****

**Application Flow Design Document**

### Project: STP – Straight through Process.

### Document Overview:

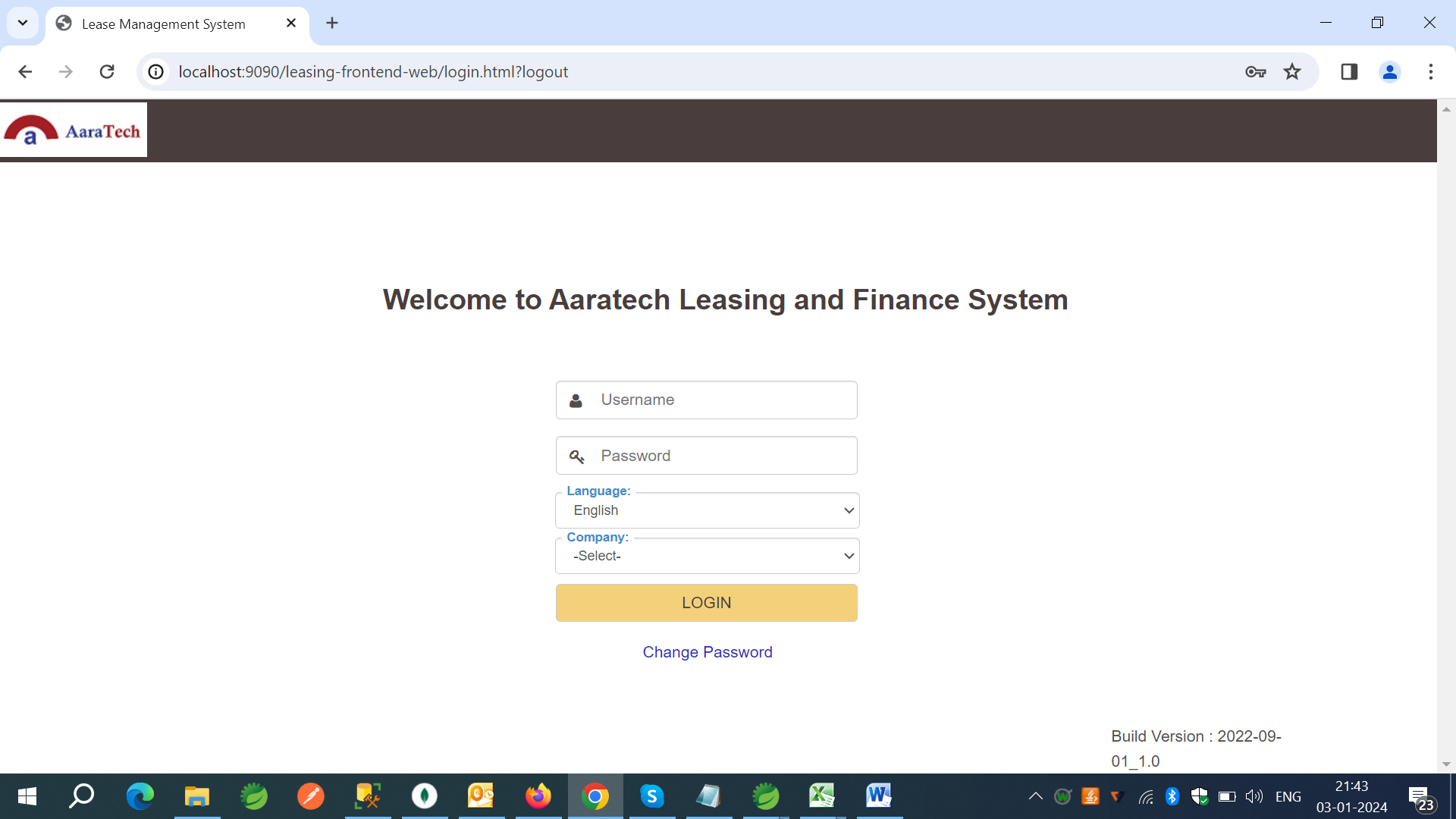
This document provides a comprehensive layout for the application flow design of the Straight through Process website. It includes a step-by-step representation of the user journey through various screens, accompanied by relevant screen shots.

### Introduction:

Straight through Processing (STP) streamlines business operations by automating tasks, optimizing workflows, and facilitating seamless data transfer. It relies on standardized APIs, enabling cohesive communication between diverse systems. The compilation of processes within an STP application enhances efficiency, reduces redundancy, and fosters agility. STP's adaptability allows businesses to respond swiftly to changing needs, all while minimizing disruptions. This approach introduces standardization without overhauling existing processes, ultimately reducing manual intervention and promoting a more agile business environment.

### High-Level Flowchart:

**Login Screen**

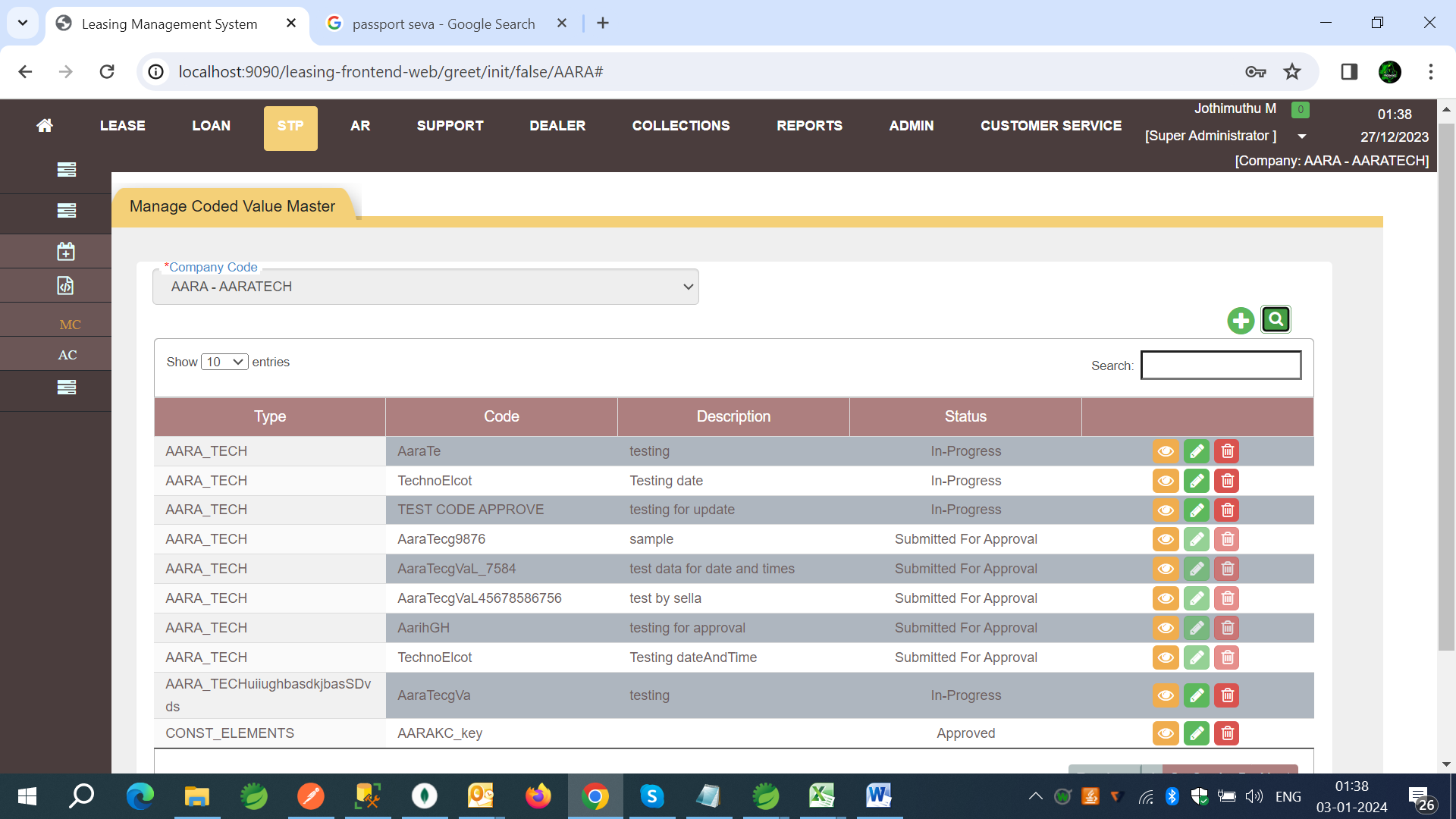


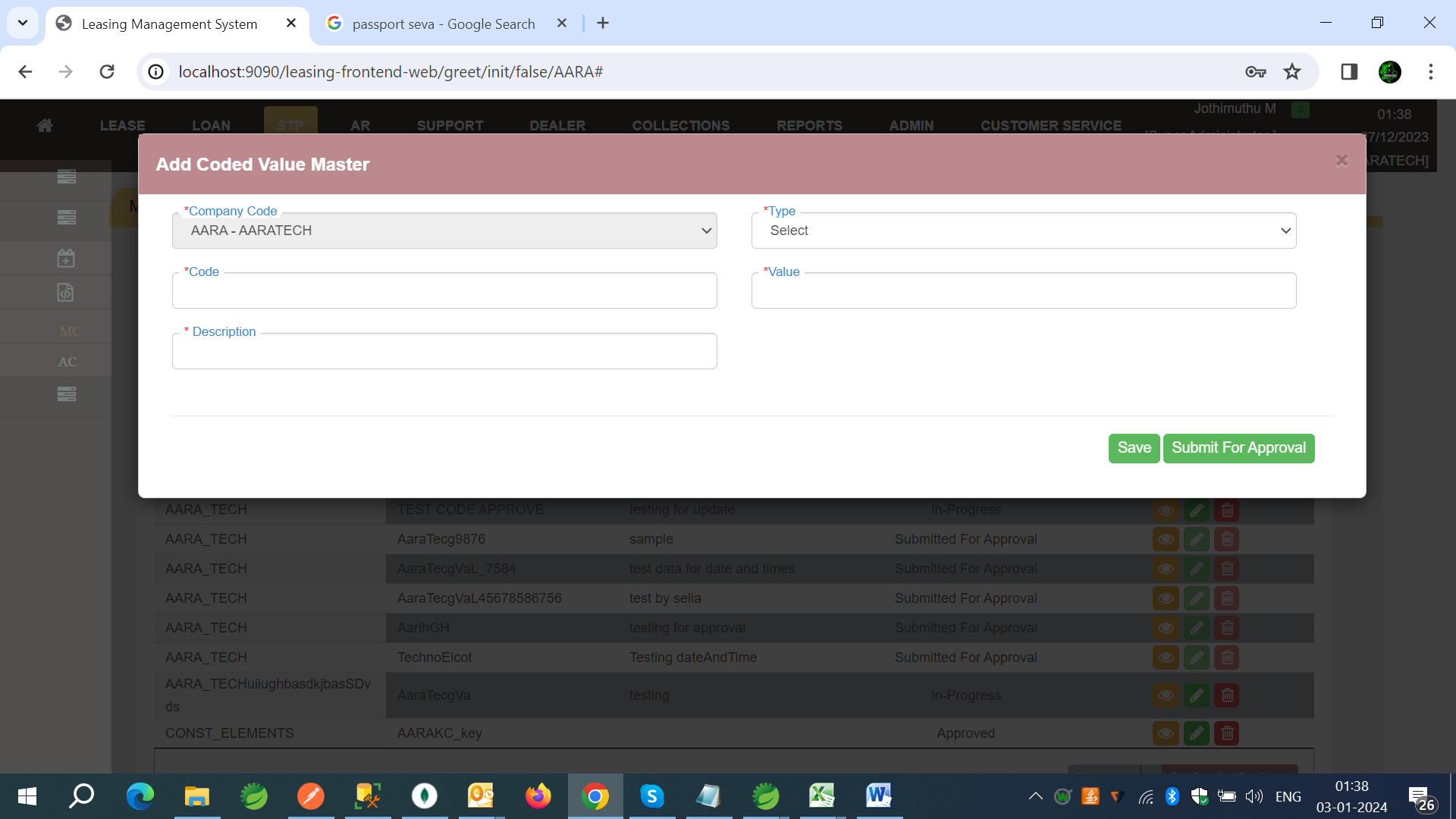
****

**STP Coded Value master:**

STP Coded Value Master screen can set-up below categories for API Mapping Data.

1. Date Patterns.
2. Custom Patterns.
3. Encrypted Data.
4. Constant Values.

**Search Screen –** Search the records based on creteria.

**Form Screen –** Create records using form screens.

Type - Choosing a type of constants or categories we want create for STP.

Code - Code is used define the key of coded value.

Value - Value is value for the entered code and it will Encrypted for Constant Elements Category.

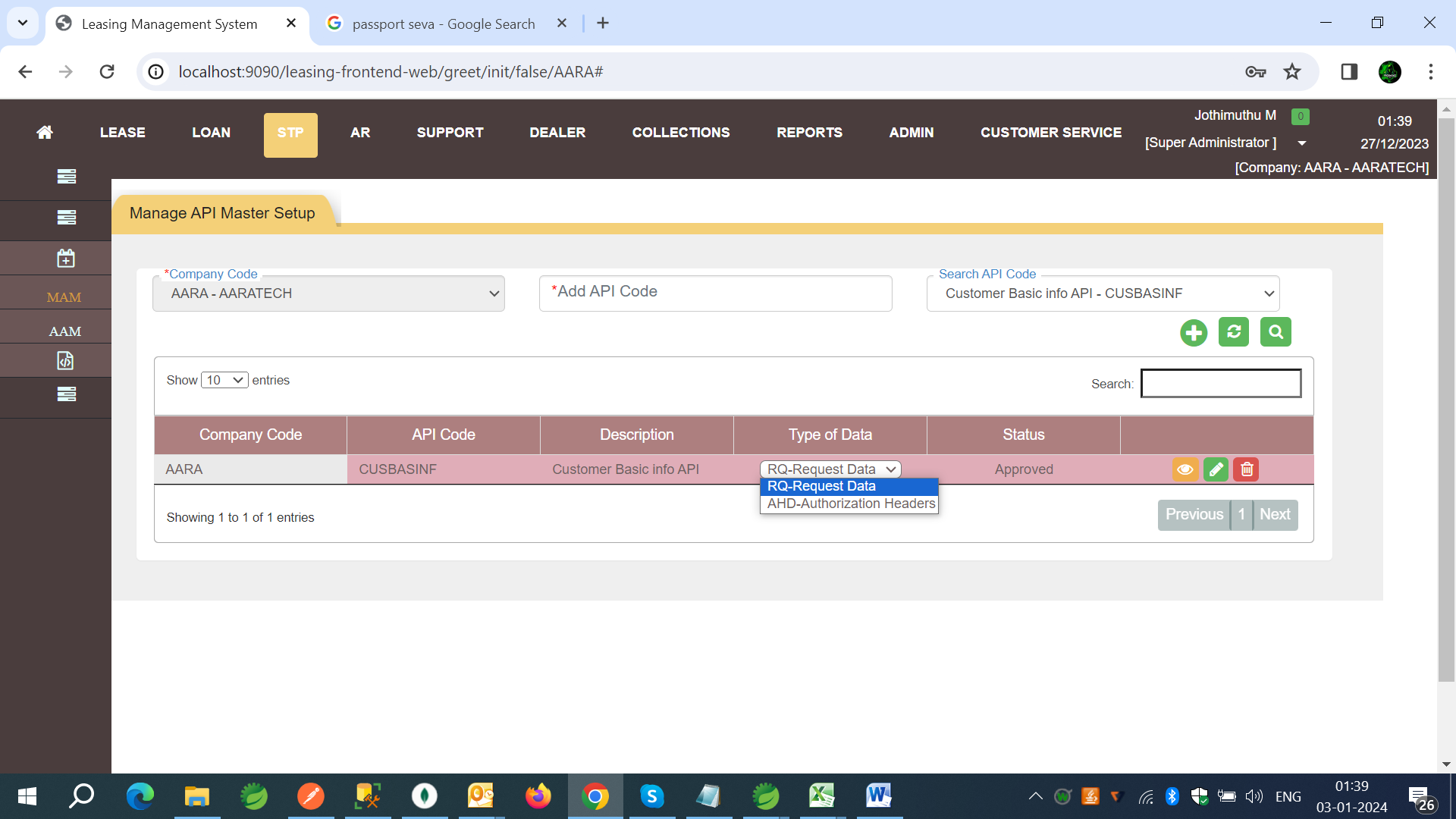
Description – Description for the code and value created.

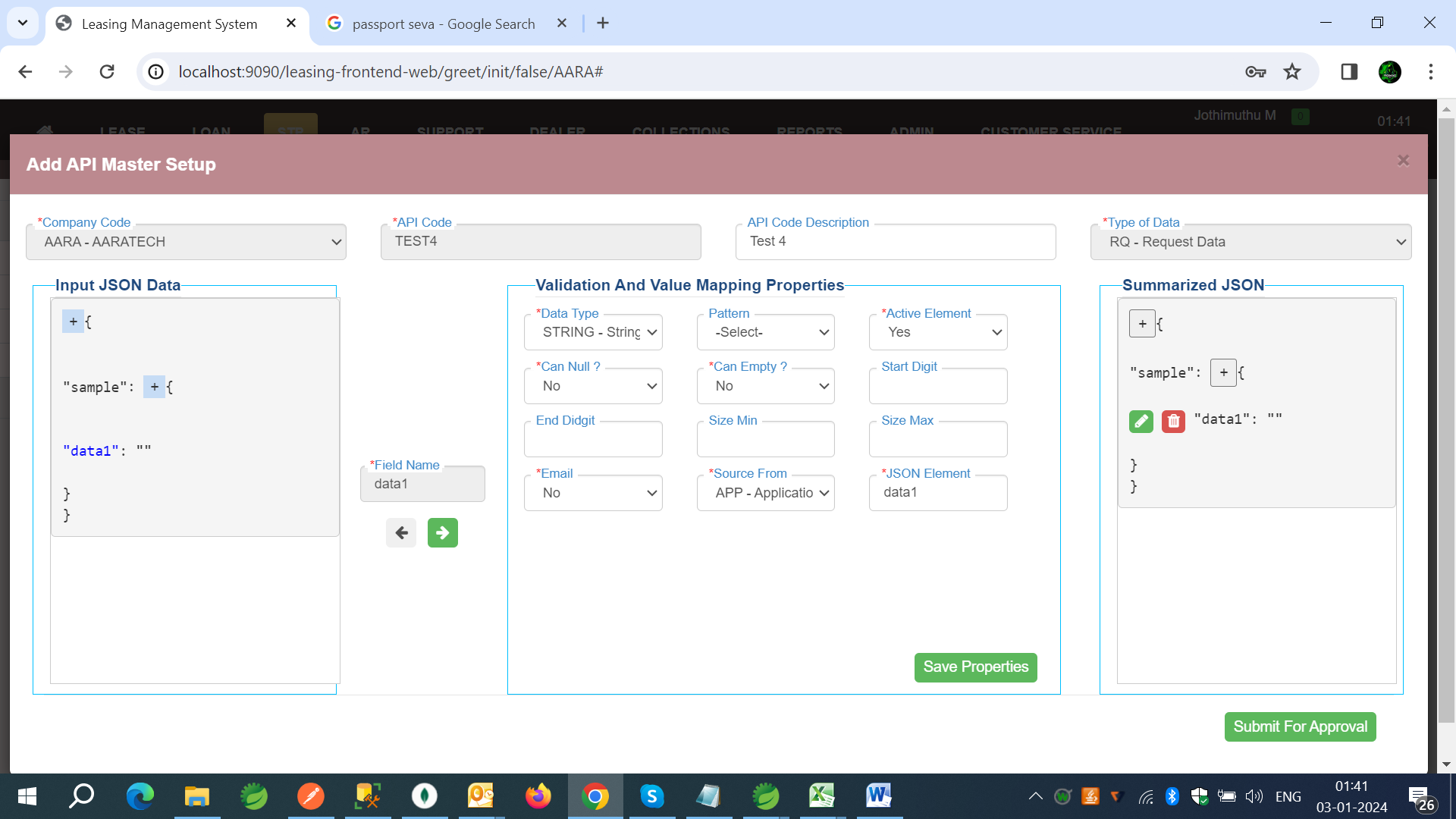
****

**STP API Master:**

STP API Master Screen set-up the API’s Request, Response, Headers, Auth Headers as JSON with mapping of values including validations. Response Objects can be decided to store in STP Engine Database or not.

API Master is based on API Code at Company Level.

**Search Screen –** Search the records based on creteria.

**Form Screen –** Create records using form screens.

API Code - Custom code can be entered to identify the API Code.

API Code Description - Description of API Code.

Type Of Data – Identify what Type of API (RQ-Request, HD-Headers, AHD-Authorization Headers, and RE-Response).

Input JSON Data – The window can the any JSON value (without element values)

****

After process by click on green button can select a Blue Elements from JSON to enter the validation and value mapping fields in Validation and value mapping properties tab.

**Validation and Value Mapping properties:**

Date Type - Data type of Element value.

Pattern - Custom Patterns can choose from Data list from Coded Value Master.

Active Element - Choose Yes or No to say can active or In-active. If in-active it will persist but not on real time API call.

Can Null? - To check the value of the element can be null or not.

Can Empty? - To check the value of the element can be Empty or not.

Start Digit - To check the value of the element is start with given Integer.

End Digit - To check the value of the element is End with given Integer.

Size Max - To check the value of the element is size max with given Integer.

Size Min - To check the value of the element is size min with given Integer.

Email - Enable value is email or not. Custom Pattern for Email is checked by System.

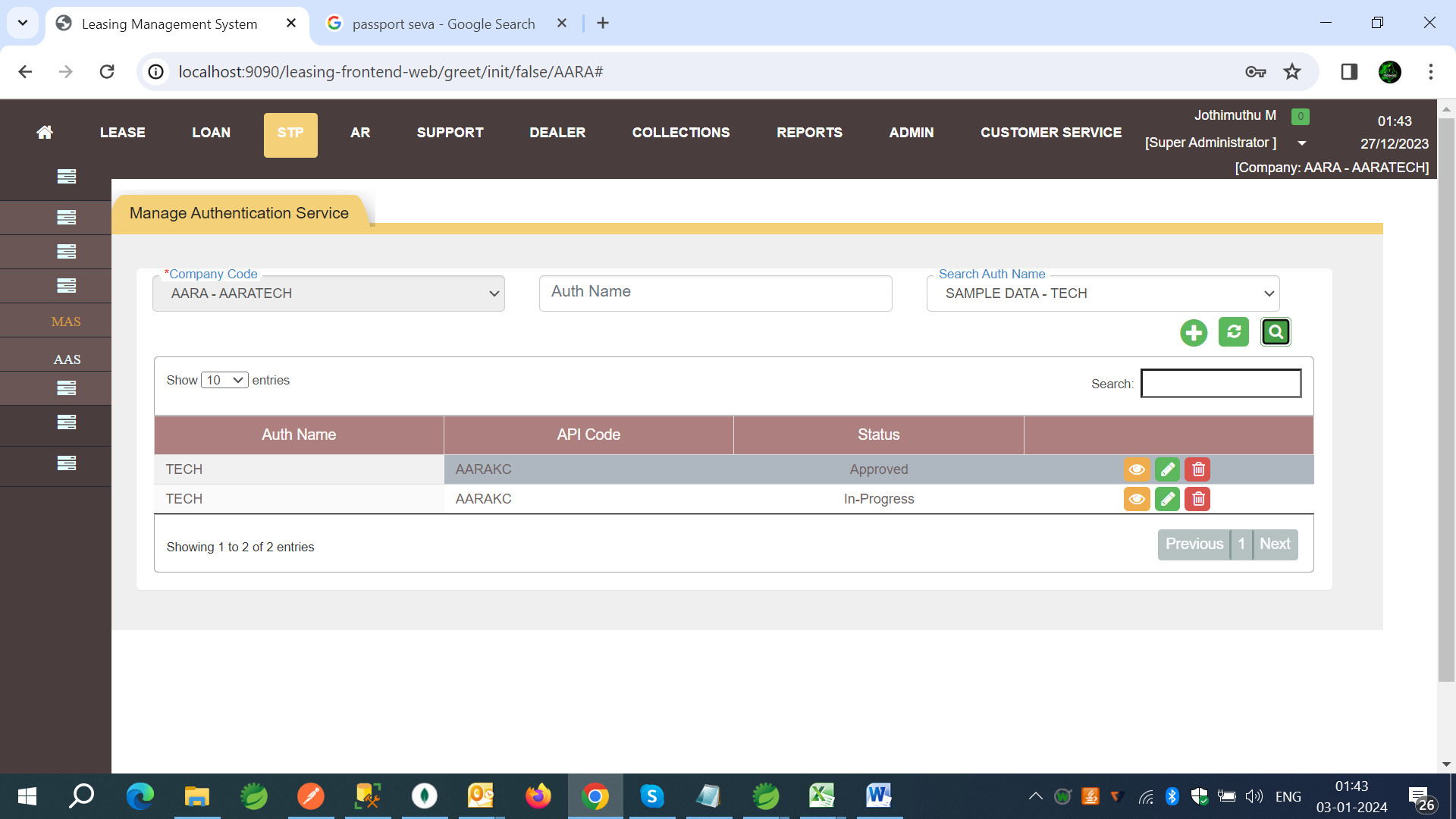
Source From - Can choose from APP-Application Data (which is client Data), CUM-Cumulative Data (Which already stored data in System Database by Request ID), CON-Constants from Coded Value master.

JSON Element - Entered Element name will be searched based on Source from Data’s.

Summarized JSON - Will show the Output of Generated JSON with on click of every Element can view update and Delete.

**STP Authentication Service:**

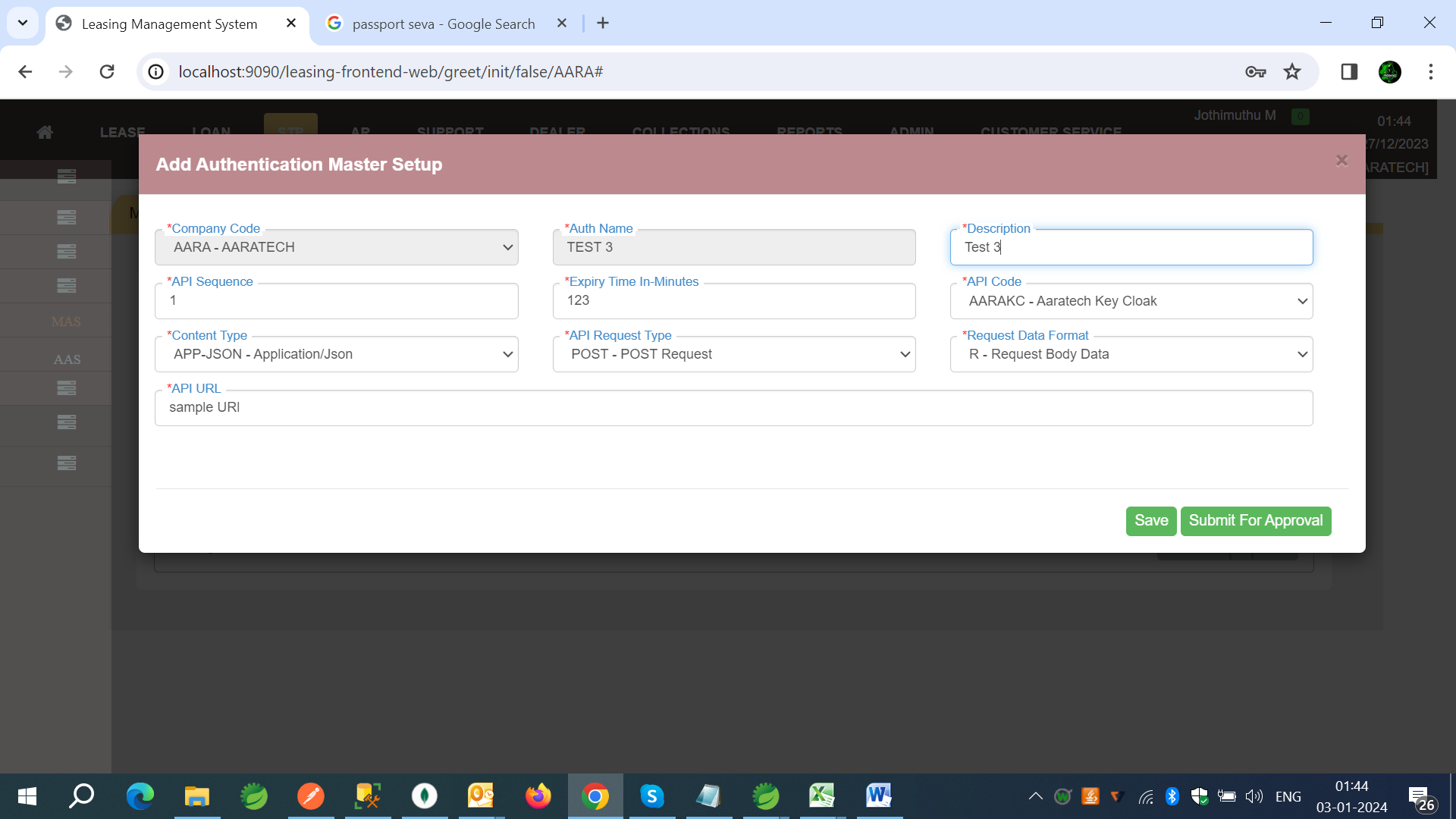
STP Authentication Service supports to set-up widely used Security API’s such as Key Cloak API, JWT or any in a parallel or sequencially based on a set-up. And Can used before the API’s called by STP if need.

**Search Screen –** Search the records based on creteria.

****

**Form Screen –** Create records using form screens.

The API Master - API Code used as Request, Response, Headers and Auth Headers before calling API.



Auth Name - Custom Auth code entered on add Screen.

Description - Description for Authentication Service.

API Sequence - It will take integers in numerical order and compile and run the API’s, If same numbers given system will treat that as a Parallel and run on parallel API’s.

Expiry Time In-Minutes - Used to re-using the Tokens by System from Stored Data.

API Code - Choose API Code which is listed from API Master Data’s.

Content Type - Choose Content Type which supports the Client API’s.

API Request Type - POST - Post Request, GET – Get Request.

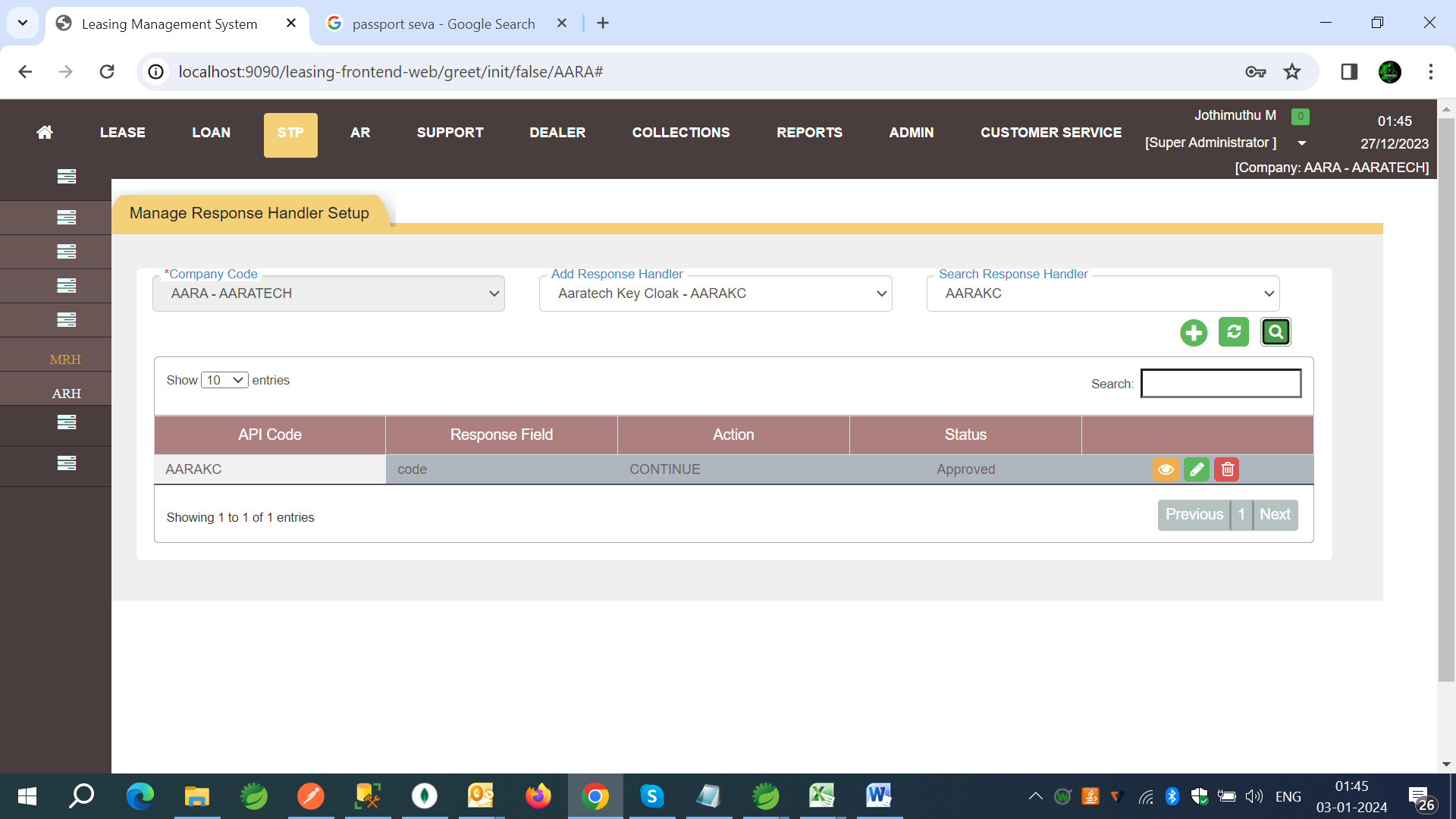
Request Data Format - R - Request Body, P - Path Variable, Q – Query Param.

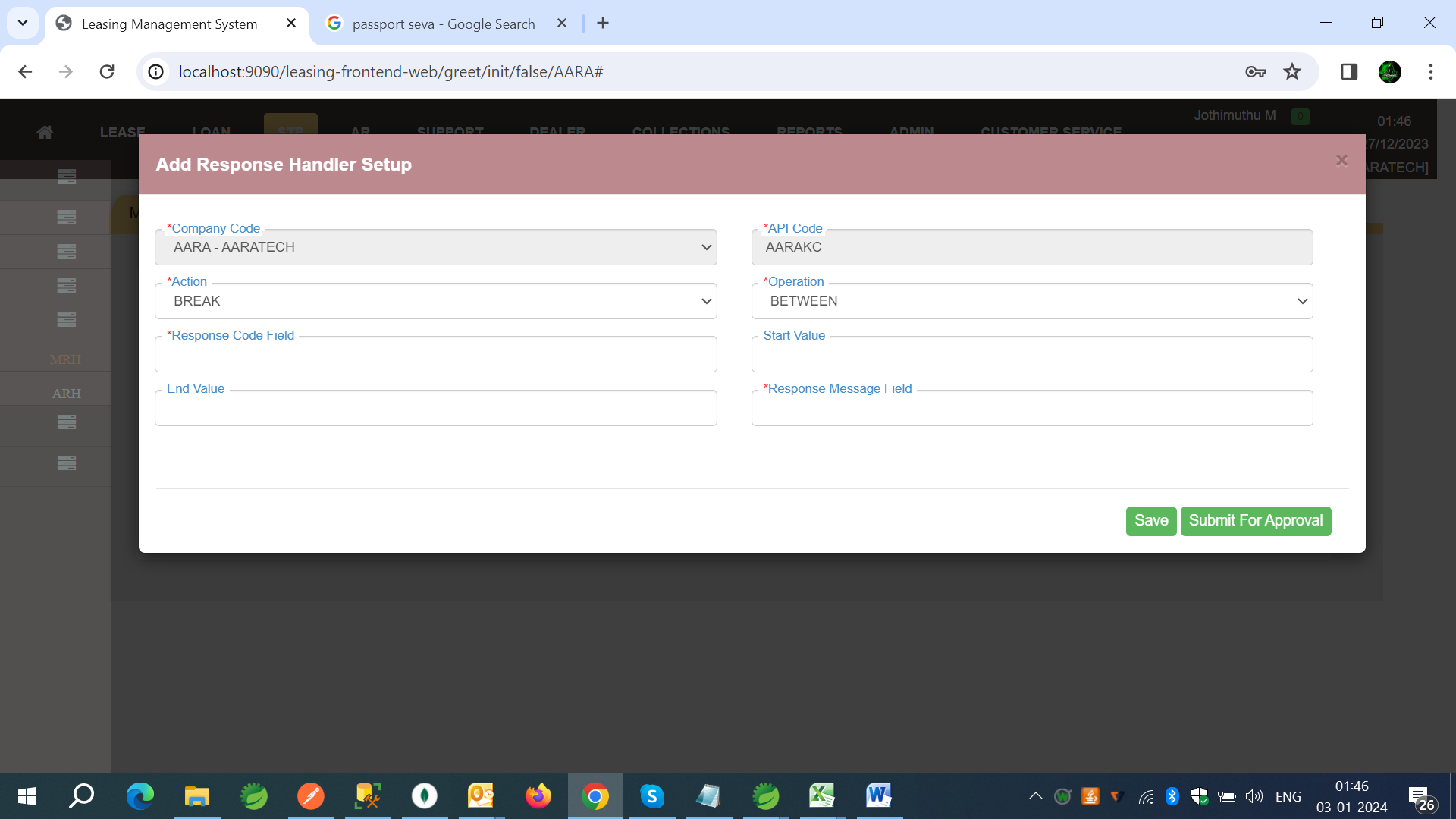
API URL - Full URL of API (System will encrypt for security).

**STP API Response Handler:**

STP Response Handler used to handle the response of API’s that are called by STP. Based on the condition created for API Code can validate the API after getting Response. So that can Process should continue or break can defined.

****

**Search Screen –** Search the records based on creteria.

**Form Screen –** Create records using form screens.

API Code - Choose API Code which is listed from API Master Data’s.

Action - Break or continue when the condition Satisfies.

Operation - Equal , Not Equal, Between and Not Between will perform based on values.

Response Code Field - Response Code Field from element of API response.

Response Value Field - Response Value for Field from Response Code Field.

Start Value and End Value are enabled when Operation Between is choose.

Response Message Field – Response Message element field from API Response.

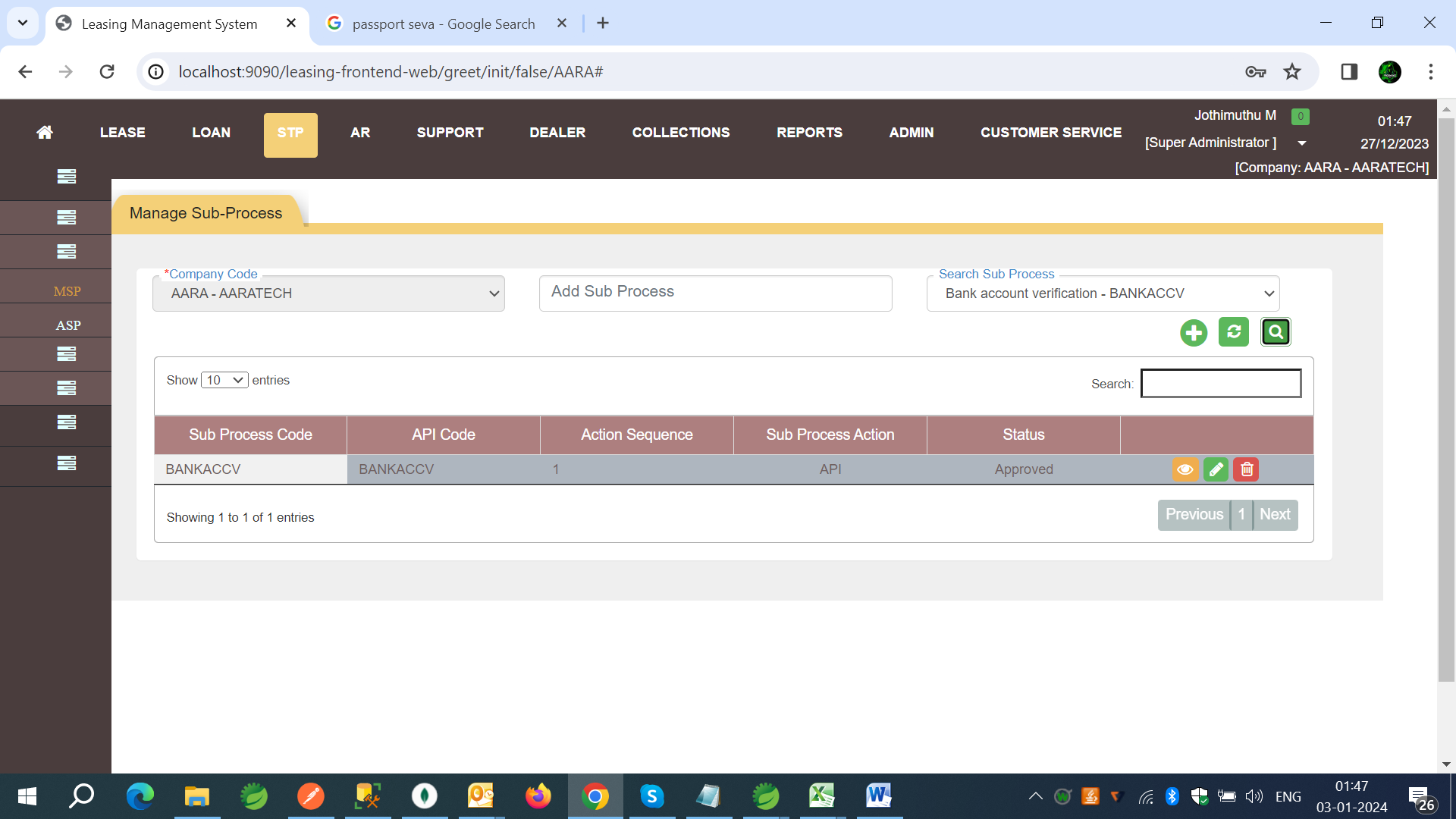
****

**STP Sub Process:**

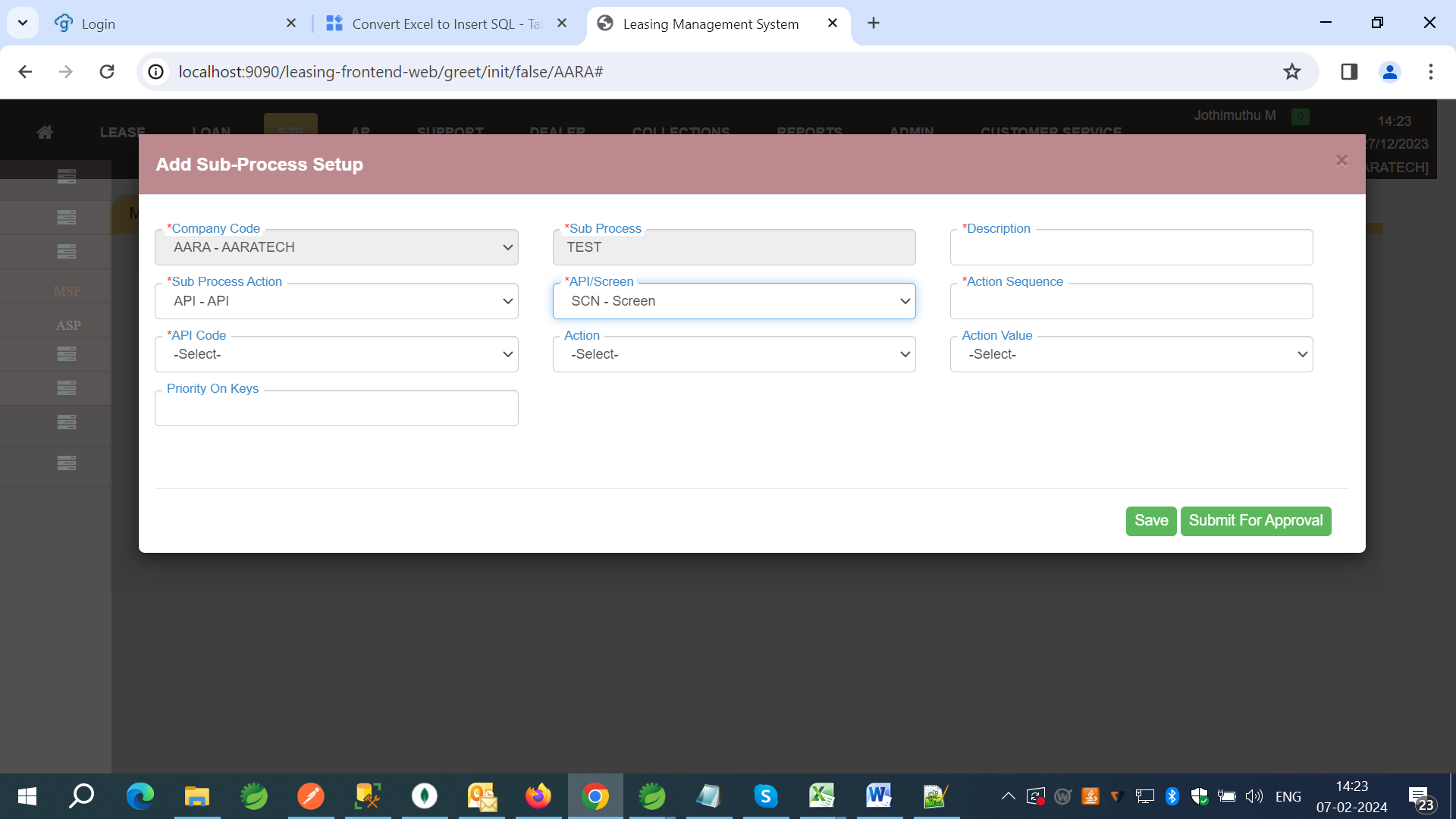
STP Sub Process supports to set-up the individual Process’s and set of API’s as group of process and also one process can mapped to another sub process.

The API related set-up are done here such as URL, Authorization Type, Authentication Service, as well as Screens as Screen ID.

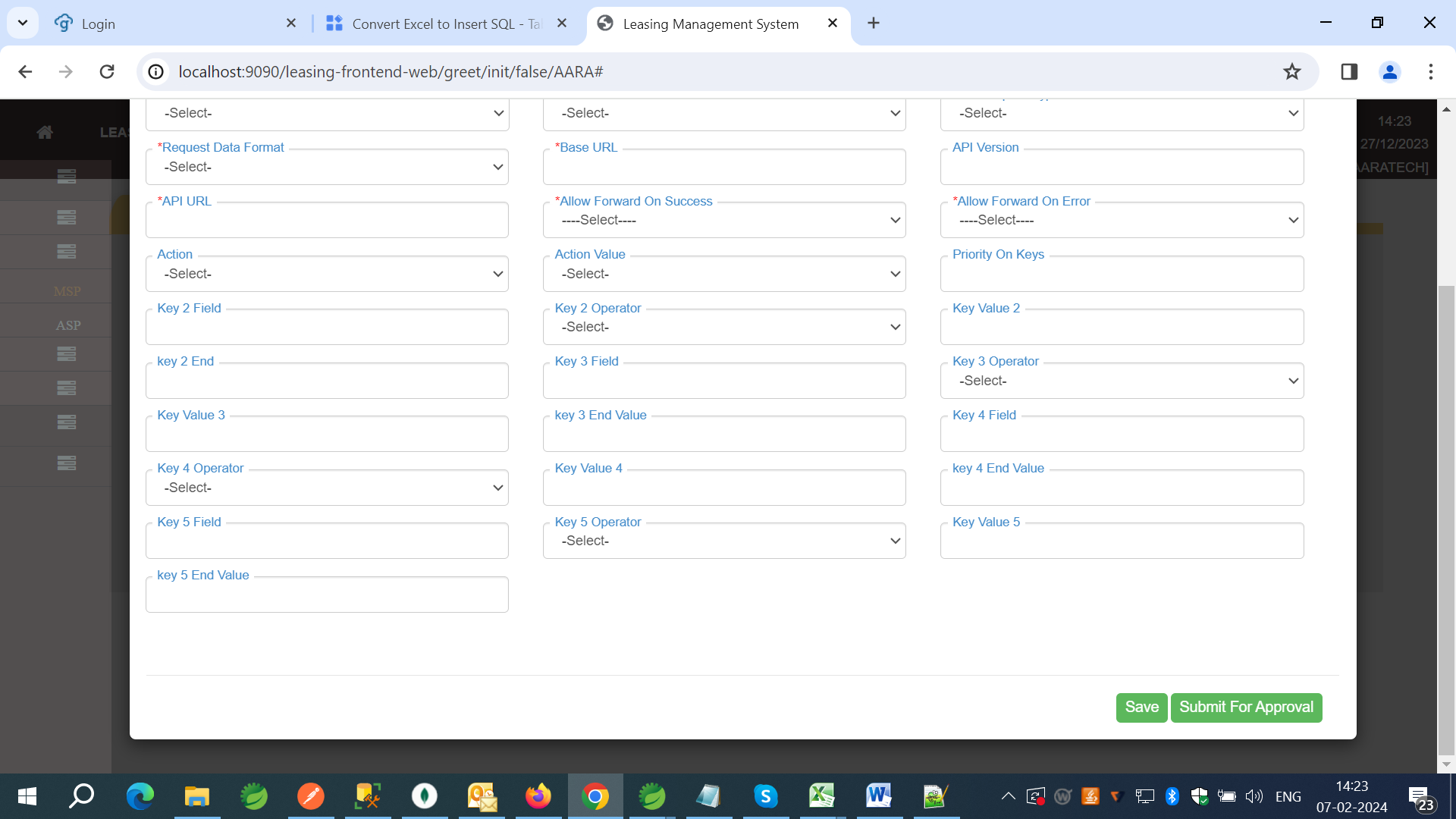
**Note:** Screen ID’s are exposed and expected to handle by client side to handle screens because STP will handle Only API’s

**Search Screen –** Search the records based on creteria.

**Form Screen –** Create Records using form screens.



****



Sub Process - Sub Process code to identify among the other process.

Description - Description for Sub Process.

Sub Process Action – API/PRS – PRS can be choose to create another process with existing process as multiple process.

API/Screen - To choose the Process is API or screen. Based on that Properties will be populated on screen.

API Code - Choose API Code which is listed from API Master Data’s.

API Sequence - It will take integers in numerical order and compile and run the API’s, If same numbers given system will treat that as a Parallel and run on parallel API’s.

Content Type - Choose Content Type which supports the Client API’s.

API Request Type - POST - Post Request, GET – Get Request.

Request Data Format - R - Request Body, P - Path Variable, Q – Query Param.

API URL - End Point of API.(/getData)

Base URL – Base URL of API will encrypted on system for security.(Ex: https:clientApplication.com/).

API Version Can also Be Entered – To maintain as part of Endpoints as concatenated as $VERSION

(Ex: v1/getData will be as /$VERSION/getData).

Auth Name - Choose Auth Name which is listed from Authentication Service Master Data’s.

Allow Forward On Success - Yes or No. while calling Parallel API’s Based on these values system will decide to continue or Stop the Process.

Allow Forward On Error - Yes or No. while calling Parallel API’s Based on these values system will decide to continue or Stop the Process.

****

Allow Forward On Reject - Yes or No. while calling Parallel API’s Based on these values system will decide to continue or Stop the Process.

Allow Forward On Deviation - Yes or No. while calling Parallel API’s Based on these values system will decide to continue or Stop the Process.

Note: As per latest update system is not capable of handling Reject and Deviations.

There are 5 more keys handling in STP Sub process for calling multiple next sub process based on evaluating response checks and actions.

Key field, Key Operation, Key Value’s.

Key field -> Element of respective API Response.

Key Operation -> Operation will performed like Equal, Not Equal, Between and Not Between, Lesser than, Greater than, Greater than or Else than.

Key Value -> Value will be checked against the value of response by field.

By evaluating conditions can choose the next actions.

Action From -> Action from drop down can choose whether the continuation is on Process or AP or Screen. Based on that will displayed the Process codes/API Codes.

Action Value -> will display the Process codes/API Codes/ Screen Process based on Action from values.

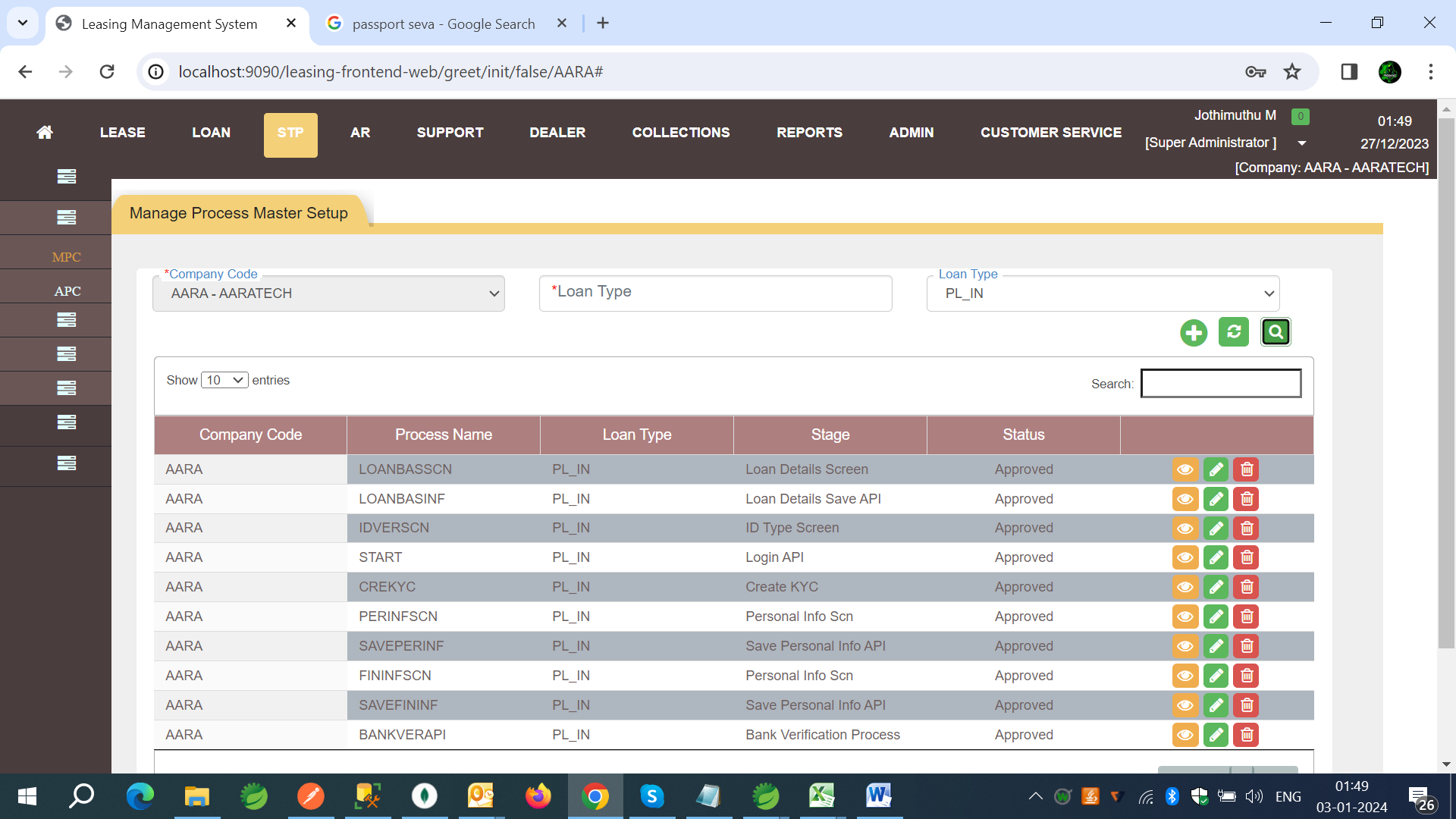
All these keys and Actions are not mandatory, can set-up if need or else will flow based on Action sequence and stopped.

**STP Process Master:**

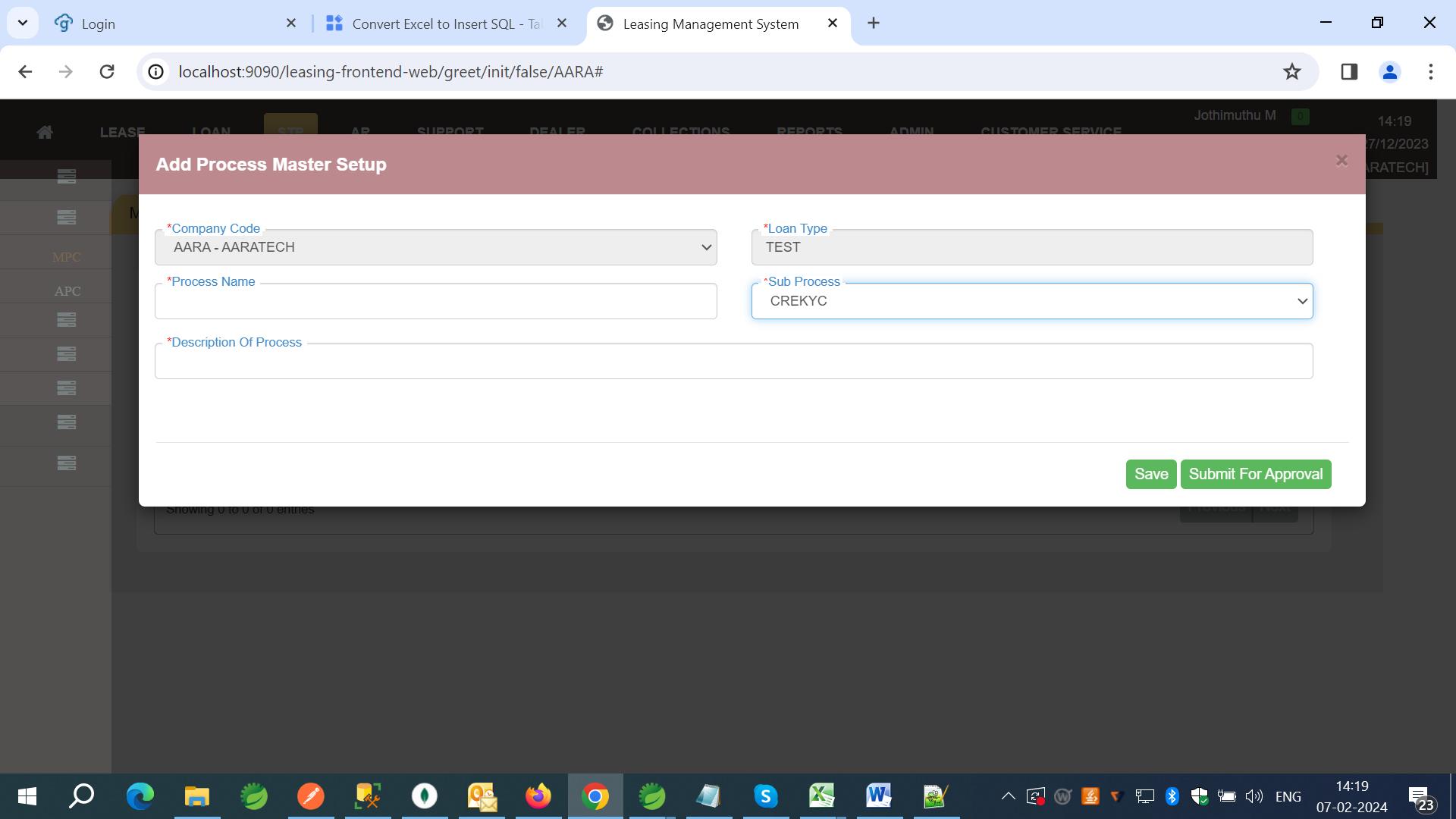
STP Process Master is screen to set-up the Process of Start to End. The Process are manipulated and Continued by combination of 4 key’s (which are key and value and referring to stored response of STP Process.

**Search Screen –** Search the records based on creteria.

Based on the Loan Type that already created can be listed and searched. For new Loan Type Set-up give Brand new Loan Type and Start of Set-up the Process for Your Company Code.



**Form Screen –** Create records using form screens.



Process Name – Process name will be default as START for every new Loan Types. For Existing it will be enterable to create client based process code.

Sub Process – Sub Process can be used for which process should perform and ideally it will compile and run the Sub Process as set-up.

Description – To tell about the Process as individually.

**Refresh Bean:**

Refresh Bean Screen to refresh the Bean Context data of STP Engine through screens.

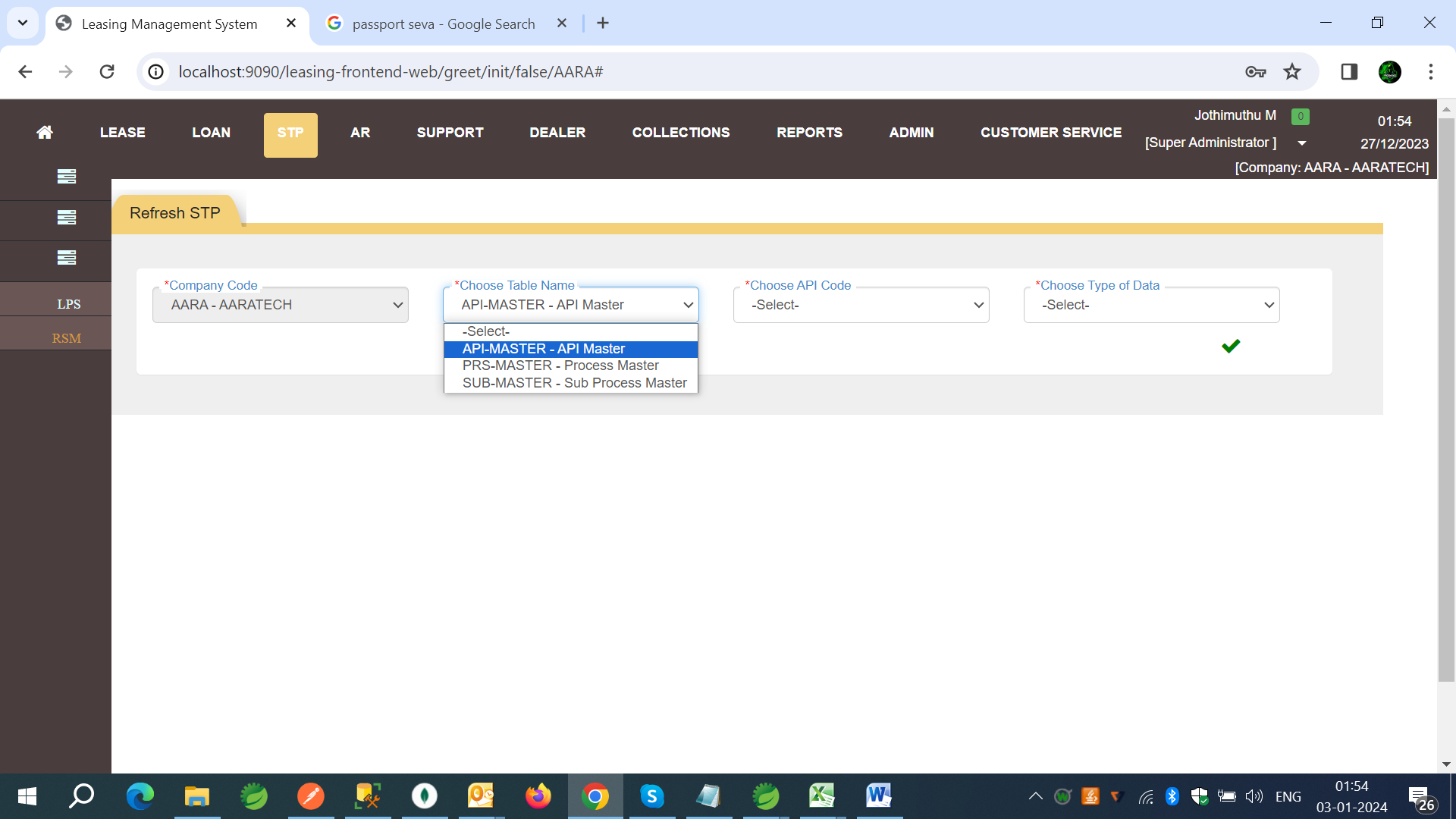
How it’s works?

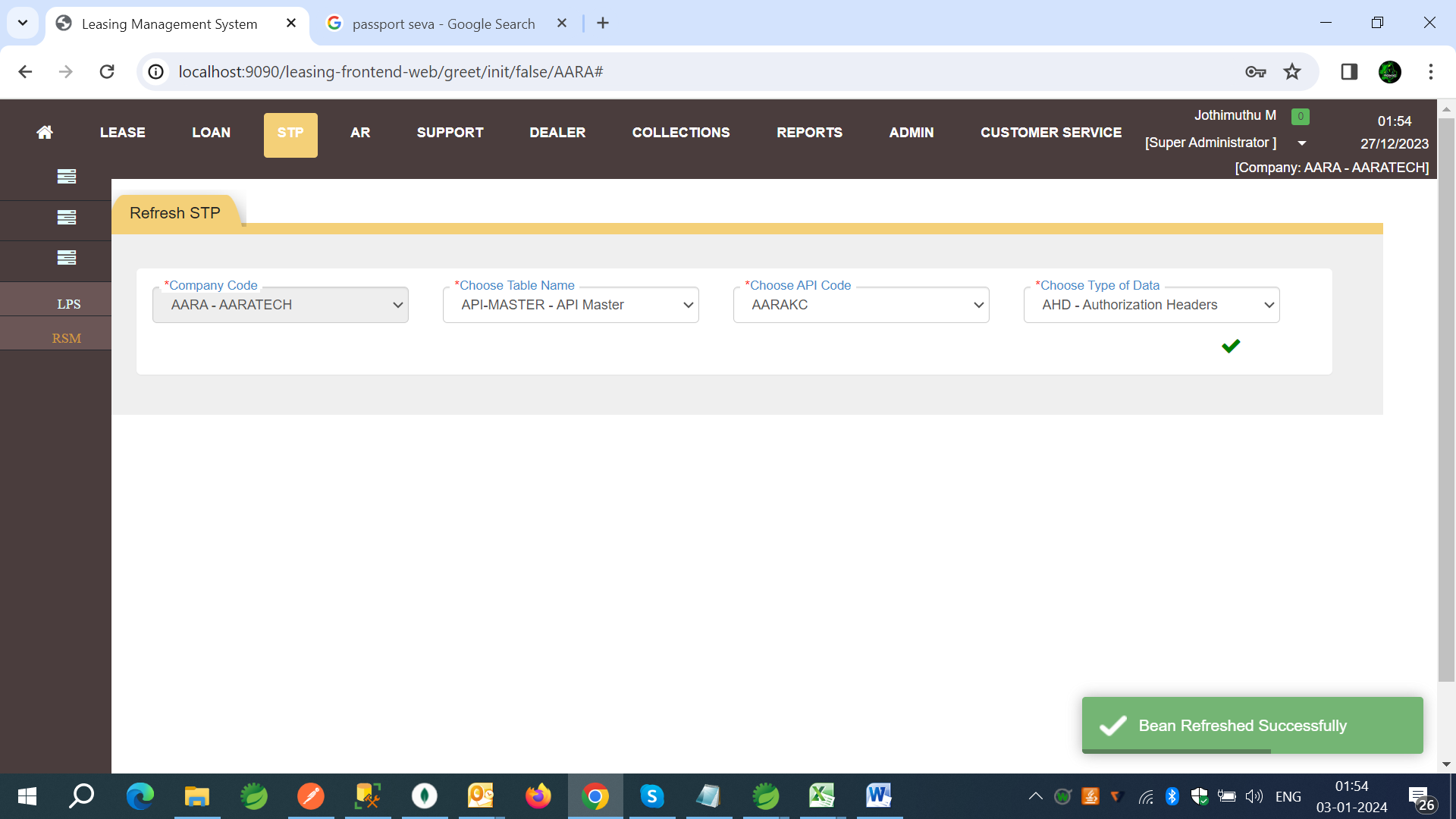
The STP System is taking all values inside the memory running if any changes made on the set-up the values will not be updated to the system on doing.

****

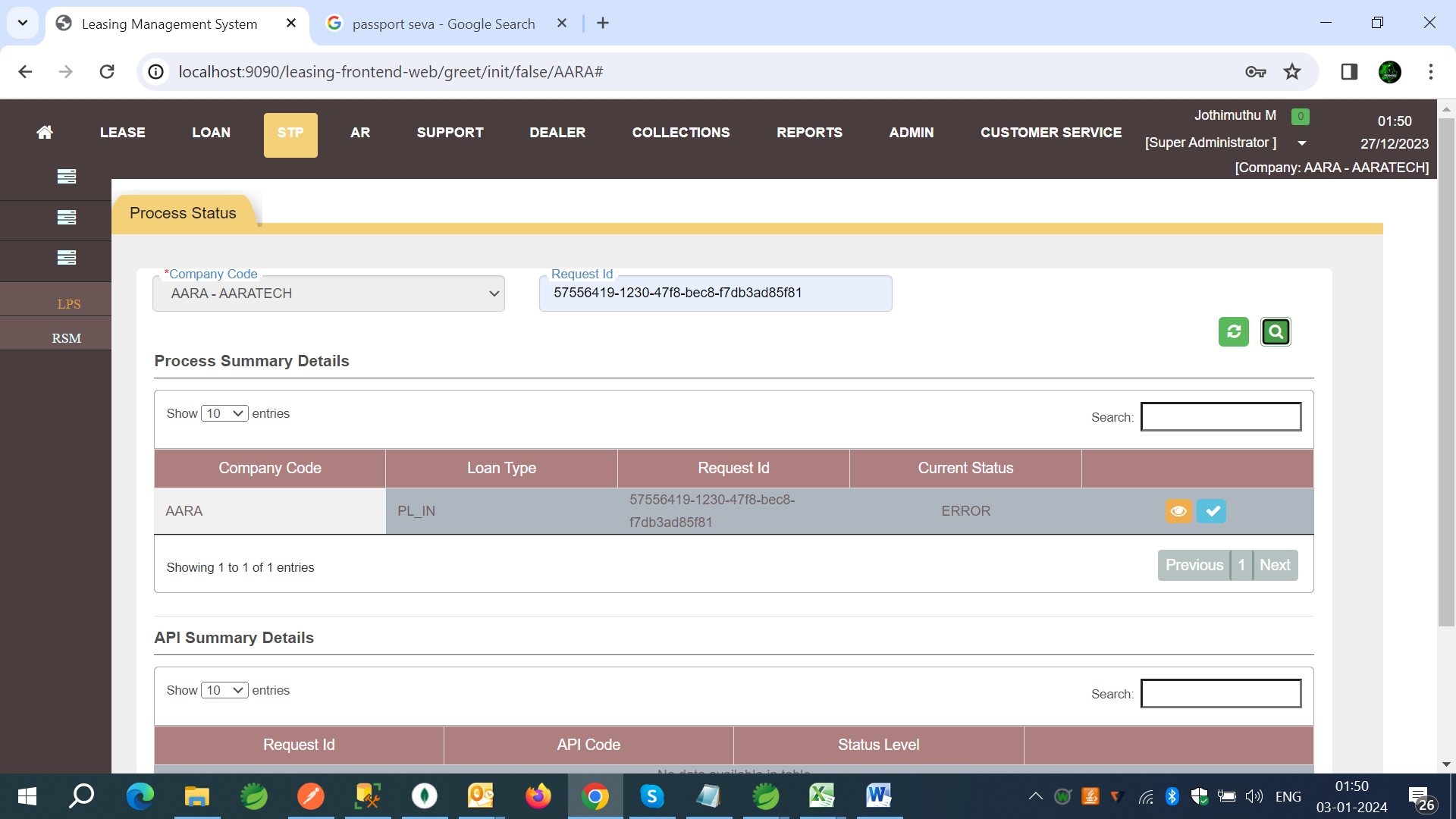
Here Comes the Refresh Bean, The Tables can be choose from the Data set (API Master, Process Master, Sub Process) Based on the selected table and data the system will updated to the current data.

Note: Latest of STP only supports for listed tables and use with care.

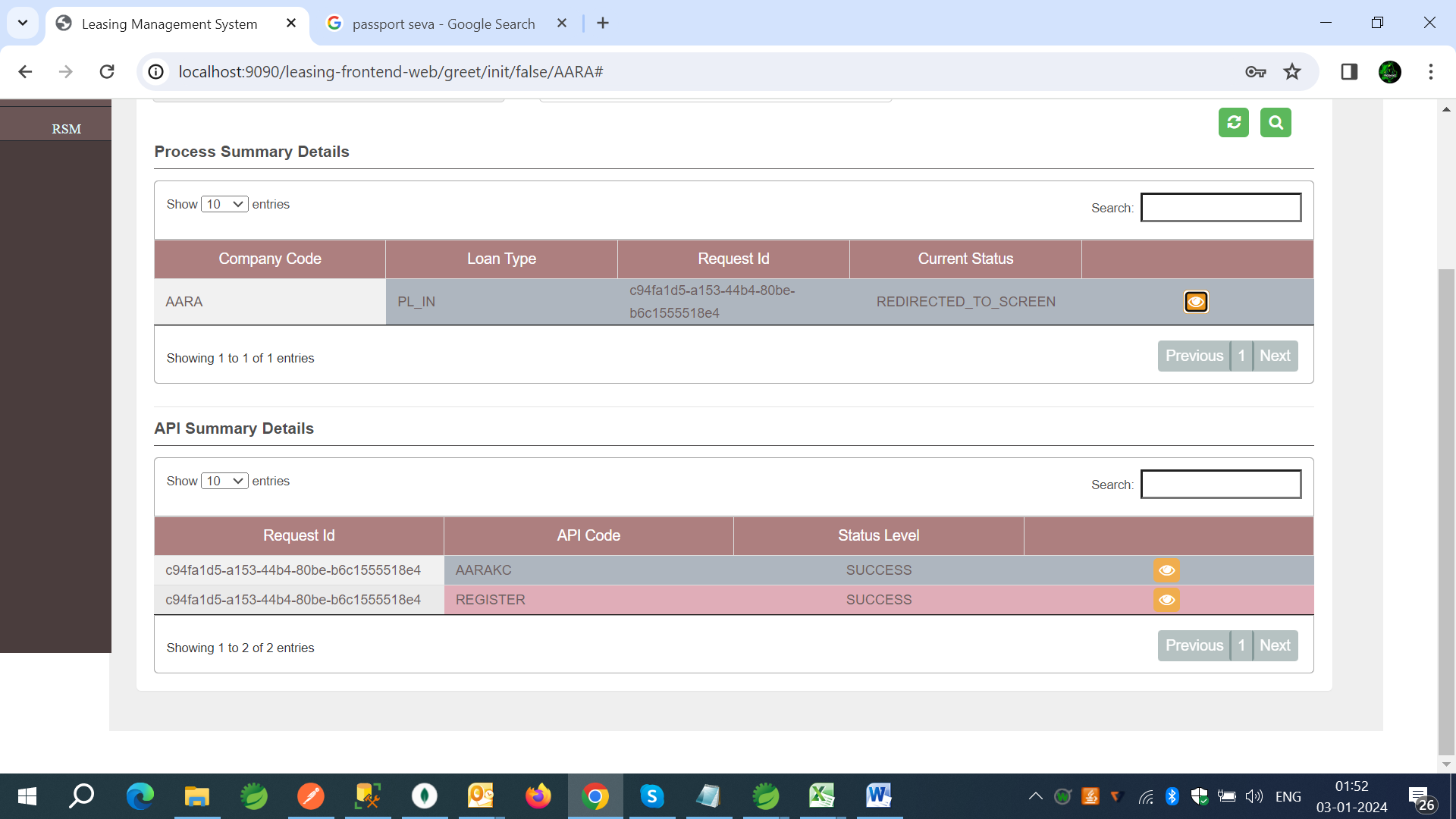




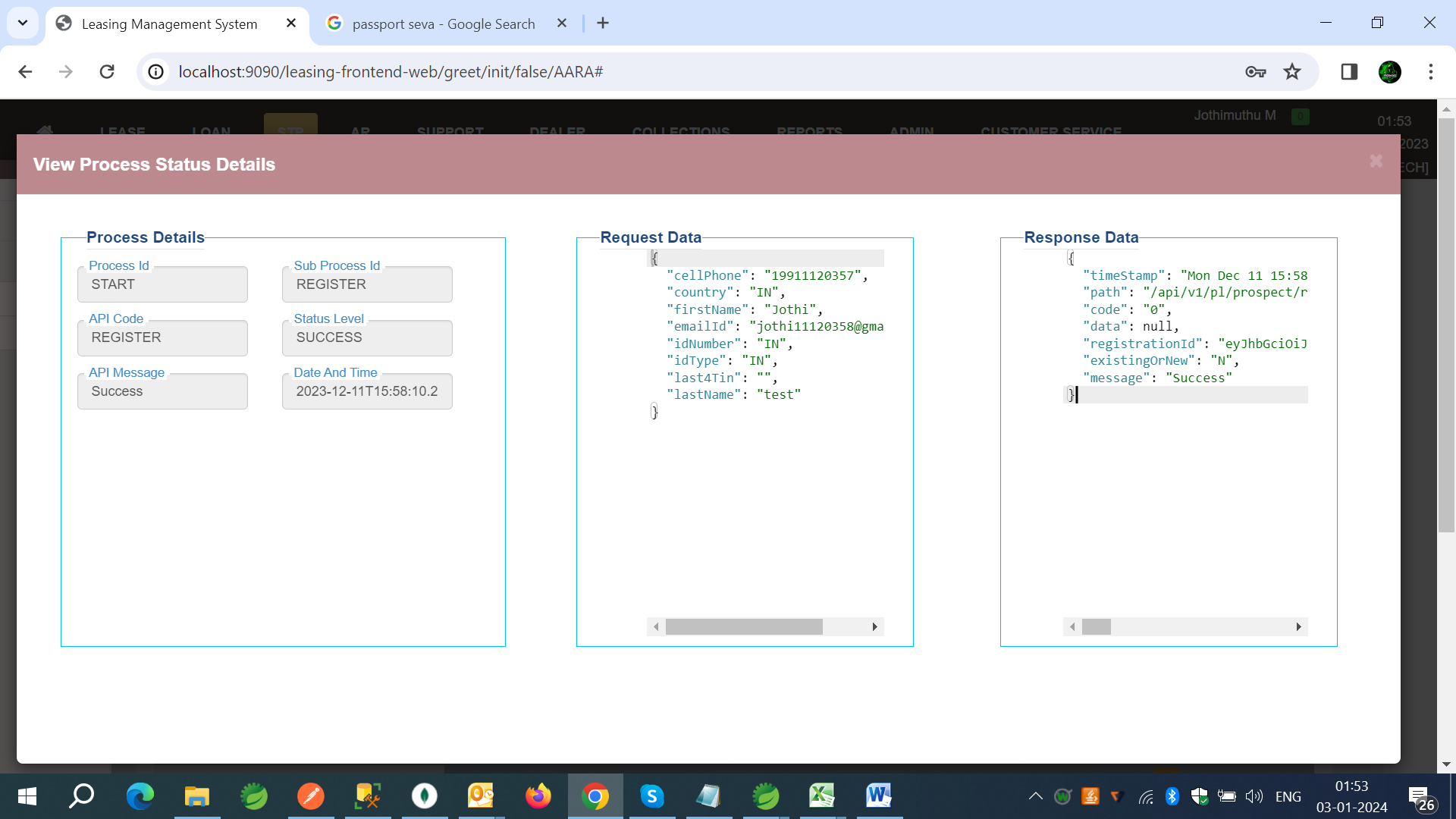
****

**STP Service: Process Status -**  Process Status we can see the history of transactions based on Request id which is returned by STP Engine API.

API Summary Details on Click of View Button will view the List of API’s compiled and called during the process.



****

On Click of API Summary Details record View the details of API’s as Date and Time and API Request and Response will displayed.

**STP Engine:**

API Details: All The supported API’s are attached below in Refrences section.

**Describing STP API:**

STP is supporting to Invoke the Process from endpoint (/invokeSTP) can be found on Collections shared below. For Authentication Key Cloak Token system is integrated call the endpoint of (/authenticate) for Token. Token is validate for 3 mins.

**Sample Request and Response of STP:**

**Request:**

{

    "companyCode": "AARA",

    "requestId": "",

    "STPLoanType": "PL\_IN",

    "stageToTrigger": "LOANBASSCN",

    "identificationId": "",

} Any Data can be sent along with this data , system will treat these data’s as Application Data.

**Response:**

{

    "code": "500 INTERNAL\_SERVER\_ERROR",

    "message": "FAILED",

    "requestId": "9a18a1f0-7e5c-483d-ad01-14a4a8f007b1",

    "identificationId": "",

    "companyCode": "AARA",

    "loanType": "PL\_IND",

    "timeStamp": "2024-01-09T02:24:27.305432500",

    "path": "/sample",

    "redirectedScreen": null,

    "nextProcess": null,

    "data": [],

    "errors": ""

}

****

Note: The Request validations and response structures are showed in swager file, can use of swagger file which is also tagged in Refrences section.

**Request Description:**

companyCode\* (String) client company code which is created in STP screens set-up.

requestId (String) Request id which is generated by STP system. For recalling from STP/Application side or as screen process Request id should send as part of request.

STPLoanType\* (String) To Which loan Type invoking STP.

stageToTrigger\* (String) Process code, for Start process default value is “START”.

identificationId (String) The Client side data can be passed to build a hand shake between stp and application if needed.

**Response Description:**

code – tell the API status code of invoke STP.

message – message of ivoke STP API.

requestId – Request id as unique id of STP.

identificationId - The Client side data can be passed to build a hand shake.

companyCode – company code of Client.

loanType – Loan Type passed from Request as STPLoanType.

timeStamp – Date and time of API Responded.

path – path of API from Both STP and Process API’s.

redirectedToScreen – if Process stoped as screen this flag will be returned as “Y”.

nextProcess – if Process stoped as screen , screen id will returned here.

data – extra information data will be passed here.

errors – errors of both STP and API’s will be returned here.( Client application handle this).

**Logging and Trouble Shooting:**

Visit to STP Application for Logging and Trouble shooting.

For any STP Transaction Details, use STP screens to view the Process status based on Request Id in Process Status screen.

****

**Conclusion:**

An additional layer of security and accuracy is embedded in STP through the principles of the maker and checker. This fundamental concept dictates that for every transaction, at least two individuals are required for its completion. This dual-layered approach ensures a thorough review, obtaining a second pair of eyes to spot any anomalies, inconsistencies, or potential errors. By adhering to the

Maker and checker principle, STP not only expedites processes but also enhances the overall reliability and integrity of business transactions.

In summary, Straight Through Processing (STP) is a key component of the business operations revolution. By combining standardized APIs and compiling processes into a STP application, companies can quickly set up and streamline their workflows. Organizations may respond to changing business requirements without having to make changes to their current operations thanks to STP's adaptability. Furthermore, the flexibility to configure the complete business-related flow via STP screens makes for a quick and easy installation.

**References and Links:**

|  |  |
| --- | --- |
| **Swagger File** |  |
| **STP Screen URL** | http://172.16.128.4:9090/leasing-frontend-web/greet/init/false/AARA# |
| **STP Engine Collection** |  |
| **STP Tables and Tech Stack Details** |  |

**Contact: info@aaratech.com**