



PAY ARENA

# Solution Design Document (SDD)

<b>Process ID</b>	P-001
<b>PU</b>	EPO
<b>Process Group</b>	EPO
<b>Process Name</b>	Pay Arena
<b>Document ID</b>	1.0

<b>Version</b>	0.1
<b>Status</b>	Initial Version
<b>Date</b>	24-12-2022

# Contents

Contents .....	2
Document Control .....	3
Template .....	3
Completion stages.....	3
Version history.....	3
1. Introduction .....	4
1.1. Document purpose .....	4
1.2. Process summary .....	4
1.3. Reference artefacts .....	4
2. Solution Overview .....	5
2.1. High level design (HLD).....	5
2.2. Solution description .....	5
3. Solution Detail.....	15
3.1. Object model .....	15
3.2. Solution components .....	16
4. Operations .....	16
4.1. Business exceptions.....	16
4.2. System exceptions .....	16
4.3. Management information (MI).....	<b>Error! Bookmark not defined.</b>
4.4. Scheduling and manual execution .....	17
4.5. Optimization and scaling.....	17
4.6. Alerting .....	17
4.7. Logging .....	17
5. Data Management.....	18
5.1. Storage.....	18
5.2. Privacy .....	18
5.3. Security .....	18
5.4. Preservation .....	18
6. Considerations .....	19
i. Business Glossary.....	20
ii. Attachments .....	21

# Document Control

## Template

The following person(s) own the format and information requested in this document template.

Team	Name	Relevant Sections

The version history of this template is as follows.

Version	Date	Author	Change Summary
0.1	24/12/2022	Bakare Sodiq	• Initial Version

## Completion stages

This document will be completed as per the following sign-off points.

Version	Phase	Relevant Sections	Producer	Sign-off
0.1	Design	All	Delivery Team	Process Owner ( )

## Version history

This document's change history is as follows.

Version	Date	Author	Approver	Change Summary
0.1	24/01/2022	Bakare Sodiq		

# 1. Introduction

## 1.1. Document purpose

The purpose of the Solution Design Document (SDD) is to describe the technical solution developed to meet the requirements outlined in the Process Design Document (PDD), including any technical prerequisites and considerations required to deploy, operate, and maintain the process. It is a living document that is incrementally developed as the technical solution is built and is finalized prior to deployment into the production automation infrastructure.

This document will refer to the automation package ("The Solution" or "Solution") throughout, which represents the Business Objects and Processes, as well as any other peripheral technical components (e.g credentials, templates, databases etc.) used to deliver the automated process.

## 1.2. Process summary

The Agency Banking captures the steps for settling all account transactions done on Visa Transactions. This process is aimed at settling different merchants for transactions processed on the Unified Payment platform at our First Bank branches

## 1.3. Reference artefacts

The Process Design Document (PDD) which captures the business-related details of the process being automated and describes how the automated process is intended to work, including risk and data management control has been included as a link in this manual

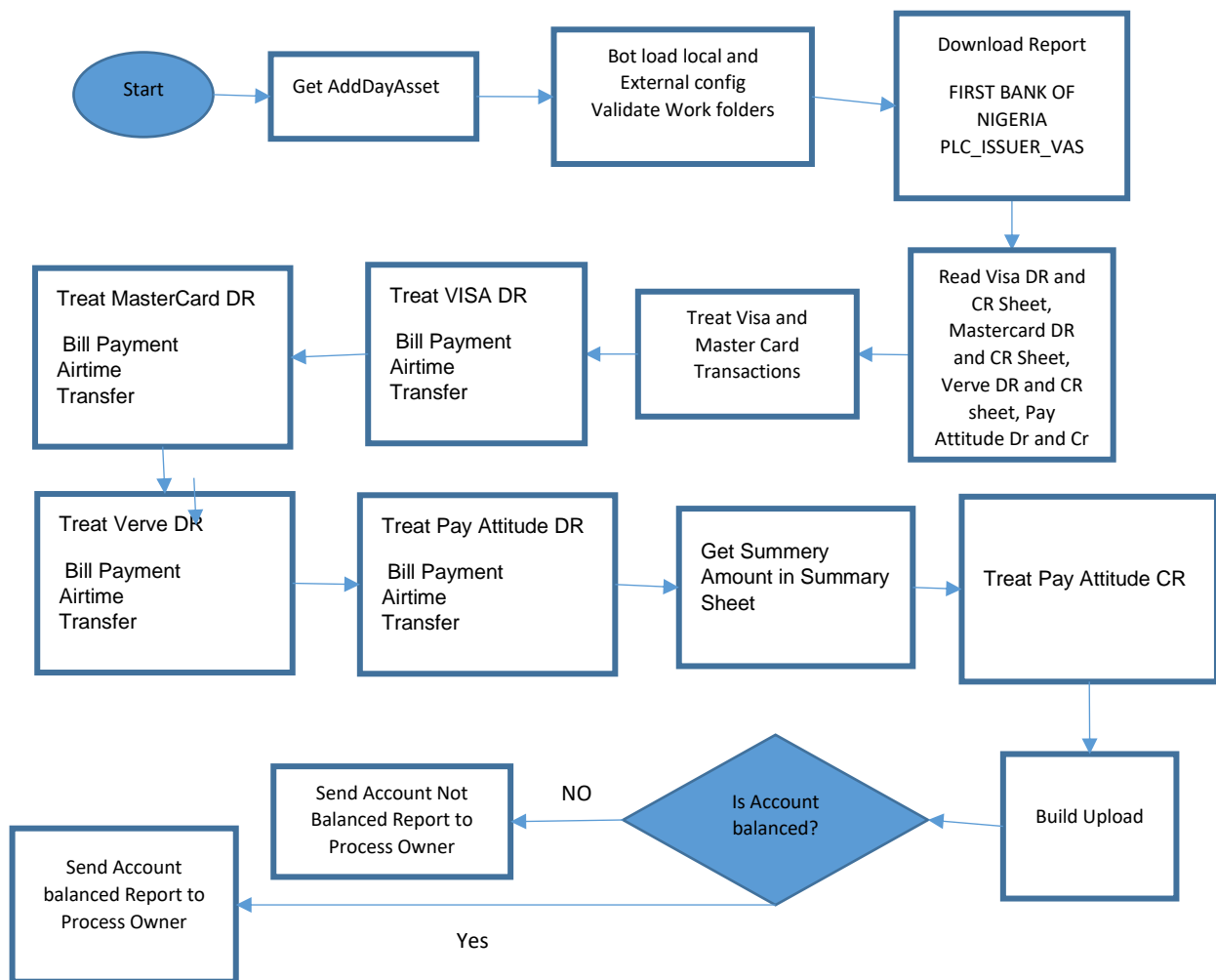
The following artefacts should be read in conjunction with this document.

Title	Version	Location
<b>Process Design Document (PDD) – Pay Arena</b>	0.1	

## 2. Solution Overview

This section describes the high-level design of the automated solution.

### 2.1. High level design (HLD)



### 2.2. Solution description

- Bot Kill all application
- Bot check the day set on Orchestrator, if it is Monday; Bot will run Saturday, Sunday and Monday. Else the bot only treats the day.
- Sequence: Load all Config Portal
  - Load Internal Config Values: These load all assets in the internal config excel In the project directory. Output Argument: Config
  - Validate Work Folders: Create folder its does not exist
  - Load External Config workflow: Read system settings output Argument Sysfig
  - Delete work folders and validate again

- Assign report Date
  - Sequent to check for Monday Logic
- Download Logic workflow
  - Assign all WF Variables: all needed variables
  - Pre-Load
    - Try Catch FTP Credentials
    - Create Processing File
  - Invoke CheckReportPath workflow
  - Sequence: Dowloa
    - If Report not Downloaded:
      - Sequence: Login and Download
      - Do While: Try to login 3 Times
      - Logout from FTP workflow
      - Login FTP workflow
      - If Login Successfully: Download Report FTP
      - Sequence If Can't Login: Exception error throw
      - If File Unable to Download: Exception error throw
  - READ ALL SHEET
  - Delete Download File
- RUN PAYARENA
  - DR Sheets Sequence
    - Visa DR Sequence
    - Read Sheet
    - Filter Visa DR by Bill payment Sequence
      - Treat Bill Payment
      - Process Bill Payment Debit
      - Get the total Sum
        - Write Range: visaDRBillPayment
        - Sum Sequence
          - Assign out\_VisaDRBillPYMTIssuerFee:  
Convert.ToDouble(visaDRBillPayment.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER FEE (IRF) VALUE"].ToString()))))
          - Assign out\_VisaDRBillPYMTIssuerVat:  
Convert.ToDouble(visaDRBillPayment.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER VAT"].ToString()))))
          - Assign out\_visaDRBillPYMTActualAmt:  
Convert.ToDouble(visaDRBillPayment.AsEnumerable().Sum(row => Convert.ToDouble(row["ACTUAL AMOUNT SETTLED"].ToString()))))
          - Assign out\_VisaDRBillPYMTNewFee:  
Convert.ToDouble(visaDRBillPayment.AsEnumerable().Sum(row => Convert.ToDouble(row["NEW FEE"].ToString()))))
          - Assign out\_VisaDRBillPYMTVat:  
Convert.ToDouble(visaDRBillPayment.A

- sEnumerable().Sum(row => Convert.ToDouble(row["VAT"].ToString()))
    - Assign out\_VisaDRBillPYMTEExtraVat: Convert.ToDouble(visaDRBillPayment.AsEnumerable().Sum(row => Convert.ToDouble(row["EXTRA VAT"].ToString()))))
  - Filter Visa DR by Transfer Sequence
    - Filter Visa DR Transfer Trans Type
    - Process Transfer Debit workflow
    - Write Range: visaDRTransfer
    - Sum Sequence
      - Assign out\_visaDRTransferIssuerFee: Convert.ToDouble(visaDRTransfer.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER FEE (IRF) VALUE"].ToString()))))
      - Assign out\_visaDRTransferActualAmt: Convert.ToDouble(visaDRTransfer.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER VAT"].ToString()))))
      - Assign out\_visaDRTransferNewFee: Convert.ToDouble(visaDRTransfer.AsEnumerable().Sum(row => Convert.ToDouble(row["ACTUAL AMOUNT"].ToString()))))
      - Assign out\_visaDRTransferVat Convert.ToDouble(visaDRTransfer.AsEnumerable().Sum(row => Convert.ToDouble(row["RATE"].ToString()))))
  - Filter Visa DR by Airtime top Sequence
    - Filter Visa DR Airtime Trans Type
    - Treat Airtime
    - Mutiple Assign
      - Assign out\_visaDRAirtimeIssuerFee: Convert.ToDouble(visaDRAirtime.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER FEE (IRF) VALUE"].ToString()))))
      - Assign out\_visaDRAirtimeIssuerVat: Convert.ToDouble(visaDRAirtime.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER VAT"].ToString()))))
      - Assign out\_visaDRAirtimeTranAmt: Convert.ToDouble(visaDRAirtime.AsEnumerable().Sum(row => Convert.ToDouble(row["TRANAMOUNT"].ToString()))))
  - Verve DR Sequence
  - Read Sheet: verveDRworkLogDT
  - Filter Verve DR by Bill payment Sequence
    - Treat Bill Payment
    - Process Bill Payment Debit
    - Get the total Sum
      - Write Range: verveDRworkLogDT

- Sum Sequence
  - Assign out\_verveDRBillPYMTIssuerFee:  
Convert.ToDouble(verageDRBillPayment.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER FEE (IRF) VALUE"].ToString())))
  - Assign out\_verveDRBillPYMTIssuerVat:  
Convert.ToDouble(verageDRBillPayment.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER VAT"].ToString())))
  - Assign out\_verveDRBillPYMTActualAmt:  
Convert.ToDouble(verageDRBillPayment.AsEnumerable().Sum(row => Convert.ToDouble(row["ACTUAL AMOUNT SETTLED"].ToString())))
  - Assign out\_verveDRBillPYMTNewFee:  
Convert.ToDouble(verageDRBillPayment.AsEnumerable().Sum(row => row.Field<double>("NEW FEE").ToString())))
  - Assign out\_VerveDRBillPYMTVat:  
Convert.ToDouble(verageDRBillPayment.AsEnumerable().Sum(row => Convert.ToDouble(row["VAT"].ToString())))
  - Assign out\_verveDRBillPYMTExtraVat:  
Convert.ToDouble(verageDRBillPayment.AsEnumerable().Sum(row => Convert.ToDouble(row["EXTRA VAT"].ToString())))
- Filter Verve DR by Transfer Sequence
  - Filter Verve DR Transfer Trans Type
  - Treat Transfer
  - Mutiple Assign
    - Assign out\_verveDRTransferIssuerFee:  
Convert.ToDouble(verageDRTransfer.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER FEE (IRF) VALUE"].ToString())))
    - Assign out\_verveDRTransferIssuerVat:  
Convert.ToDouble(verageDRTransfer.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER VAT"].ToString())))
    - Assign out\_verveDRTransferActualAmt:  
Convert.ToDouble(verageDRTransfer.AsEnumerable().Sum(row => Convert.ToDouble(row["ACTUAL AMOUNT"].ToString())))
    - Assign out\_verveDRTransferRate:  
Convert.ToDouble(verageDRTransfer.AsEnumerable().Sum(row => Convert.ToDouble(row["RATE"].ToString())))



- Assign out\_verveDRTransferVat:  
Convert.ToDouble(verageDRTransfer.AsEnumerable().Sum(row => Convert.ToDouble(row["VAT"].ToString()))))
- Filter Verve DR by Airtime top Sequence
  - Filter Visa DR Airtime Trans Type
  - Treat Airtime
  - Mutiple Assign
    - Assign out\_verveDRAirtimeIssuerFee:  
Convert.ToDouble(verageDRAirtime.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER FEE (IRF) VALUE"].ToString()))))
    - Assign out\_verveDRAirtimeIssuerVat:  
Convert.ToDouble(verageDRAirtime.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER VAT"].ToString()))))
    - Assign out\_verveDRAirtimeTranAmt:  
Convert.ToDouble(verageDRAirtime.AsEnumerable().Sum(row =>  
Convert.ToDouble(row["TRANAMOUNT"].ToString()))))
- Mastercard DR Sequence
- Read Sheet: mastercardDRworkLogDT
- Filter Mastercard DR by Bill payment Sequence
  - Treat Bill Payment
  - Process Bill Payment Debit
  - Get the total Sum
    - Write Range: mastercardDRworkLogDT
    - Sum Sequence
      - Assign out\_NMCDRBillPYMTIssuerFee:  
Convert.ToDouble(mastercardDRBillPayment.AsEnumerable().Sum(row =>  
Convert.ToDouble(row["ISSUER FEE (IRF) VALUE"].ToString()))))
      - Assign out\_NMCDRBillPYMTIssuerVat:  
Convert.ToDouble(mastercardDRBillPayment.AsEnumerable().Sum(row =>  
Convert.ToDouble(row["ISSUER VAT"].ToString()))))
      - Assign out\_NMCDRBillPYMTActualAmt:  
Convert.ToDouble(mastercardDRBillPayment.AsEnumerable().Sum(row =>  
Convert.ToDouble(row["ACTUAL AMOUNT SETTLED"].ToString()))))
      - Assign out\_NMCDRBillPYMTNewFee:  
Convert.ToDouble(mastercardDRBillPayment.AsEnumerable().Sum(row =>  
Convert.ToDouble(row["NEW FEE"].ToString()))))
      - Assign out\_NMCDRBillPYMTVat:  
Convert.ToDouble(mastercardDRBillPayment.AsEnumerable().Sum(row =>

- Convert.ToDouble(row["VAT"].ToString()))
    - Assign out\_NMCDRBillPYMTEExtraVat:  
Convert.ToDouble(mastercardDRBillPayment.AsEnumerable().Sum(row => Convert.ToDouble(row["EXTRA VAT"].ToString()))))
  - Filter Mastercard DR by Transfer Sequence
    - Filter Mastercard DR Transfer Trans Type
    - Treat Transfer
    - Mutiple Assign
      - Assign out\_NMCDRTransferIssuerFee:  
Convert.ToDouble(mastercardDRTransfer.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER FEE (IRF) VALUE"].ToString())))
      - Assign out\_NMCDRTransferIssuerVat:  
Convert.ToDouble(mastercardDRTransfer.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER VAT"].ToString())))
      - Assign out\_NMCDRTransferActualAmt:  
Convert.ToDouble(mastercardDRTransfer.AsEnumerable().Sum(row => Convert.ToDouble(row["ACTUAL AMOUNT"].ToString())))
      - Assign out\_NMCDRTransferRate:  
Convert.ToDouble(mastercardDRTransfer.AsEnumerable().Sum(row => Convert.ToDouble(row["RATE"].ToString())))
      - Assign out\_NMCDRTransferVat:  
Convert.ToDouble(mastercardDRTransfer.AsEnumerable().Sum(row => Convert.ToDouble(row["VAT"].ToString())))
  - Filter Mastercard DR by Airtime top Sequence
    - Filter Mastercard DR Airtime Trans Type
    - Treat Airtime
    - Mutiple Assign
      - Assign out\_NMCDRAirtimeIssuerFee:  
Convert.ToDouble(mastercardDRAirtime.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER FEE (IRF) VALUE"].ToString())))
      - Assign out\_NMCDRAirtimeIssuerVat:  
Convert.ToDouble(mastercardDRAirtime.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER VAT"].ToString())))
      - Assign out\_NMCDRAirtimeTranAmt:  
Convert.ToDouble(mastercardDRAirtime.AsEnumerable().Sum(row => Convert.ToDouble(row["TRANAMOUNT"].ToString())))
- PayAttitude DR Sequence
- Read Work Log: payattitudeDRworkLogDT
- Filter PayAttitudeDR by masked pan "901105" and "901102"
- 901105 PayAttitude masked pan Sequence

- Filter Payattitude DR for 901105
- Process PayAttitude
- Write Range
- Sum Sequence
  - Assign out\_PayattitudeDrOriginalAmount:  
Convert.ToDouble(payattitudeDR901105.AsEnumerable().Sum(row => Convert.ToDouble(row["ORIGINAL AMOUNT"].ToString()))))
  - Assign out\_PayattitudeDrNewFee:  
Convert.ToDouble(payattitudeDR901105.AsEnumerable().Sum(row => Convert.ToDouble(row["NEW FEE"].ToString()))))
  - Assign out\_PayattitudeDrIssuerVat:  
Convert.ToDouble(payattitudeDR901105.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER VAT"].ToString()))))
  - Assign out\_PayattitudeDrIssuerFee:  
Convert.ToDouble(payattitudeDR901105.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER FEE (IRF) VALUE"].ToString()))))
- 901102 PayAttitude masked pan Sequence
  - Filter PayAttitudeDR for 901102 mased pan
  - Treat Masked Pan 901102
    - Assign out\_PayattitudeDR901102TranAmt:  
Convert.ToDouble(payattitudeDR901102.AsEnumerable().Sum(row => Convert.ToDouble(row["TRANAMOUNT"].ToString()))))
    - Assign out\_PayattitudeDrIssuerFee:  
out\_PayattitudeDrIssuerFee +  
Convert.ToDouble(payattitudeDR901102.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER FEE (IRF) VALUE"].ToString()))))
    - Assign out\_PayattitudeDrIssuerVat:  
out\_PayattitudeDrIssuerVat+  
Convert.ToDouble(payattitudeDR901102.AsEnumerable().Sum(row => Convert.ToDouble(row["ISSUER VAT"].ToString()))))
- Summary Sheet Sequence
  - Read Range: summaryDT
  - Invoke code: To get out\_summary
- Payattitude CR Sequence
  - Assign out\_PayattitudeCrIssuerFee:  
Math.Abs(Convert.ToDouble(payattitudeCRworkLogDT.AsEnumerable().Where(x => Double.TryParse(x["ISSUER FEE (IRF) VALUE"].ToString(), out test) == true).Sum(row => Convert.ToDouble(row["ISSUER FEE (IRF) VALUE"].ToString()))))
  - Assign out\_PayattitudeCrIssueVat:  
Math.Abs(Convert.ToDouble(payattitudeCRworkLogDT.AsEnumerable().Where(x => Double.TryParse(x["ISSUER VAT"].ToString(), out test) == true).Sum(row => Convert.ToDouble(row["ISSUER VAT"].ToString()))))

- Assign out\_PayattitudeCrTransAmount:  
`Math.Abs(Convert.ToDouble(payattitudeCRworkLogDT.AsEnumerable().Where(x => Double.TryParse(x["TRANAMOUNT"].ToString(), out test) == true).Sum(row => Convert.ToDouble(row["TRANAMOUNT"].ToString()))))`
- Build Upload File
  - Multiple Assign: all needed variables
  - Red Work Log: DT
  - Build Upload File Data Table: UploadFileDT
  - Update Summary Amount
    - GetMerchantDetails workflow: PAYARENA BILL PYMT DR vat/extra vat
    - Confrim on Finacle or N\EFT
      - DB\_Connection workflow
      - QueryFinacleDB workflow
      - If Result is Not Found in Finacle
        - Check NEFT
          - Sequence: Get NEFT from Storge Bucket
          - Read PDF NEFT Report
          - Generate Data Table
          - Assign ResultDataRow
          - If DataRow is not Null
            - Sequence: Get Amount
            - Assign SplitRow:  
`ResultDataRow["Column1"].ToString().Split(' ')`
            - Assign Amount:  
`Convert.ToDouble(SplitRow[6].ToString())`
      - If Summary Amount not found
        - Throw BRE : new  
`BusinessRuleException("Summary Amount not found the in Finacle and NEFT report")`
  - Update Debit Transaction
    - Bill Payment DR Actual Amount
      - GetMerchantDetails workflow: "PAYARENA BILL PYMT SETTLE DR Actual Amt"
      - Update VisaDRBillPYMTActualAmt Row
      - Update Verve Actual Amount Row
      - Update Mastercard Actual AmountRow
    - Bill Payment DR NewFee
      - GetMerchantDetails workflow: "PAYARENA BILL PYMT SETTLE DR New Fee"
      - Update Visa Fee
      - Update Verve Fee
      - Update Mastercard Fee
    - Bill Payment DR Vat and Extra Vat
      - GetMerchantDetails workflow: "PAYARENA BILL PYMT DR vat/extra vat"
      - Visa vat and Extra vat

- Update BILL PAYMT VAT
  - Update BIL Pay Extra VAT
- Verve vat and Extra vat
  - Update in\_VerveDRBillPYMTVat
  - Update in\_verveDRBillPYMTExtraVat
- Mastercard vat and Extra vat
  - Update in\_NMCDRBillPYMTVat
  - Update in\_NMCDRBillPYMTExtraVat
- Airtime DR
  - Airtime DR Tran Amt
    - GetMerchantDetails workflow: PAYARENA DR Tran Amt
    - Update Visa Tran Amt
    - Mastercard Tran Amt
    - Update Verve Tran Amt
  - Airtime Issuer Fee
    - GetMerchantDetails workflow PAYARENA DR Issuer Fee
    - Update VisaDRAirtimeIssuerFee
    - Update NMCDRAirtimeIssuerFee
    - Update VerveDRAirtimeIssuerFee
  - Airtime Issuer Vat
    - GetMerchantDetails workflow: "PAYARENA DR Issuer vat"
    - Update Update VisaDRAirtimeIssuerVat
    - Update NMCDRAirtimeIssuerVat
    - Update VerveDRAirtimeIssuerVat
- Transfer DR
  - Transfer DR Actual Amt
    - GetMerchantDetails workflow: PAYARENA TRANSFER ACTUAL AMT
    - Update in\_visaDRTransferActualAmt
    - Update Mastercard Actual Amt
    - Update Verve Actual Amt
  - Transfer DR New Fee
    - GetMerchantDetails workflow: "PAYARENA Transfer New fee"
    - Update Visa Transfer New Fee
    - Update Mastercard New Fee
    - Update Verve New Fee
  - Update Transfer DR Vat
    - GetMerchantDetails workflow: "PAYARENA BILL PYMT DR vat/extra vat"
    - Update Visa Transfer Vat
    - Update Mastercard Vat
    - Update Verve New Vat
  - Update Transfer Issuer Fee
    - GetMerchantDetails workflow: "PAYARENA DR Issuer Fee"
    - Update in\_visaDRTransferIssuerFee
    - Update in\_NMCDRTransferIssuerFee

- Upadte in\_erveDRTransferIssuerFee
- Update Transfer Issuer Fee
  - GetMerchantDetails workflow: "PAYARENA DR Issuer vat"
  - Upadte in\_visaDRTransferIssuerVat
  - Update in\_NMCDRTransferIssuerVat
  - Upadte in\_erveDRTransferIssuerVat
- Payatitude DR
  - PayAttitude 901105
    - GetMerchantDetails workflow: PAYATTITUDE Actual Amt"
    - Update in\_payattDR901105ActualAmt
  - 901105 DR New Fee
    - GetMerchantDetails workflow: "PAYATTITUDE FEE"
    - Update in\_payattDR901105NewFee
  - PayatDr Issuer Fee
    - GetMerchantDetails workflow: "PAYATTITUDE Issuer Fee"
    - Update payatt IssuerFee
  - PayatDr Issuer Vat
    - GetMerchantDetails workflow: "PAYATTITUDE Issuer Vat"
    - Update in\_payDRIssuerVat
  - 901102 DR Tran Amount
    - GetMerchantDetails workflow: "PAYATTITUDE TraAmt 901102"
    - Update payDR901102TranAmt
- Update Credit Sequence
  - Reversal Part for Visa, MasterCard and Verve (Need to be updated)
  - \*\*\*\*\*
  - Visa, NMC and Verve CR Issuer vat and fee
    - GetMerchantDetails workflow: "PAYARENA BILL PYMT DR vat/extra vat"
    - CR Issuer Fee
      - Update CR Issuer Fee
      - Update verveCRIssuerFee
      - Update NMCCRissuerFee
    - CR Issuer Vat
      - Update visaCRIssuerVat
      - Update verveCRIssuerVat
      - Update NMCCRissuerVat
  - Pay Credit Transaction
    - Payatt CR Tran Amount
      - GetMerchantDetails workflow: "PAYATTITUDE DR/CR Tran Amt"
      - Update payAttCR TranAmt
    - Payatt CR Fee
      - GetMerchantDetails workflow: "PAYATTITUDE DR/CR Tran Amt"

- Update payAttCR TranAmt
  - Payatt CR Vat
    - Mutiple Assign
    - Update payAttCR TranAmt
  - END the Upload File
    - Update Row
  - Assign debit:
 

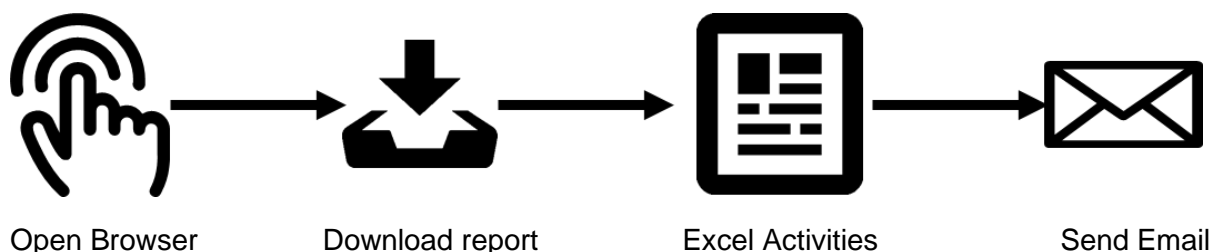
```
Convert.ToDecimal(UploadFileDT.AsEnumerable().Where(x =>
x.Field<string>("Type") ==
"D").Sum(z=>z.Field<decimal>("Amount").ToString()))
```
  - Assign Credit :
 

```
Convert.ToDecimal(UploadFileDT.AsEnumerable().Where(x =>
x.Field<string>("Type") ==
"C").Sum(z=>z.Field<decimal>("Amount").ToString()))
```
  - If Account is Balanced:  $\text{Math.Round}(\text{debit}, 1) - \text{Math.Round}(\text{credit}, 1) == 0$ 
    - Assign out\_IsBalanced : True
  - Else
    - Assign out\_IsBalanced : False
  - Assign out\_UploadFileName
  - Copy File: Uploade Template File to Processing folder
  - Write Range: UploadFileDT
- Is Account Balanced?
  - If True:
    - Send Job Report workflow: Account is Balanced
  - Else
    - Send Job Report workflow: Account not Balanced

### 3. Solution Detail

This section describes the low-level design of the automated solution.

#### 3.1. Object model



## 3.2. Solution components

Ref.	Type	Name	Purpose
<b>C1</b>	Process	Open Application	Opens unified Payment portal to download the required reports
<b>C2</b>	Process	Excel Activities	<ul style="list-style-type: none"><li>• Read Excel document into datatable</li><li>• Add rows</li><li>• Sort document by columns</li><li>• Perform computations i.e Add, Subtract, Multiply</li></ul>
<b>C3</b>	Process	Send Email	<ul style="list-style-type: none"><li>• Attach excel file to email</li><li>• Send email</li></ul>

## 4. Operations

This section describes the controls, reporting and alerting required to operate the solution.

### 4.1. Business exceptions

Events classified as Business Exceptions are those that are not expected to be handled by the virtual worker. That is, they are out of scope of what is described in the PDD.

Business Exceptions are marked as follows.

No	Exception	Solution
<b>1</b>	<b>Change in report Format</b> <ul style="list-style-type: none"><li>- Change in number of columns.</li></ul>	<ul style="list-style-type: none"><li>• Send email alert to process owner</li></ul>
<b>2</b>	<b>Inability to find report</b> <ul style="list-style-type: none"><li>- The bot is unable to download report due to change in naming convention of the required report.</li></ul>	<ul style="list-style-type: none"><li>• Send email alert to process owner</li></ul>
<b>2</b>	<b>Inability to find report</b> <ul style="list-style-type: none"><li>- The bot is unable to download report due to change in naming convention of the required report.</li></ul>	<ul style="list-style-type: none"><li>• Send email alert to process owner</li></ul>
		<ul style="list-style-type: none"><li>•</li></ul>
		<ul style="list-style-type: none"><li>•</li></ul>

### 4.2. System exceptions

System exceptions can fall in one of two categories:



1. Known system exceptions – which are known problem or risky areas in the process (e.g. to common system unreliability) that have been specifically catered for with extra retries or redundancies, or at least a specific error description.
2. Unknown system exceptions – which are unplanned errors.

### Known system exceptions

System Exceptions with specific catches are marked as follows.

Scenario	Work Queue	Status	Tags	Required Action
N/A	N/A	N/A	N/A	N/A

### Unknown system exceptions

Unknown System Exceptions will be represented as follows.

Scenario	Work Queue	Status	Tags	Required Action
N/A	N/A	N/A	N/A	N/A

## 4.4. Scheduling and manual execution

The robot will run everyday.

## 4.5. Optimization and scaling

To be determined.

## 4.6. Alerting

Any alerting built into The Solution is described below as per PDD specification.

Ref.	Scenario	Method	Recipient(s)
AL1	Bot is unable to find Approved/Failed Transaction report	Send Email	Process owner
AL2	Bot has completed its execution	Send Email	Process owner

## 4.7. Logging

- There is an in-built audit trail which captures actions and timelines of robot's activities

## 5. Data Management

### 5.1. Storage

The downloaded files i.e Approved/Failed Transactions report and the final report will be stored in a specified folder on the system/server running the bot

### 5.2. Privacy

The bot will not transmit documents/files to external locations (outside Firstbank Bank) and access will be restricted to assigned members of the E-business team

### 5.3. Security

At a specified date (to be determined by members of the COE), the bot will delete all downloaded reports from its download folder.

### 5.4. Preservation

The bot will log onto the applications using credentials supplied by Firstbank bank's IT

## 6. Considerations

- Stable internet connectivity will be readily available for the bot to function
- Any changes in the naming conventions of documents which are downloaded by the bot may require some updates to robot configuration/process design

## i. Business Glossary

Acronyms and terms used throughout this document are described below.

Acronym or Term	Synonym(s)	Full Description
<b>VW</b>	PAC, Robot, Bot	Virtual Worker
<b>HW</b>		Human Worker
<b>PDD</b>		Process Design Document
<b>SDD</b>		Solution Design Document
<b>VM</b>	VM	Virtual Machine
<b>VDI</b>	VDI	Virtual Desktop Interface
<b>RPA</b>	RPA	Robotic Process Automation
<b>SSO</b>		Single Sign-On

## ii. Attachments

The following attachments relate to this document.

Attachment	Description