A picture containing sitting, monitor, plate, clock

Description automatically generated

**Functional Specification Document**

**Visual Investigator**

Version 1

# **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 and Part 2).…………….….…………………………………………………….…..…………….……**5

**3.2 User Journey …………………………………………………………………………………………………………….………………………...**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

# **Document Change History**

# 

|  |  |  |  |
| --- | --- | --- | --- |
| **Modified By** | **Description of Change** | **Versions** | **Date of release** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **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

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
* User will Login and the page will load and now activity pages will be displayed.
* These activity pages are fixed and they will be selected accordingly by the user.
* Once a page is selected, user can create templates (E.g.: Debit 1 for Debit page)
* Among these templates any of them could be selected to further create a widget.
* REST APIs 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 Template

Debit 3

Debit 1

Debit 2

**Part 2:**

Debit 1

Widget 1

Widget 2

Widget n

Widget 3

Widget 4

UI

If OK then show Data on UI else show an error message in case of exception

Data Presentation Layer

Data Mart

ETL

* 1. **User Journey**

This part of the document covers user journey end-to-end. Along with description of the components wherever necessary.

User Login

Landing Page

1. General Info  
2. Credit  
3. Debit  
4. Alert History  
5. KYC

Menu Items

Templates

Widgets

Widgets (UI)

**Visualization Rendering**

Template 2

Widget 1

Template 1

Template n

Widget n

Widget 2

Presentation Layer

API server  
erver

Response: OK

Response: Failed

**1. User Login:** Username and Password based basic-auth will be used for authentication purpose. Once user is authenticated it’ll be redirect to the landing page.

**2. Landing Page:** This page consist of block menu of the existing items on which the analysis shall be performed. User can now select a customer and check any one these item.

**3. Menu Items:** List of items available for the current visualization. For this project scope we have 5 items mentioned above in flow. On selection user goes to the template section.

**4. Templates:** Once landing on template section, user has option to either run visualization on previously created templates or she/he can create a new template. Visualization will be generated in accordance of the template design.

**5. Widgets:** Widget is the building block of a template and very core feature of this application. During creation of a new template user can add any necessary number of widgets from the fixed list available.

Based on the template design the visualization data will be fetched from the presentation layer using REST API server.

# **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 save 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**

ETL

Postgress

Data Presentation layer

Pyspark

Connector

Oracle

MS SQL