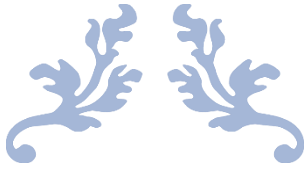

FUNCTIONAL AND TECHNICAL DOCUMENT

PAID IN FULL & SETTLED IN FULL AUTOMATION SCRIPT

TEAM AUTOMATION
TECH-OPS

Purpose:

The purpose of this functional and technical document is to provide documentation on the requirements of Automating the paid in full and settled in full job to reduce the manual process and increase the accuracy and make it timed in a systematic manner.

Scope:

Due to the time consumption and the repetition of the same process while doing this for multiple loans at a time, better we have to automate this process, to do this for large number of loans in a single stretch within time and in efficiently.

Process:

Paid in Full (PIF) & Settled in Full (SIF):

- Paid in Full means the customer paid off the entire amount due on a loan.
- Settled in full means the customer negotiated with the lender to pay off some amount less than the actual payoff amount.
- So, we must update this in all the relevant fields in our system.
- within 10days, we must send a communication/confirmation E-mail to the customer stating about the PIF/SIF with the attachment of watermarked loan agreement as 'PAID' for PIF and 'CLOSED' for SIF.
- We should store these watermarked documents in the S3 bucket for audit and other purpose and we are storing these details in a separate table called *reports.PIF_SIF_Email* as well.

Functional Requirement:

- Create a separate table (PIF_SIF_Mail) in reports schema and keep update the table daily basis to track all the loans, for those who have completed the PIF/SIF process after Paid/settled in full the loan.
- Split the loans into three category on each day such as "Paid_in_full","Paid_in_full_no_refund" and "Settled_in_full".
- Download the E-signed loan agreement of each customer from S3 bucket in adfcontracts folder.

- Create a watermark into the contract as “PAID” for PIF and “CANCELED” for SIF loan.
- Upload the document in the Exact target sftp location to attach it with the communication E-mail which we will trigger to the customer.
- Fetch the required details from DB to trigger the Email with the watermarked loan contract.
- Do the Backend changes in the relevant fields in both CL and decision database.
- Upload the watermarked document into S3 bucket PIF_SIF folder for the future reference and audit purposes.
- Create a task in salesforce and appdesk for future reference as well.

Technical functionality:

We have created a separate function for each step explained above as a functional requirement.

Function-1 (initial_insert):

In this function we will make the reports.PIF_SIF_Email table up to date with all the closed loans. So that we can use the specified values from this table for the whole process.

- Fetching the customer details who Paid in full/settled in full their loans by applying the below conditions.
- The loan Status should be 'Closed - Obligations met'.
- Closure email date and Closure email field should be null.
- Loan closed date should be less than 10 days from the current date or Loan closed date is equal to 10 days from the current date.

Function-2 (Eligible_loans):

In this function, by joining with the table which we have updated in insert_query function. We will fetch the below three categories of loans by applying the corresponding conditions.

The output values of this function will be used in the entire script to call the function one by one.

1. Paid_In_Full_NoRefund

    - Closure email date is null and Closure Email field is null
    - Sub Status is not equal to 'Settlement'.
    - Excess amount is lesser than 1 dollar.

2. Paid_In_Full

    - Closure Email Date is null, and Closure Email is null.
    - Refund Type is not equal to 'RCC'.
    - Refund Date is not null.
    - Sub Status is not equal to 'Settlement'.
    - Refund Amount is not equal to 'None'.
    - Excess amount > 0.99.
    - Excess amount is equal to Refund Amount.

3. Settled_In_Full

    - Closure Email Date should be 'None'.
    - Closure Email Type should be 'None'.
    - Sub Status is equal to 'Settlement.

- If the result is empty, then code will get terminate from the whole script at this point rather than executing all the steps and finding out there is no data. If it returns any data, then it will fetch the below list of fields from the DB [*LoanID, Contact ID, First Name, Last Name, Location, TemplateID, Update_field, PIF_SIF*].

- At last we will update the pick date as current date in the reports.PIF_SIF table for loans which is picked by this function. To make sure these loans are picked for this process.

Function-3 (From_S3):

In this function we will download the E-signed loan agreements of each customers which we have stored in the S3.

- We are passing the list of loanID's which we retrieved from the ***Eligible_loan*** function to the DB (Decision and RPOS) to get the corresponding loanID and the document link.
- Connecting the S3 bucket to download the loan agreement of the respective loanid's which we got from the above step.
- We will store all the documents in the respective path.

Function-4 (Watermark):

In this function we will make the agreements as watermarked and will upload that in Exact target's ftp location.

- Concatenate the script directory with the loanID and assigning this as a variable 'loc_name'.
- The watermark image was assigned as 'imgname'.
- By using the pdf_file_reader function, the code will open the variable 'loc_name' (downloaded loan agreement) and imgname (the pdf format of watermark).
- Assigning the variable as page_count to get the page count of loan agreement document.
- Using the for loop to merge the watermark into the loan agreement in all pages with the help of page count variable.
- Store the watermarked loan agreement in the script directory with the name pattern as loan_number.pdf.
- By using the 'getsize()' function the code will get the size of actual loan agreement and the watermarked loan agreement. For Example, if the actual loan agreement size is 100kb and size of the watermark is 10kb then the size of the outcome should be 110kb.
- If the size of watermarked loan agreement is greater than the actual agreement then we will pass the list of watermarked loan agreements to *Upload_to_ET* function.

- If the size is not greater than the actual agreement, then the code will exclude the loan agreement by printing this as exception message.

Function-5 (Upload_to_ET):

In this function we will upload the list of watermarked loan agreements to the ET.

- If the watermarked agreement from the size is greater than the actual loan agreement, then the code will store the list of documents into the sftp location of Exact target.

Function-6 (mail_trigger):

This function is used to trigger the Email for the three categories of customers with the watermarked loan agreement.

- We will fetch the required details from the DB for the payload to trigger the communication email to the customer.

- Based on the three different categories of customers (paid in full, paid in full refund and settled in full) we will trigger the communication Email by calling the DE mail trigger API.

- If the triggered response is success (200) then it will update the mail sent response field in the reports.PIF_SIF_Mail table as 'true'. If the response is not success, then it will update the mail sent response field as 'false'.

As a final step of this function, it will update the below fields in the reports.PIF_SIF_Mail table.

  - mail sent = 1, this indicates we have sent the E-mail to the customer.
  - Mail sent response = true or false depends on the triggered email response which is already explained above the 3rd point.
  - Acc_No field, the account number of the customer and Email_type, the type of Email we have triggered to the customer.

Function-7 (To_S3):

In this function we will upload the watermarked loan agreement in S3, which we have sent to the customer.

- We were appending all the watermarked loan agreement as a python list in the **pif_sif** function.

- So, this function we will upload all the watermarked loan agreement in the S3 bucket.
  - Folder name = sif.pif.contract
  - Format of file Name = LoanID.pdf

Function-8 (bulkapi):

In this function we will update the below three fields in the Loan page and also will create a detailed Task stating all this process in salesforce.

- First, we will keep the below field in a config file.
  - Closure_Email_Date
  - Closure_Email_Type
  - update_date

- We will fetch the three required details from the DB and by using the Zip function, we will create this combination as a python dictionary (fieldname: value).

- After that, by calling the bulk API we will do this update in the loan page (*Loan_loan_accounts_c*) of Salesforce and we will update the same in our *reports.PIF_SIF_Email* table as well.

For adding task in SF, We will fetch the required fields from database and we will pass the below SF fields in the bulk API function.

- subject = "PIF/SIF : + current date".
- Status = 'Completed'.
- Priority = 'Normal'.
- Ownid = '005j000000DbwqnAAB'.
- Description = "PIF/SIF Mail sent dated on : + current date + templateID" .
- Whoid = ‘ContactID’.
- ActivityDate = ‘current date’.

As a final step of this function, we will update the ‘Task_Response’ field in the reports.PIF_SIF_Email as ‘1’ for those who we made this successful task creation

Function-9 (App_Desk_Update):

This function is used to create a notes in Appdesk regarding the PIF/SIF process.

- In this function we will fetch all the records which we have inserted in the reports.PIF_SIF_Mail for the current date.
- We have separate function for this process named app_desk.
- After that in a for loop we will pass the loan Number and doctype into the function to add notes in it.

**********THANK YOU **********