



MIGS VISA SETTLEMENT

Solution Design Document (SDD)

Process ID	P-001
PU	EPO
Process Group	EPO
Process Name	Migs Visa Settlement
Document ID	1.0

Version	0.1
Status	Initial Version
Date	20-12-2022

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.....	9
3.1. Object model	9
3.2. Solution components	9
4. Operations	9
4.1. Business exceptions.....	10
4.2. System exceptions	10
4.3. Management information (MI).....	Error! Bookmark not defined.
4.4. Scheduling and manual execution	11
4.5. Optimization and scaling.....	11
4.6. Alerting	11
4.7. Logging	11
5. Data Management.....	12
5.1. Storage.....	12
5.2. Privacy	12
5.3. Security	12
5.4. Preservation	12
6. Considerations	13
i. Business Glossary.....	14
ii. Attachments	15

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	20/12/2022	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	20/01/2022	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 Migs Visa Settlement process captures the steps for settling all account transactions done on Visa Domestic 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

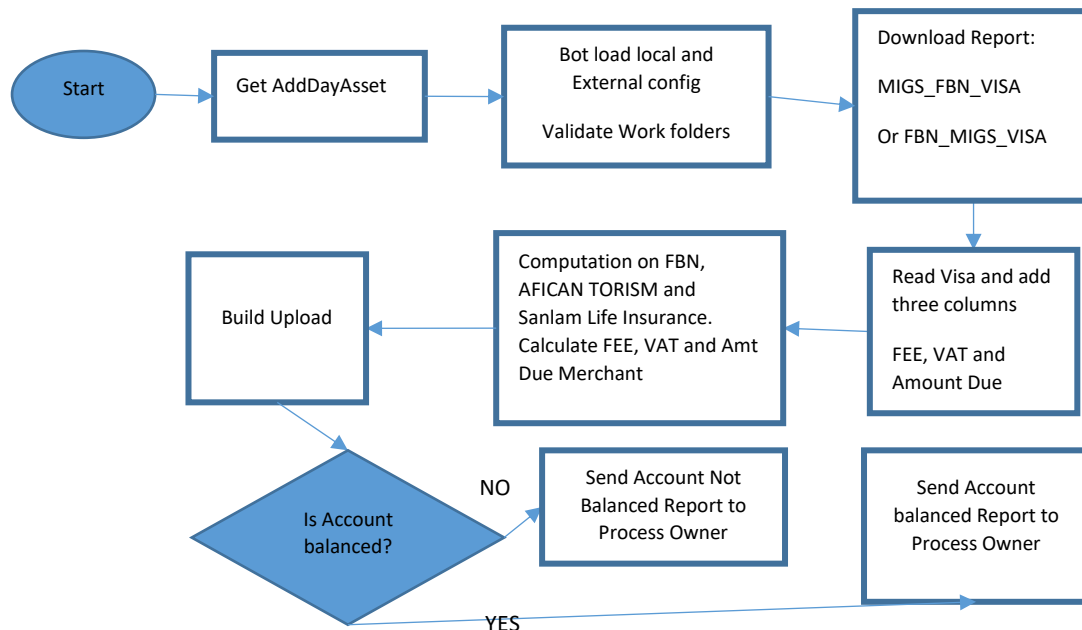
The following artefacts should be read in conjunction with this document.

Title	Version	Location
Process Design Document (PDD) – Migs Visa Settlement	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
 - Try Catch FTP Credentials: Load UPSL FTP Credentials
 - CheckReportPath workflow: Check if the file to be download already exist
 - Sequence Download: Logic of Downloading report
 - If Download File exist
 - Downloading is skip else:
 - Else: 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
- Bot run MIGS: Computation workflow: Out Argument: totalMerchant
 - Multiple Assign: all needed variables
 - Read Work Log: workLogDT
 - Filter Data Table: only keep "ORIGINALCCY" Column = "566"
 - Assign workLogRowCount; To check if they are transactions
 - If there is transaction Bot continue else throw exception of No Jobs
 - Assign distinctDT; workLogDT.AsEnumerable().
 - GroupBy(x=>x.Field<string>("MERCHANTTITLE")).Select(g=>g.First()).CopyToDataTable() ()): only return the one number of unique transactions
 - Assign out_totalMerchant:

Convert.ToDecimal(workLogDT.AsEnumerable().Sum(row =>

row.Field<double>("ORIGINALAMT").ToString Reurnt the Total

Merchant Amount
 - For Each on the distinctDT
 - Assign currentMerchant:

CurrentMerchantRow["MERCHANTTITLE"].ToString()
 - Filter Databale: Filter the Merchant to work on from workLogDT using currentMerchant as filter criteria
 - Write Current Merchant to WorkLog: write all the transaction for one merchant in a new sheet
 - Read Each Current Merchant (the new sheet created): runMerchantDT
 - For Each Row in runMerchantDT
 - Get Merchant Details Sequence: Read the external config to get details of the current merchant: "merchantDataRow"
 - Assign merchantRate: get rate of the current merchant
 - Assign FEE:

Convert.ToDecimal(EachMerchant["ORIGINALAMT"]) * merchantRate
 - If merchant is AFRICAN TOURISM CORPOR

Check if the fee is more 2000 then

Fee is set to 2000
 - Assign AMOUNT DUE MERCHANT:

Convert.ToDecimal(EachMerchant["ORIGINALAMT"]) -

Convert.ToDecimal(EachMerchant["FEE"])
 - Assign Vat: $(7.5/107.5) *$

Convert.ToDouble(EachMerchant["FEE"])
 - Write Current Merchant to WorkLog: To save all the calculations back to the runMerchantDT

- Build Upload File
 - Multiple Assign: all needed variables
 - Read Work Log: DT
 - Filter Data Table: only keep "ORIGINALCCY" Column = "566"
 - Assign distinctDT; DT.AsEnumerable().
 - GroupBy(x=>x.Field<string>("MERCHANTTITLE")).Select(g=>g.First()).CopyToDataTable() ()))) : only return the one number of unique transactions
 - Build Upload File Data Table: UploadFileDT
 - Update the first Row
 - GetMerchantDetails workflow: get details for "FBN INSURANCE LIMITED"
 - Multiple Assign: MIGSAcctNo, reportDate,, Narration, Amount, transType.
 - Add Data Row: new row is added to the UploadFileDT with details gotten from multiple Assign
 - For Each on the distinctDT
 - Assign merchantName: CurrentMerchant["MERCHANTTITLE"].ToString()
 - Switch merchantName
 - Case AFRICAN TOURISM CORPOR : If Current Merchant is: AFRICAN TOURISM CORPOR
 - Read AFRICAN TOURISM Sheet: resultDT
 - Get All Required Value
 - GetMerchantDetails
 - Assign counter = 0
 - While totalLoop > 0
 - Switch 0 - 5
 - Case 0: -----
 - Case 1: Update the all-merchant Transaction
 - For each row in resultDT
 - Update the UploadFileDT Row
 - Case 2: Update Fee
 - Update the UploadFileDT Row
 - Case 3: Update the Vat
 - Update the UploadFileDT Row
 - Case 4: Update isStampDuty
 - If isStampDuty is True
 - Update the UploadFileDT Row
 - Case FBN INSURANCE LIMITED: If Current Merchant is FBN INSURANCE LIMITED
 - Read FBN INSURANCE LIMITEDSheet: resultDT
 - Get All Required Value
 - GetMerchantDetails
 - Assign counter = 0
 - While totalLoop > 0
 - Switch 1 - 4
 - Case 1: Update the all-merchant Transaction
 - For each row in resultDT

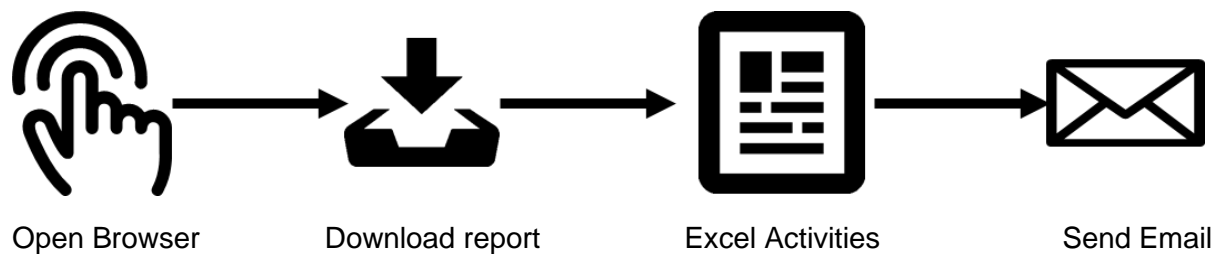
- Update the UploadFileDT Row
 - Case 2: Update Fee
 - Update the UploadFileDT Row
 - Case 3: Update the Vat
 - Update the UploadFileDT Row
 - Case 4: Update isStampDuty
 - If isStampDuty is True
 - Update the UploadFileDT Row
 - Case SANLAM LIFE INSURANCE: If Current Merchant is SANLAM LIFE INSURANCE
 - Read SANLAM LIFE INSURANCE Sheet: resultDT
 - Get All Required Value
 - GetMerchantDetails
 - Assign counter = 0
 - While totalLoop > 0
 - Switch 0 - 5
 - Case 0: -----
 - Case 1: Update the all-merchant Transaction
 - For each row in resultDT
 - Update the UploadFileDT Row
 - Case 2: Update Fee
 - Update the UploadFileDT Row
 - Case 3: Update the Vat
 - Update the UploadFileDT Row
 - Case 4: Update isStampDuty
 - If isStampDuty is True
 - Update the UploadFileDT Row
 - END The Upload File
 - Multiple Assign: MIGSAcctNo, reportDate,, Narration, Amount, transType.
 - Add Data Row: new row is added to the UploadFileDT with details gotten from multiple Assign
 - Output Data Table
 - 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") == "D").Sum(z => z.Field<decimal>("Amount").ToString()))
 - If Account is Balanced: debit - credit == 0
 - If is True : Assign out_IsBalanced = True
 - Else Assign out_IsBalanced = False
 - Copy File: Upload 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
C1	Process	Open Application	Opens Paydirect portal to download the required reports
C2	Process	Excel Activities	<ul style="list-style-type: none">• Read Excel document into datatable• Add rows• Sort document by columns• Perform computations i.e Add, Subtract, Multiply
C3	Process	Send Email	<ul style="list-style-type: none">• Attach excel file to email• Send email

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
1	Change in report Format <ul style="list-style-type: none">- Change in number of columns.	<ul style="list-style-type: none">• Send email alert to process owner
2	Inability to find report <ul style="list-style-type: none">- The bot is unable to download report due to change in naming convention of the required report.	<ul style="list-style-type: none">• Send email alert to process owner
2	Inability to find report <ul style="list-style-type: none">- The bot is unable to download report due to change in naming convention of the required report.	<ul style="list-style-type: none">• Send email alert to process owner
		<ul style="list-style-type: none">•
		<ul style="list-style-type: none">•

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
VW	PAC, Robot, Bot	Virtual Worker
HW		Human Worker
PDD		Process Design Document
SDD		Solution Design Document
VM	VM	Virtual Machine
VDI	VDI	Virtual Desktop Interface
RPA	RPA	Robotic Process Automation
SSO		Single Sign-On

ii. Attachments

The following attachments relate to this document.

Attachment	Description