- RDBMS table

* + Maintain your own RDBMS table for login and password, instead of API call
  + Create table “userstore” with login, password and timestamp column(s)
  + Read login and password information from table and based on verification, options to allow user to continue to next pages or error message

### Create Customer Screen:

To allow new customer-account executive to create new customer

Input fields on screen:

Customer SSN ID, Customer Name, Age, Address Line1, Address Line2, City, State

Output message on screen after user clicks submit:

“Customer creation initiated successfully” Or Relevant error message to be displayed

**API Call:** Upon submit, Create customer API is invoked [<https://apphonics.tcs.com/customers/1.0>/crtcus]

**[API base URL -** <https://apphonics.tcs.com/customers/1.0>] [**Resource -** /crtcus]

### Update Customer Screen:

To allow account executive to update existing customer information

Before account executive can update anything, he/she should be able to query for existing customer

* 1. Query/Pull existing customer:

Input fields on screen:

Either SSN ID or Customer ID as two input fields, with any one mandatory

Both fields are numeric and should have 9 digits

Output message on screen after user clicks submit:

Display following fields on screen: SSN ID, Customer ID, Name, Address, Age

**API Call**: Upon submit, Get customer details API is invoked [<https://apphonics.tcs.com/customers/1.0>/getcus]

Allow editing of fields, except SSN and Customer ID, by having update/edit button and editable fields.

After editing of one or multiple fields, user can submit updated data

**API Call**: Upon submit, Update customer API is invoked [<https://apphonics.tcs.com/customers/1.0>/updcus]

Output message on screen after user clicks submit:

“Customer update initiated successfully” Or Relevant error message to be displayed

### Delete Customer Screen:

To allow account executive to delete existing customer.

Before account executive can delete customer, he/she should be able to query for existing customer by using query/pull existing customer screen.

Allow account executive to delete customer, with a delete button

**API Call**: Upon submit, Delete customer API is invoked [<https://apphonics.tcs.com/customers/1.0>/delcus]

Output message on screen after user clicks submit:

“Customer deletion initiated successfully” Or Relevant error message to be displayedCreate Account Screen:

To allow new customer-account executive to create new account for existing customer

Input fields on screen:

Customer ID

Drop down – Savings or Current Account

Deposit amount – Integer / Number (No decimals)

Output message on screen after user clicks submit:

“Account creation initiated successfully” Or Relevant error message to be displayed

**API Call:** Upon submit, Create Account API is invoked [<https://apphonics.tcs.com/accounts/1.0>/actcr]

**[API base URL -** <https://apphonics.tcs.com/accounts/1.0>] [**Resource -** /actcr]

### Delete Account Screen:

To allow account executive to delete existing account.

Before account executive can delete account, he/she should be able to query for existing account for a customer by using query/pull existing customer accounts screen. Input is either Customer ID or SSN ID.

Allow account executive to delete account, with a delete button

**API Call**: Upon submit, Delete account API is invoked [<https://apphonics.tcs.com/accounts/1.0>/actdl]

Output message on screen after user clicks submit:

“Account deletion initiated successfully” Or Relevant error message to be displayed

### View Customer Status Screen:

This screen shows rows of customers/SSN IDs and status of create/update/delete status. Each rows can have about five columns, i.e. SSN ID, Status, Message, Last Updated, Refresh Button

This can be either a separate screen or part of Create/Update/Delete Customer screen

There is no input in this screen, but it shows all customers, where the process is in progress or complete.

Refresh button, in last column on each row, allows to pull latest status of each customer

**API Call**: Upon refresh, Get Customer Status API is invoked [<https://apphonics.tcs.com/customers/1.0>/chkcus]

### View Account Status Screen:

This screen shows rows with columns - customer ID, account type, status, message, Last updated and refresh button.

This can be either a separate screen or part of Create/Delete Account screen

There is no input in this screen, but it shows all accounts, where the process is in progress or complete.

There should be refresh button, against each row, to pull latest status of each account

**API Call**: Upon refresh, Get Account Status API is invoked [https://apphonics.tcs.com/accounts/1.0/actck]

### Create database table to store entries for customer and account status

**Option1**: Create single table with column(s) e.g. Customer ID, SSN ID, Account ID, Account type, status, message, Last updated

**Option2**: Create two separate tables – CustomerStatus, AccountStatus

CustomerStatus table columns – SSN ID, Customer ID, Status, Message, Last Updated

AccountStatus table columns – Customer ID, Account ID, Account Type, Status, Message, Last Updated

## For Cashier/Teller

### Get Accounts and Account Details Screen:

To allow cashier to pull information for a specific account of a customer

Input fields:

Customer/SSN ID or Account ID

Output fields:

Displays account(s), as a drop drown if input is customer ID/SSN ID. If account ID is input, then display relevant account information – Customer ID, Account ID, Account type, Balance

**API Call:** Upon submit, invokes Get accounts API [https://apphonics.tcs.com/customers/1.0/actst]

Upon selection of any one account from drop down, invokes Get account details API **(API Call)**

[https://apphonics.tcs.com/accounts/1.0/actdt]

Output fields:

Customer ID, Account ID, Account Type, Account Balance, Amount to deposit

### Deposit Screen:

From account details screen, cashier can have option to deposit, withdraw or transfer money, as three different buttons

In case of deposit screen, input fields:

Deposit amount

Output fields:

Customer ID, Account ID

Balance before deposit

Latest balance

Status message: “Amount deposited successfully”

**API Call:** Upon submit, invokes Deposit API [https://apphonics.tcs.com/accounts/1.0/deposit]

### Withdraw Screen:

From account details screen, cashier can have option to deposit, withdraw or transfer money, as three different buttons

In case of withdraw screen, input fields:

Withdraw amount

Output fields:

Account ID

Balance before withdraw

Latest balance

Status message: “Amount withdrawn successfully” or in case balance is not enough “Withdraw not allowed, please choose smaller amount”

**API Call:** Upon submit, invokes Withdraw API [https://apphonics.tcs.com/accounts/1.0/withdrw]

### Transfer Screen:

From account details screen, cashier can have option to deposit, withdraw or transfer money, as three different buttons

In case of transfer screen, input fields:

Transfer amount, Source Account, Target Account

Output fields:

Source Account ID, Target Account ID

Balance before transfer for both account

Latest balance for both accounts

Status message: “Amount transfer completed successfully” or in case balance is not enough in source account “Transfer not allowed, please choose smaller amount”

**API Call:** Upon submit, invokes Transfer API [https://apphonics.tcs.com/accounts/1.0/transfer]

### Get Statement Screen:

To allow cashier to pull account statement for a customer-account

Input fields:

Account ID, Last N Transactions (option to choose from drop down 1-10) or Start date and End date

Output:

Account statement, with each row showing date, transaction description, credit or debit, balance

**API Call1:** Upon submit, invokes Get Statement API (To get statement between start and end date)

[https://apphonics.tcs.com/accounts/1.0/ministmt]

**API Call2:** Upon submit, invokes Get Last N Transactions API

[https://apphonics.tcs.com/accounts/1.0/laststmt]

# Appendix

## Links to API documentation:

Customer API

<https://apphonics.tcs.com/store/registry/resource/_system/governance/apimgt/applicationdata/provider/685613/SandboxBankCustomer/1.0/documentation/files/Sandbox_bank_customer_api_doc_v1.pdf>

Account API

<https://apphonics.tcs.com/store/registry/resource/_system/governance/apimgt/applicationdata/provider/836639/SandboxBankAccount/2.0/documentation/files/Sandbox_bank_accounts_api_doc_v1.pdf>

## Validations

Validations required for input.

1. SSN, Account ID and Customer ID should be 9 digit

2. Date should be of the format CCYY-MM-DD

3. Amount should always have 00 added as suffix. Please take care of this when amount is retrieved also.

e.g. 100 entered on web page should be sent as 10000 to web API

Vice versa, 10000 received from web API should be displayed as 100 on web page.

|  |  |  |  |
| --- | --- | --- | --- |
| **Customer** |  |  |  |
| Input | SSN | ws\_ssn | 9 digit numeric |
|  | Customer id | ws\_cust\_id | 9 digit numeric |
|  | Name | ws\_name | alphabet |
|  | Address | ws\_adrs | alphnumeric |
|  | Age | ws\_age | 3 digit numeric |
|  |  |  |  |
| **Account** |  |  |  |
| Input | Customer id | ws\_cust\_id | 9 digit numeric |
|  | Account id | ws\_acct\_id | 9 digit numeric |
|  | Account type | ws\_acct\_type | S or C |
|  | Balance | ws\_acct\_balance | number add suffix 00 |
|  | CR data | ws\_acct\_crdate | CCYY-MM-DD |
|  | CR last date | ws\_acct\_lasttrdate | CCYY-MM-DD |
|  | Duration | ws\_acct\_duration | number |
|  |  |  |  |
| **Transactions** |  |  |  |
| Input | Customer id | ws\_cust\_id | 9 digit numeric |
|  | Account type | ws\_accnt\_type | S or C |
|  | Amount | ws\_amt | number |
|  | Transacation date | ws\_trxn\_date | CCYY-MM-DD |
|  | Source Acct type | ws\_src\_typ | S or C |
|  | Target Acct type | ws\_tgt\_typ | S or C |