A picture containing sitting, monitor, plate, clock

Description automatically generated

**Functional Specification Document**

**Visual Investigator**

Version 0.2

# **Table of contents**

[**Table of contents** 2](#_Toc111310449)

[**1.** **Document Change History** 3](#_Toc111310450)

[**2.** **Introduction** 4](#_Toc111310451)

[**2.1.** **Purpose of the document** 4](#_Toc111310452)

[**2.2.** **Project Scope** 4](#_Toc111310453)

**3. Solution flow…………………………………………………………………………**4

**3.1 High level Workflow (Part 1).…………….….……………………….……**5

**3.2 High level Workflow (Part 2) ……………………………………………...**6

**4. Functional Modules**………………………………………………………………..7

**4.1. Activity Pages…………………………………………………………………….**7

**4.1.2 Features & Description…………………………………………………….**7

**4.2 Widgets………………………………………………………………………………**7

**4.2.1 Purpose…………………………………………………………………………..**7

**4.2.2 Data models & Description…………….……………………………….**8

**5. ETLs………………………….………………………………………………………….**8

**5.1 ETL Process flow………………………………………………………………...**8

**6. Technical Specifications………………………………………………………..**9

**7. Workflow…………………………………………………………………………….**10

# **Document Change History**

# 

|  |  |  |  |
| --- | --- | --- | --- |
| **Modified By** | **Description of Change** | **Versions** | **Date of release** |
| Pradyot Parashar | Document Creation | 0.1 | 13th Aug 2022 |
| Pradyot Parashar | Change in workflows | 0.2 | 25th Aug 2022 |
|  |  |  |  |

# **Introduction**

# **2.1 Purpose of the document**

The purpose of this document is to specify the functional capabilities of the Visual Investigator and its functional requirement.

This document captures the detailed functional specification of the design of the modules and widgets that would be part of deliverables of the project.

# **Project Scope**

The scope of this covers the functional details for the Visual Investigator as well as system capabilities based on below lined items.

* Solution flow
* Activity Pages
* Widget Page
* Reports and Dashboard Page

1. **Solution flow**

Following steps will define the solution flow

* Client is expected to load data in a staging area with a fixed data structure format
* ETLs will read data from staging and apply transformation and derivation on the data and load into presentation layer
* Java API’s will fetch the data from presentation layer and prepare visuals in visual Investigator

**3.1 High level workflow**

**Part 1:**

Page Load

Credit

Debit

De

KYC

General Info

Alert History

Activity Page

Select Page Page

Create Widget

**Part 2:**

Selected Page

Widget Created

Specific Widget 1

Specific Widget 2

Specific Widget 5

Specific Widget 3

Specific Widget 4

GET API response POST API response

Data Fetch

OK

Decision Layer

Decision Layer

Not OK

Data Presentation Layer

OK

Not OK

ETL

Data Mart

# **Functional Modules**

There are five types of activity pages provided and they all are fixed. According to their names they hold particular information about debit, credit KYC, the general information and alert history. User is expected to create widgets of these pages only. The different types of data situated with all five of them, user can easily analyse all the different trends and can take the insights.

**4.1**  **Activity Pages**

**4.1.1 Purpose**

To allow the users to select various activities like Debit, Credit, KYC, General Info and Alert history. This page will contain icons for these activities that users would be able to click as per their requirement.

**4.1.2 Features & Description**

|  |  |
| --- | --- |
| **Features** | **Description** |
| **Debit** | User can analyse the debit data using widget by clicking on debit option from activity page |
| **Credit** | User can select Credit option from activity page for analysis and visualisation using widget |
| **KYC** | User can select KYC option to analyse the customer data using widget |
| **General Info** | User can view general information related to transaction from activity page and analyse the same using widget |
| **Alert History** | User can check the transaction alerts etc. from Alert history option and analyse the same using widget |

**4.2 Widgets**

**3.2.1 Purpose**

Widgets will call the already existing API that will fetch data according to the customer id and populate the data as per widget format or graph.

Java will be used to create APIs to get the data from presentation layer, which will be based on the various frequency or filters selected. Once widgets are created the page will be saved as a template

**4.2.2 Data Models & Description**

|  |  |
| --- | --- |
| **Model** | **Description** |
| **Transactional**  **Data model** | The transactional data like credit and debit etc. would be available for analysis through widget after ETL process |
| **Data Processing model** | Data stored in Data mart will be processed through ETL and available for analysis through widget |

**5. ETL**

ETLs will read data from staging and apply transformation and derivation on the data and load into Presentation Layer. Pyspark or SQL ETLs both can be used but Pyspark gives the advantage of connecting with different databases and libraries to transform data.

**5.1 ETL process flow**

Postgress

Oracle

MY SQL

Extracting data from different sources

Staging Area

Load the extracted data

Data Presentation Layer

Transform (presumably important data)

for analysis

PySpark

**6. Technical specifications**

1. **For ETLs**

* Pyspark or SQL ETLs both can be used but Pyspark gives the advantage of connecting with different databases and libraries to transform data

1. **Database**

* Application database can be MS SQL or postgres

1. **Frontend**

* R&D is ongoing on how we can render the charts and graphs on UI

1. **Backend**

* Java can be used to create API’s to fetch data from presentation layer based on the various frequency or filters selected

**7. Workflow:**

Fetched data from the presentation layer

With the help of API (Backend Process)

GET API response POST API response

Data Fetch

OK

Decision Layer

Decision Layer

Not OK

Data Presentation Layer

OK Not OK

Raw Data Mart

Queries used to pick data from Raw data mart

& Put into the Presentation Layer