Arrivo ltd.  4th Floor, Plot – 13, Road – 26, Section – 6, Uttara, Dhaka, Bangladesh.

**eBanking Modular Documentation**

Contents

[Introduction 3](#_Toc428279334)

[Technology 3](#_Toc428279335)

[Time Line 3](#_Toc428279336)

[User Management 3](#_Toc428279337)

[Role Management 3](#_Toc428279338)

[Authentication & Access Control 4](#_Toc428279339)

[User Registration/Sign Up 4](#_Toc428279340)

[Menu Management 4](#_Toc428279341)

[Implementation 4](#_Toc428279342)

[Vendor Management 4](#_Toc428279343)

[Create Vendor 4](#_Toc428279344)

[Integrate Vendor API (Manually) 4](#_Toc428279345)

[Create Vendor Rates (Service and Destination wise) 4](#_Toc428279346)

[System will auto pick the lowest rate vendor for each transaction 5](#_Toc428279347)

[Keep Vendor Log 5](#_Toc428279348)

[Implementation 5](#_Toc428279349)

[Basic Data 5](#_Toc428279350)

[1. Destination 5](#_Toc428279351)

[Implementation 5](#_Toc428279352)

[2. Currency 5](#_Toc428279353)

[Implementation 5](#_Toc428279354)

[Service Management 5](#_Toc428279355)

[Services 5](#_Toc428279356)

[Implementation 6](#_Toc428279357)

[Rate Plan 6](#_Toc428279358)

[Implementation 6](#_Toc428279359)

[Transaction Management 6](#_Toc428279360)

[Transaction Request Pages 6](#_Toc428279361)

[Transaction Queue and update processing status 7](#_Toc428279362)

[Transaction Search page 7](#_Toc428279363)

[1. Manage Transfer: 7](#_Toc428279364)

[2. Transaction History: 7](#_Toc428279365)

[3. Transfer History: 7](#_Toc428279366)

[Implementation 7](#_Toc428279367)

[Payment 8](#_Toc428279368)

[Integrate PayPal 8](#_Toc428279369)

[Search Payment History 8](#_Toc428279370)

[Credit Transfer 8](#_Toc428279371)

[Implementation 8](#_Toc428279372)

[API 8](#_Toc428279373)

[Implementation 8](#_Toc428279374)

[Base URI 8](#_Toc428279375)

[Login 8](#_Toc428279376)

[Registration 8](#_Toc428279377)

[LoginOff 9](#_Toc428279378)

[Login 9](#_Toc428279379)

[Recharge 9](#_Toc428279380)

[Credit Transfer 9](#_Toc428279381)

[Money Transfer 9](#_Toc428279382)

[Top Up 10](#_Toc428279383)

[Transfer History 10](#_Toc428279384)

[Cancel Transaction 10](#_Toc428279385)

[Get Pending Money Transfer 10](#_Toc428279386)

[Mobile App 10](#_Toc428279387)

[1. Connection with Main Server 10](#_Toc428279388)

[2. Services 10](#_Toc428279389)

[Web Service - Registered User 11](#_Toc428279390)

[1. Authentication 11](#_Toc428279391)

[2. User Specific Transaction History 11](#_Toc428279392)

[3. User Specific Payment History 11](#_Toc428279393)

[4. Registered User Services 11](#_Toc428279394)

# Introduction

This document focuses on the main modules of eBanking project. The description, business logic and implementation can be found here.

# Technology

eBanking project is developed in ASP.NET MVC using .net framework 4.5. Authentication is extended using Identity Framework v2. Razor view engine and jQuery are used for view optimization.

# Time Line

# User Management

There are several types of users based on their roles and accessibility. User management consists of the types of users, their roles and enabling/disabling user specific permissions.

## Role Management

User Role Management can create, edit and delete User Roles. Current available user roles in eBanking are

1. System: Have access to error reporting, role management, access control etc.
2. Admin:
   1. Edit specific user attributes (user role, password, details etc.),
   2. Pin Generation/Manage
   3. Create/Edit Currency
   4. Create/Edit Destination
   5. Create/Edit Service
   6. Create/Edit RatePlan
   7. View Full Transaction History
   8. Edit Transactions
3. Customer Support: Can edit transactions
4. Vendor: Can get pending money transfer and change status to complete/failed
5. Reseller: Can have multiple pins
6. Registered:
   1. User Services : Registered user can use the following services for transaction’s –
      1. TopUp
      2. Money Transfer
      3. Credit Transfer
      4. Recharge
   2. Reports: User have access to their own transaction history. This also enables them to cancel any money transfer within a time limit defined by specific service.

## Authentication & Access Control

User authentication and access control is implemented extending the Identity framework. User access can be changed by the **Admin** or **System** user. This is handled out in **Role Management**.

## User Registration/Sign Up

This also carried out and taken care of by Identity framework. Registration process requires a valid mobile phone number including country code. The user password is auto generated and sent to the user through SMS. By default all users are assigned to the role **Registered** User. A **System** user or **Admin** can change the user role in **Role Management**.

## Menu Management

User Role specific set of navigation links are defined in **Links** table. This is used to prepare the main navigation menu for every specific user roles.

## Implementation

Full user management is implemented extended the Asp.Net Identity V2 package. Login and registration process is taken care of by the Identity framework. Identity framework has the following default table –

1. ASPNETUSERS
2. ASPNETROLES
3. ASPNETUSERROLES
4. ASPNETLOGINS
5. ASPNETUSERCLAIMS
6. Links

Access control using **Role Management** is further implemented extending Asp.net User Roles by adding specific access levels to each User Roles. The added table is – **ROLEDETAILS**. This table maps each role id to each Controller/Action combination that the role have access to.

# Vendor Management

Vendors can get pending Money Transfer Requests form Transaction table and process it to Complete or Failed. This ends the life cycle for that transaction.

## Create Vendor

(Not Implemented)

## Integrate Vendor API (Manually)

(Not Implemented)

## Create Vendor Rates (Service and Destination wise)

(Not Implemented)

## System will auto pick the lowest rate vendor for each transaction

(Not Implemented)

## Keep Vendor Log

A log of ‘which transaction requests a vendor is receiving from server’ and ‘what the vendor is submitting’ are kept for future reference. This also ensures the vendors accessibility to get more pending money transfer request.

## Implementation

This module and data set is used by the system for further processing of the T**ransactions**. Services like **Money Transfer** need a **Vendor** to process the Transaction. It has a Vendor Log System implemented to keep record of the changes made by **Vendor**s.

Table Names – VENDOR, VENDORREQUESTLOG, VENDORREQUESTLOGDETAILS  
 Models – Vendor, VendorRequestLog, VendorRequestLogDetails  
 Class – VendorLogHelper (Implemented inside TransactionController)

# Basic Data

## Destination

**Admin** user have access to Create/Edit Destination.

### Implementation

Table name - DESTINATIONS  
 Model name- Destination  
 Controller name - DestinationController

## Currency

**Admin** user have access to Create/Edit Currency

### Implementation

Table name - CURRENCIES  
 Model name- Currency  
 Controller name - CurrencyController

# Service Management

## Services

The events that can be fired/carried out by **Registered** users are primarily called services. These are also categorized as parent-child elements. Current available services are –

1. Top Up
   1. Air Time – 10 BDT
   2. Air Time – 50 BDT
2. Money Transfer
   1. bKash Personal
   2. bKash Agent
   3. DBBL Money
   4. mCash
3. Recharge
   1. Voucher Recharge
   2. Paypal

Here **Top Up, Money Transfer** and **Recharge** are parent services. All the rest are actual services that can be used to initiate a transaction by **Registered** user.

### Implementation

Table name - SERVICES  
 Model - Service   
 Controller – ServiceController

## Rate Plan

**RatePlan** is associated and unique to each Services that are provided by eBanking. This holds some primary information regarding the service like –

1. **Base Currency**: Base currency is unique to each service. All transaction and conversion of currency is dependent on Base Currency and Conversion Rate.
2. **Conversion Rate**: Conversion rate defines the currency conversion when required for any service.
3. **Request Cancellation Time Out:** This is stored as PendingTime in database and defined in minutes. A user will be able to cancel his money transfer request.
4. **Bank/Service Commission:** Services might require processing fee. This is saved as **ServiceCharge** in RatePlan Table.

### Implementation

Table Name – RATEPLANS  
 Controller – RatePlanController  
 Model - RatePlan

# Transaction Management

All the credit/money related services are carried out by **Transaction**. Every transaction is executed using one of the **Service**s that are available for **Registered** users.

## Transaction Request Pages

Each service have their own Transaction Request Page. This is specifically designed for **Registered User**s. Current Available pages are –

1. Top Up: **Registered** Users can purchase phone talk time with this page
2. Money Transfer: **Registered** Users can send money through specific Bank/Agent service to others
3. Recharge: **Registered** Users can recharge their eBanking account
4. Credit Transfer: **Registered** Users can transfer their eBanking balance to another **Registered** User

## Transaction Queue and update processing status

Transactions have total four statuses.

1. Pending: assigned to a transaction when it is successfully created (after the expiration of TimeOut if exists for that particular service)
2. Canceled: when the **Registered** user cancels the transaction
3. Processing: when the transaction is sent to a vendor for further processing
4. Complete: when the transaction is successful
5. Failed: if any error occurs and the transaction couldn’t be complete

## Transaction Search page

Three separate pages allow transaction search.

### Manage Transfer:

Admin have access to this module. This gives access to all **Money Transfer** transactions. They can change the statuses except when the transaction is already **Canceled/Complete/Failed.**

### Transaction History:

This allow the **Admin** to monitor and check all kind of transactions. No changes can be made from here.

### Transfer History:

This allow the **Registered** user to monitor all his **Money Transfer** records and provides the option to **Cancel** if it has a remaining **TimeOut** Limit.

## Implementation

This process needs access to DB tables – **Transactions, Service, RatePlan, Currency, Destination, User, Vendor, VendorRequestLog, VendorRequestLogDetails, Pins,** and **StatusMsgs. Service**s can be considered as the types for processing **Transaction**s. Base components for implementing **Transaction** are –

Table Name - TRANSACTIONS

Controller Name - TransactonController

Model Name – Transaction

Different type of transaction uses different set of components to execute. All type of transactions use **User** table for authentication and **User Balance.** It also uses **Currency** and **Destination** for local specifications; **Service**s and **Rateplan** for specific transaction types; **Sequencers** asserial number of that type of transaction on that day; and **StatusMsgs** to keep record of the transaction process flow. Here is a list of most important **Transaction** types that uses additional Model/Table/Controller -

1. **Money Transfer:-** No additional data
2. **Credit Transfer:-** Don’t need **Currency** and **Destination**
3. **TopUp:-** No additional data
4. **Recharge – VoucherRecharge:-** Needs access to **Pins**(Model/Table/Controller)

# Payment

## Integrate PayPal

(Not Implemented)

## Search Payment History

(Not Implemented)

## Credit Transfer

**Registered** users can transfer eBanking credit to other **Registered** users.

## Implementation

**Payment** is also another type of Transaction. So it is implemented within **Transaction**.

# API

## Implementation

A Web API is developed for outside platforms to remotely and securely interact with the core server software. The Purpose and links are provided below.

### Base URI

Local with debug - <http://localhost:36032/>  
 Global (published) - <http://117.58.243.230/eBanking/>  
 Local (published) - <http://192.168.1.103:81/eBanking/>

### Login

Type - POST   
URI - Account/ApiLogin  
Data Format - { "UserName":UserName,

"Password":Password,

"ApkVersionName":VERSION

}

### Registration

Type - POST   
URI - Account/ApiRegister  
Data Format - { "UserName":UserName

}

### LoginOff

Type - GET   
URI - Account/ApiLoginOff

### Login

Type - POST   
URI - Account/ApiLogin  
Data Format - { "UserName":UserName,

"Password":Password,

"ApkVersionName":VERSION

}

### Recharge

Type - POST  
URI - api/ServiceApi/ReCharge  
Data format - {"PinCode":PinCode}

### Credit Transfer

Type - POST

URI - api/ServiceApi/CreditTransfer  
Data format - { "ToUser":ToUser,

"Balance":Balance

}

### Money Transfer

Type - GET

URI - api/ServiceApi/MoneyTransfer

Description - this will get the required data from server to perform money transfer.

Type - POST

URI - api/ServiceApi/MoneyTransfer  
 Data format - { "FromCurrencyId”:FromCurrencyId,

"MobileNo”:MobileNo,

"ServiceId":ServiceId,

"Amount":Amount,

"AmountInUSD":AmountInUSD,

"TotalAmount":TotalAmount,

"ProcessingFee":ProcessingFee

}

Description - this will post the full formatted data to server to perform money transfer.

### Top Up

Type - GET   
URI - api/ServiceApi/TopUpApi  
Description - this will get the required data from server to perform Top Up.

Type - POST  
URI - api/ServiceApi/TopUpApi  
Data format - { "AmountInUSD":AmountInUSD,

"FromCurrencyId":FromCurrencyId,

"ProcessingFee":ProcessingFee,

"RatePlanId":RatePlanId,

"ServiceId": ServiceId,

"Value": Value,

"TotalAmount":TotalAmount,

"ToUser":ToUser, "ValueId":ValueId

}

Description - this will post the full formatted data to server to perform TopUP.

### Transfer History

Type - GET   
URI - api/ServiceApi/TransferHistory

### Cancel Transaction

Type - GET   
URL - api/ServiceApi/CancelTransaction

URL Parameter – Transaction Id

### Get Pending Money Transfer

Type - GET

URI - api/ServiceApi/PendingMoneyTransferApi  
URL Parameter – Transaction Id

# Mobile App

## Connection with Main Server

Mobile app uses web api services to connect with the main server.

## Services

* 1. Login/Registration
  2. Current Balance
  3. Money Transfer
  4. Credit Transfer
  5. Top Up
  6. Recharge
  7. Money Transfer Record & Cancelation option

# Web Service - Registered User

### Authentication

Login/Registration portal are active for requesting authentication from Web API

### User Specific Transaction History

Have access to monitor **Money Transfer** transactions only and can cancel individual Money Transfer Transactions within TimeOut Limit.

### User Specific Payment History

(Not Implemented)

## Registered User Services

* 1. Money Transfer
  2. Credit Transfer
  3. Top Up
  4. Recharge