|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Topic | **Fraud Detection** |

**Requirements specification**

# 1 Motivation

**1.1 Current situation:**

* All the payments done by Vendor is associated with some payment term.
* Payment term defines the invoice payment due date as per baseline date.
* There are different kind of payment terms for which a Vendor is eligible.
* Detection of anomaly is currently done by Fraud Management Team manually.
* For different modules help of different Subject Matter Experts are required.
* Auditing of data is done at different levels by visualizing them.
* There is an existing solution in SAP which checks for certain features and gives us the duplicate invoice only if it is an exact match.
* System will not allow the user to enter any duplicate invoice in a typical scenario
* Invoice comes with a Reference number which act as the unique identifier for any posting. Same Reference No. can occur for different Vendor ID’s or for different sets of amount.
* Posting date is filled manually so there is no restriction.

**1.2 Concerns or pain areas:**

* If any unusual payments are posted then it cannot be detected easily in large dataset.
* Every data has to be manually checked resulting in some fraudulent transactions getting missed.
* There is less transparency in checking which may lead to give some false results.
* Since lots of manual work is needed it is prone to errors.
* The process is very time consuming and resource intensive.
* The company incurs huge losses due to fraudulent transactions.
* Even if there is a slight difference between two invoices SAP will not catch it.
* If two Vendors are posting same amount at a particular date with similar Reference number then it won’t be caught.

# Capabilities

## 2.1 Roles

The following roles are supported in the application.

|  |  |  |
| --- | --- | --- |
| **Role** | **Actions** | **Responsibility** |
| Business Manager | * Handling Investigation * Closing of alerts | If no further analysis is needed close the alerts. If Any investigation is created take the necessary actions. |
| Business Analyst | * Alert/Notification Handling * Creating Investigation * Analytical Reports | View the alerts and try to find the anomalies. Forward the alerts to other suitable analysts. Add attachments and notes. Create investigations if need be. |
| Data Analyst | * Run Algorithm * Manage Data * Alert/Notification Generation | Run the algorithm on the required data with the suitable parameters. Once completed create alerts for the data. |

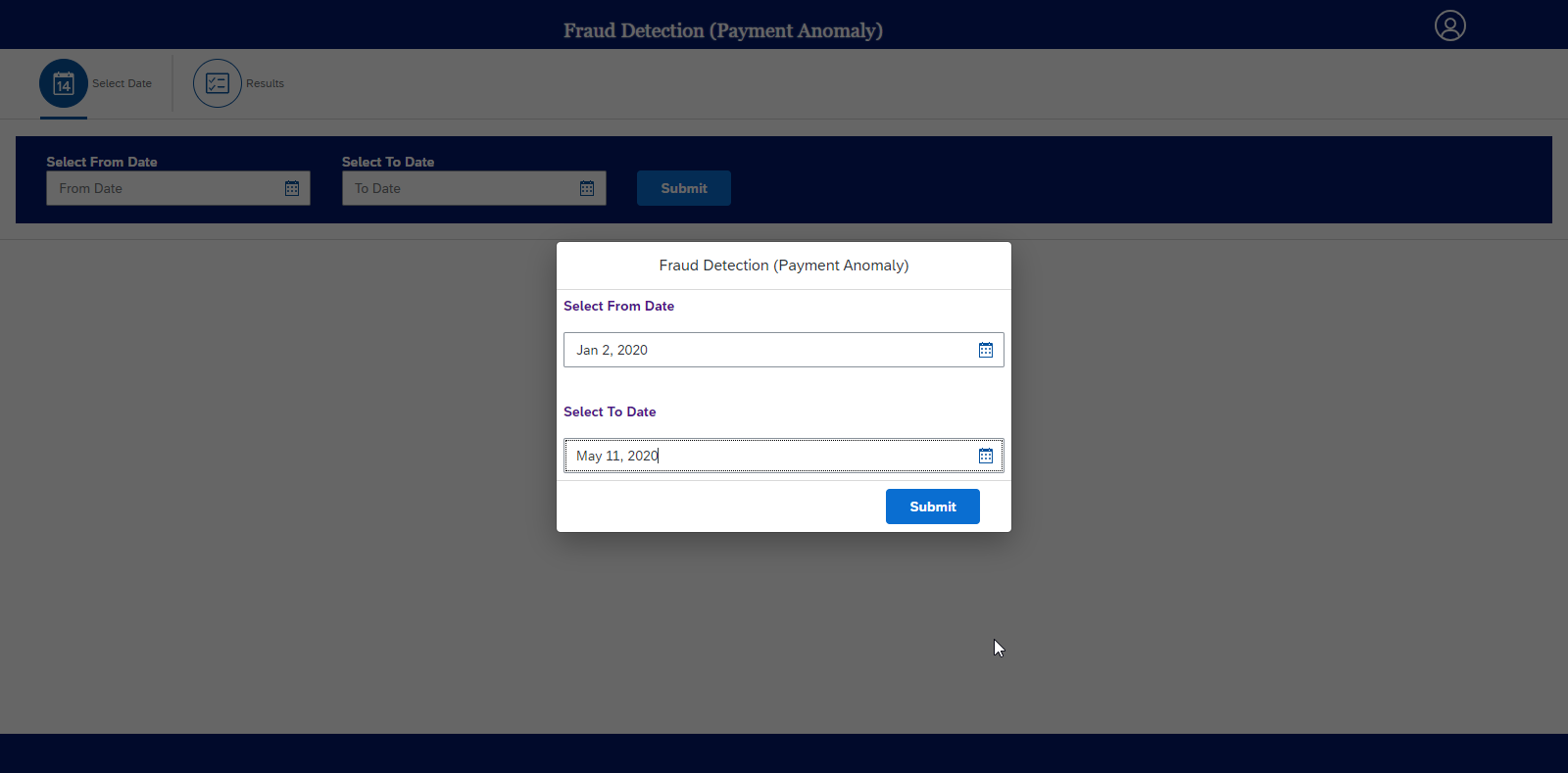
## Functionality

### Component Brief

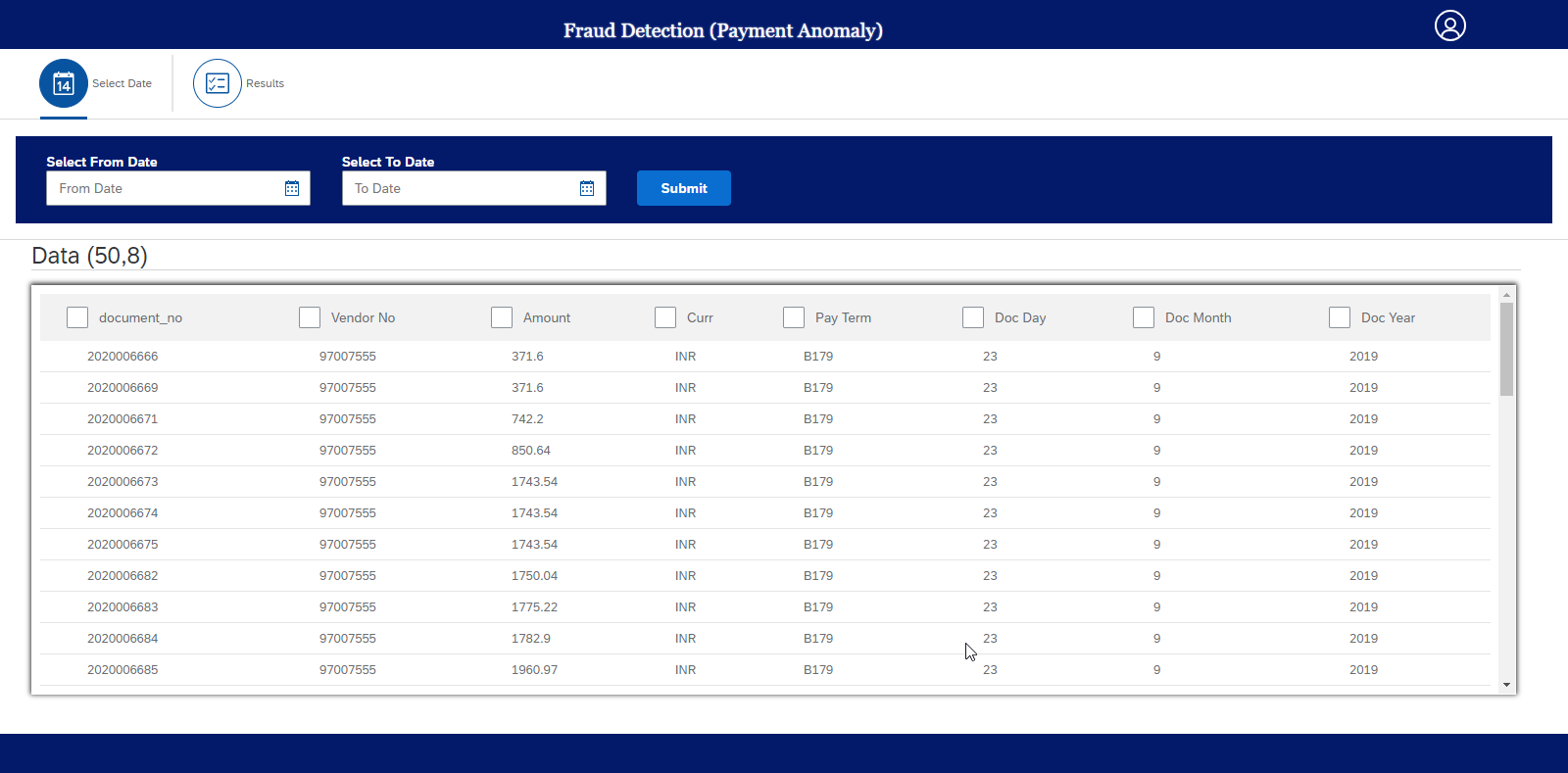
Users of Fraud Detection will have access to the following components.

|  |  |
| --- | --- |
| Payment Anomaly | Fraud detection in payment transaction.  Unusual payment terms used by vendor in transaction is called payment term anomaly |
| Duplicate Invoice | Fraud detection Duplicate invoice.  Duplicate Invoice postings happen when a single invoice received from vender is posted in the system more than once and consequently paid more than once |
| Alerts | The alerts are created automatically or manually and require action from the recipient either in this application |
| Alert Report | Report which gives the details of all the alerts that is generated  This screen allows the user to view all the alerts and its details in one place. |
| Event Details | Event details page will give timeline of operation done for the given alert such as Notes created forwarded etc. |

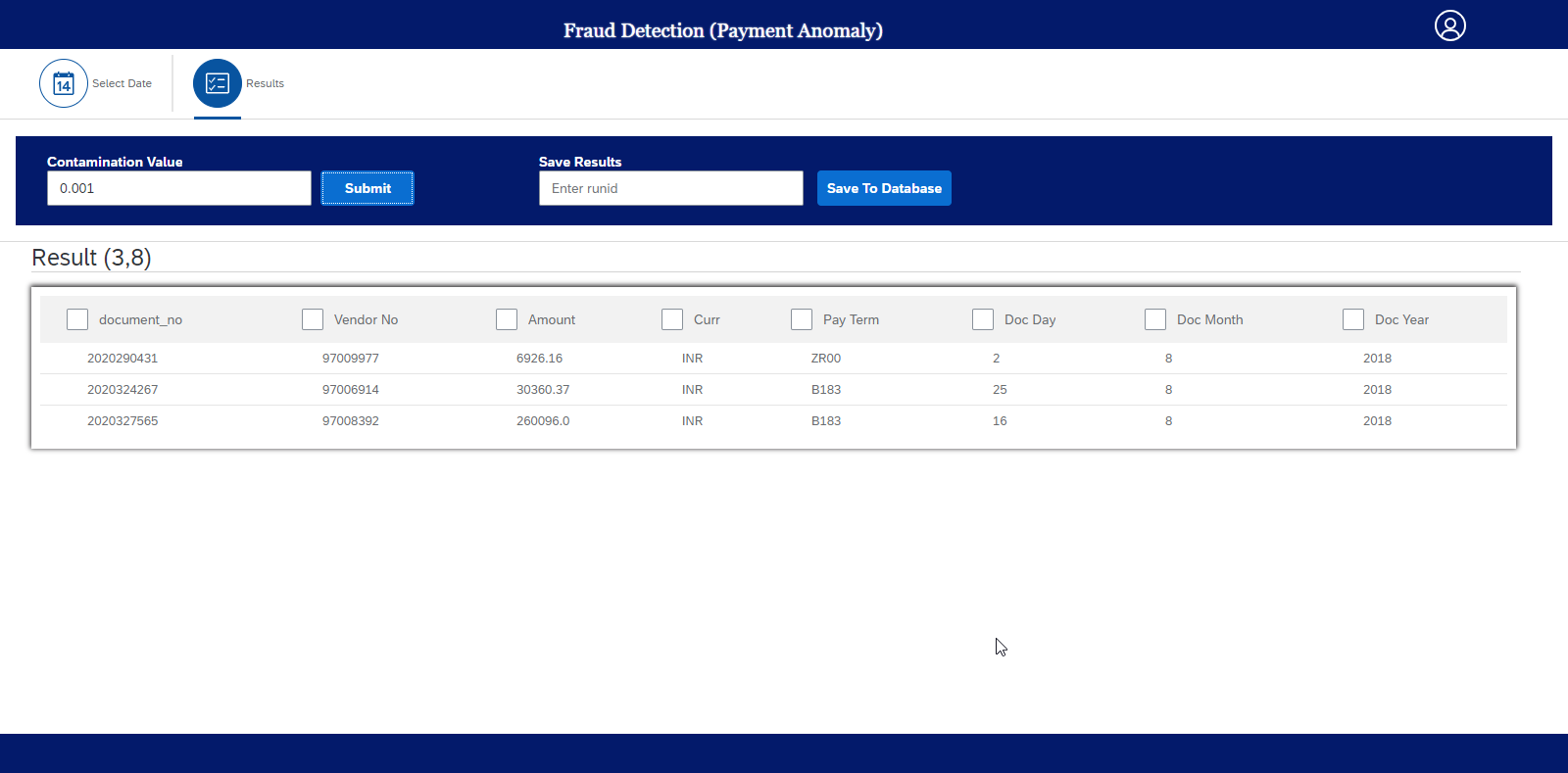
* + 1. **Selection of Dates**



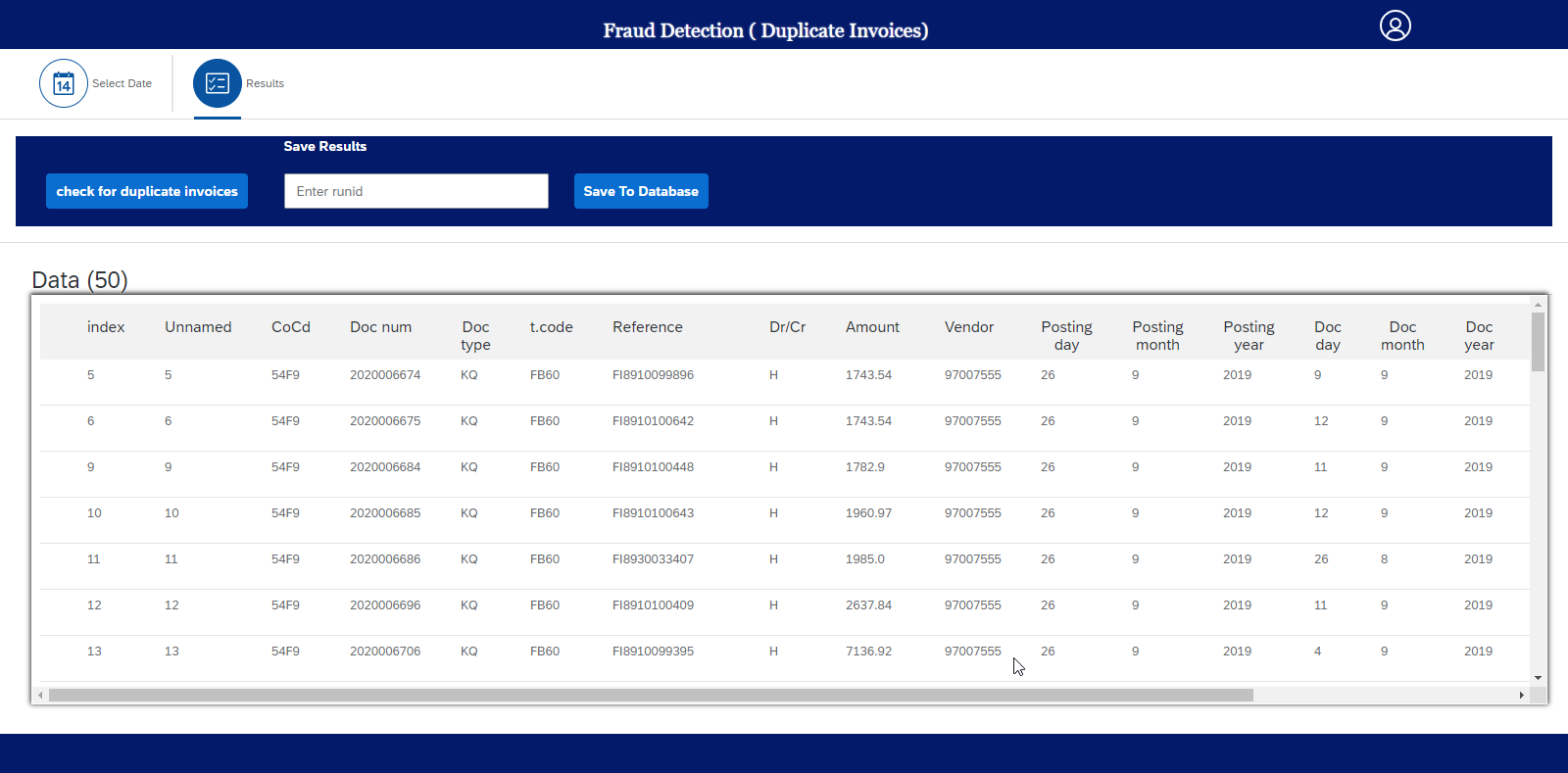
* User can select the ‘From’ & ‘To’ Dates. Which will in return fetch the transaction or records from the data base.
* Dialog box is displayed if the user is logging first time to the application.
* Dialog Box contains “From” and “To” date-picker .which is the input for API for fetching records
  + 1. **Display of Records**



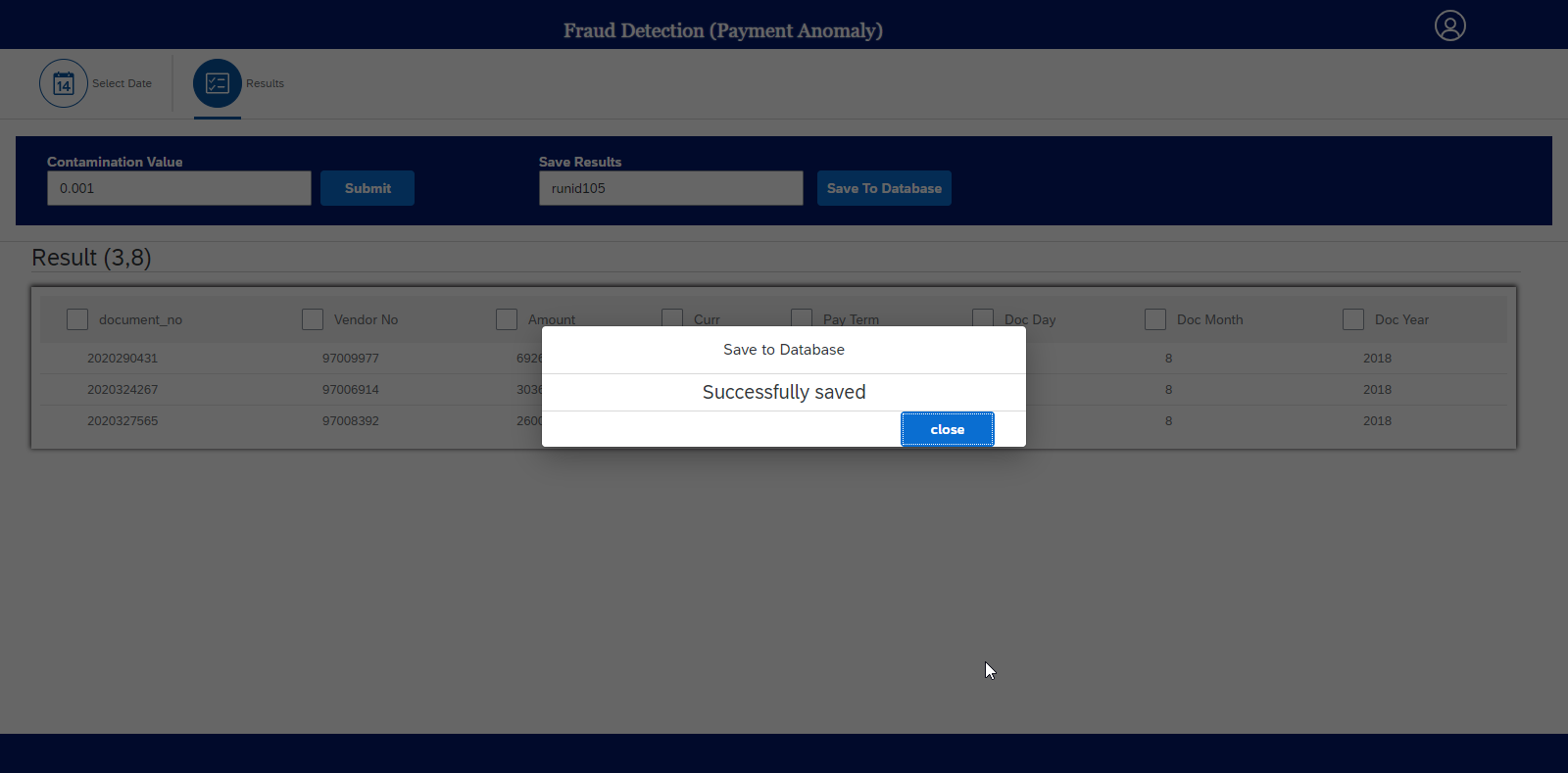
* Date picked from previous step will fetch the records from database.
* Records are displayed in tabular format.
* Row and columns are displayed in brackets
* User can even change dates in the same page.
* User can select the particular column for detection of anomalies based on selected columns. Selection can be done using checking the checkbox.
* If no data is fetched from backend the table will display no data found.
  + 1. **Contamination Value** / **Check for Duplicate Invoice**



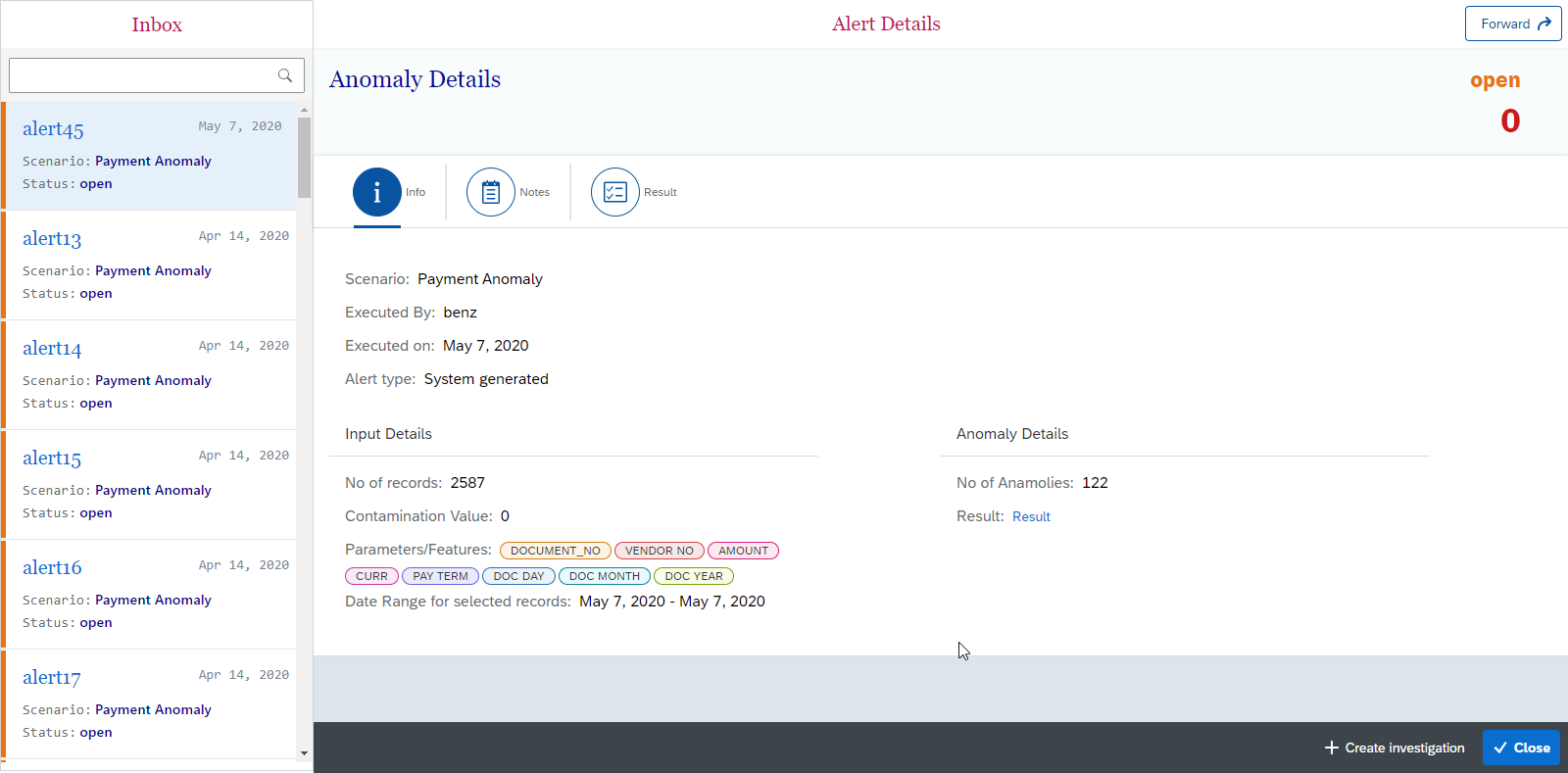
* Contamination is an important value for processing the algorithm. It stands for percentage of outlier points in the data.
* On submit of contamination value processing happens and result will be displayed.
* By default contamination value is zero.



* In **Fraud detection (Duplicate invoice)** check for Duplicate Invoice Button will fetch the result after processing.
* Result are sorted according labels and displayed
  + 1. **Save To Database**



* User can save the processing details to database.
* Details are:
  + From & To Dates
  + Contamination Value
  + Result name
  + User ID.
* Dialog box will give the status of saving to database.
* The alerts are created automatically and require action from the recipient in alert application.
  + 1. **Alerts Details**

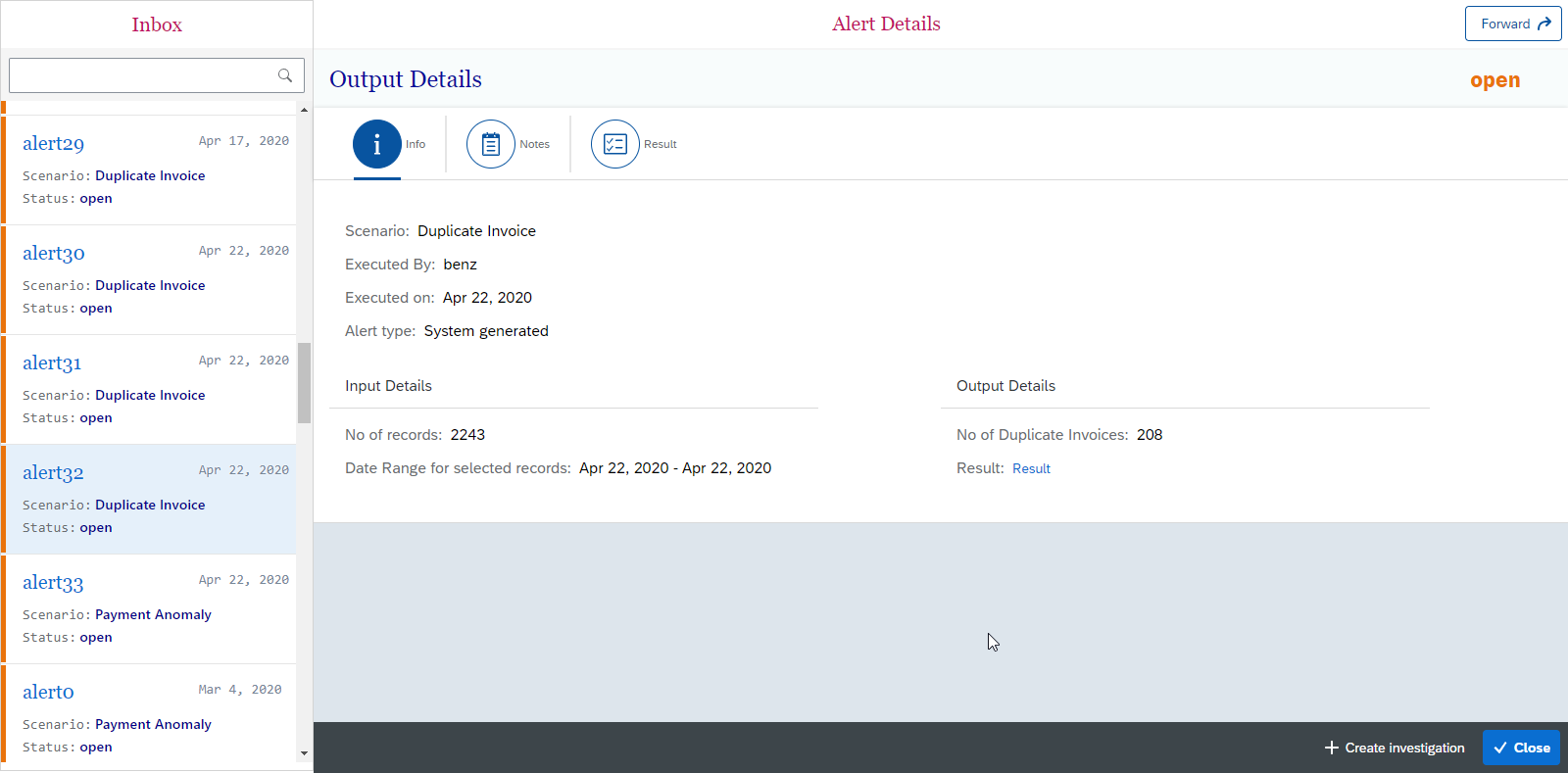


The alerts are created require action from the recipient in this application.

The above UI page follows Master-Detail layout.

**Master Layout**

* Alerts partial information is displayed:
* Alert name
* Scenario
* Status
* Date of creation
* Alerts generated can be of type payment or duplicate invoice
* Status of alert can be open held closed forwarded based on user action.
* Search filter to search the particular alert based on different parameter.

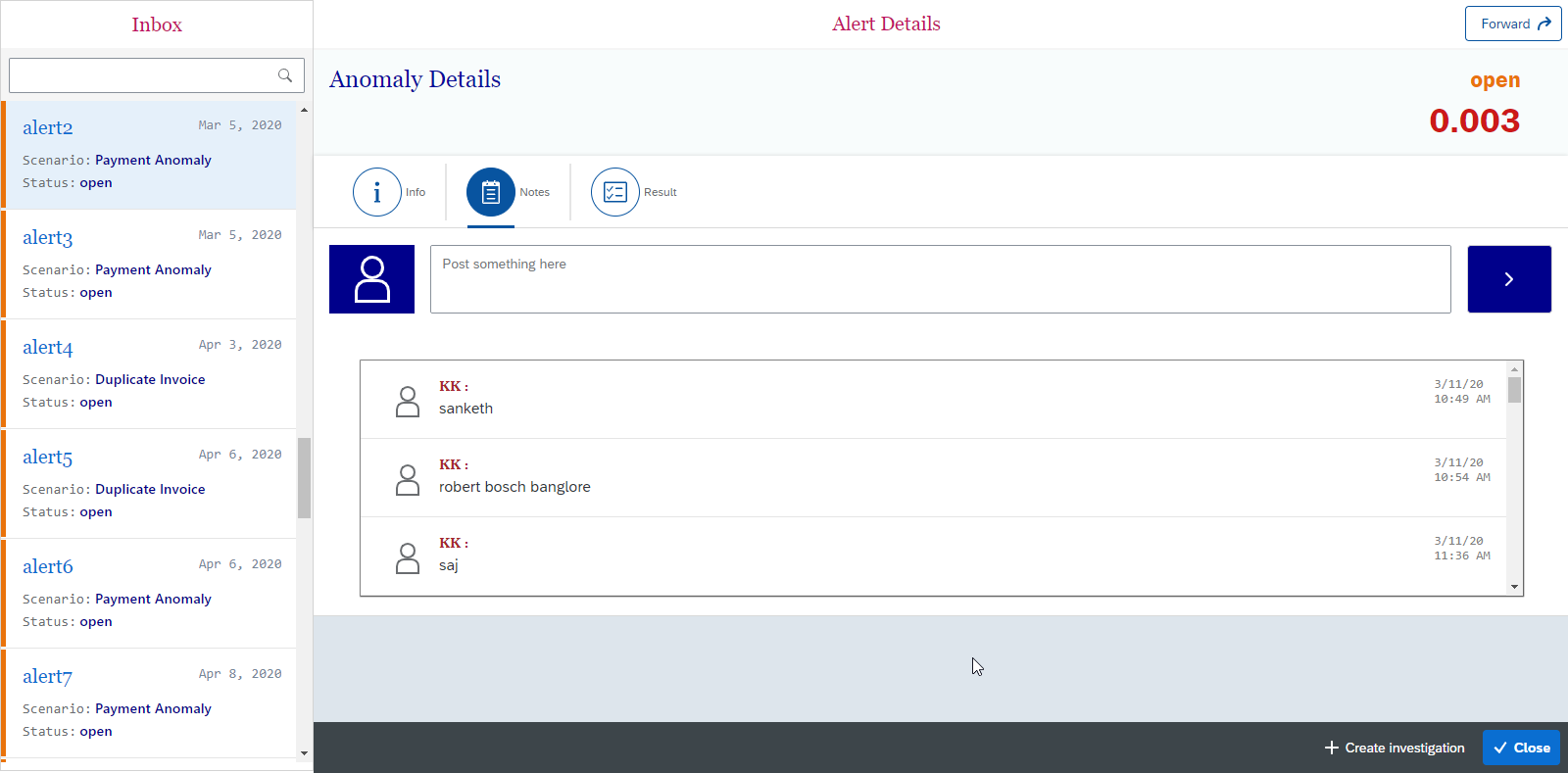


**Details Layout**

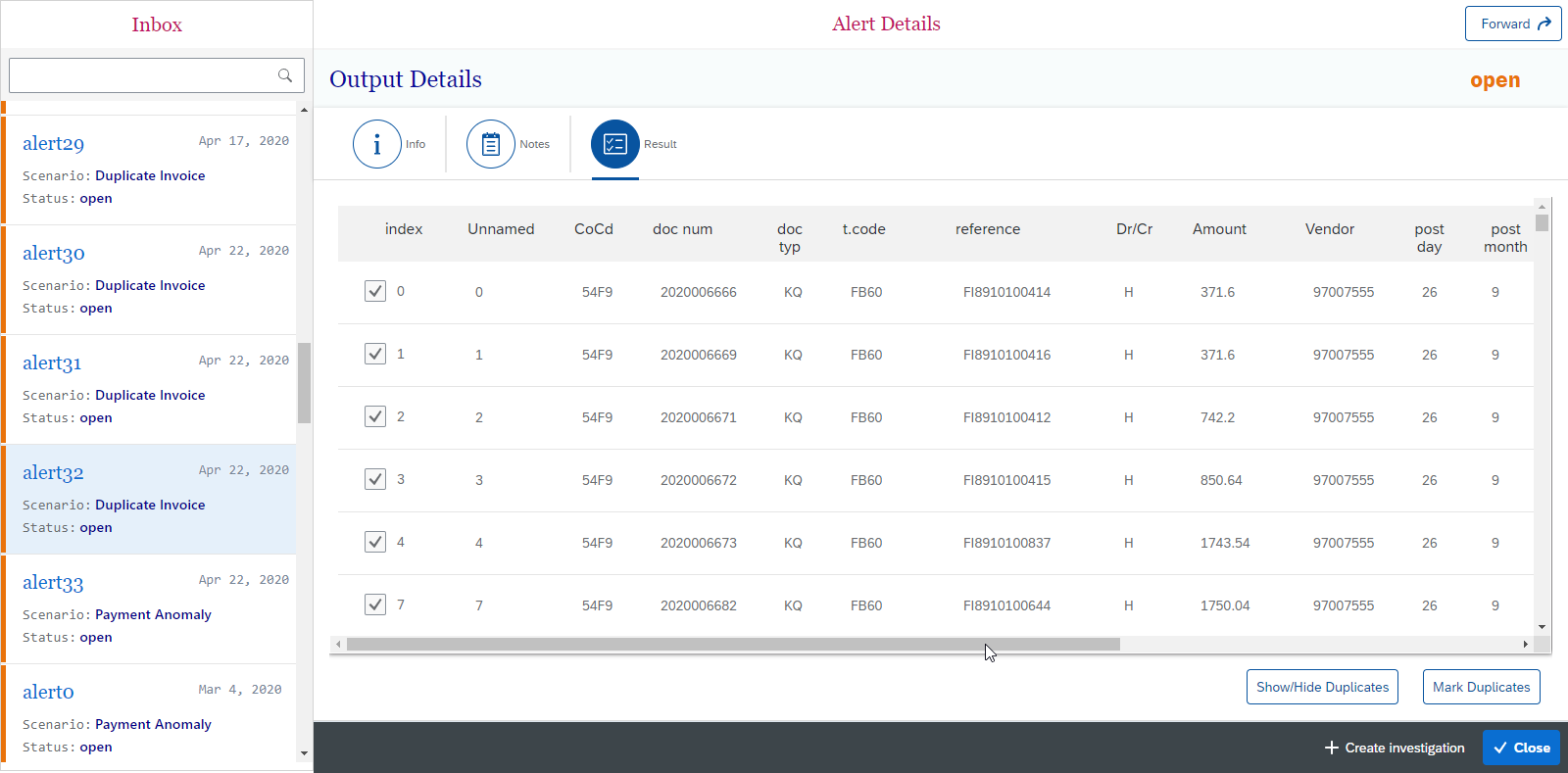
* Information of particular selected in master layout will be displayed.
* Information include :
  + Scenario
  + Executed By
  + Executed On
  + Alert type

**Input Details**:

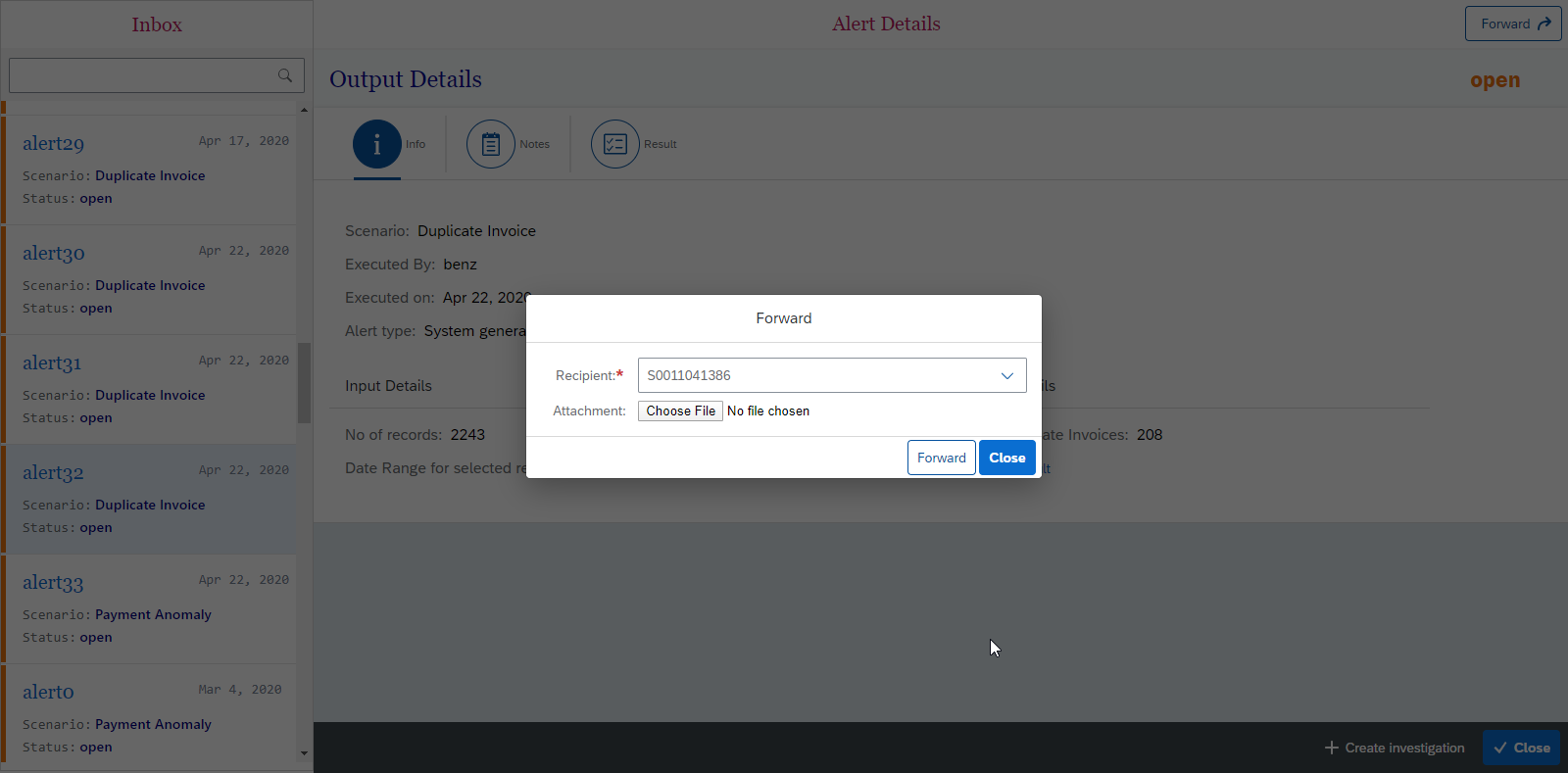
* + No of records
  + Contamination Value
  + Parameters/Features
  + Date Range
* **Anomaly details** includes : Number of anomalies and result
* User can download the result excel by clicking on the result text.
  + 1. **Alert Notes**



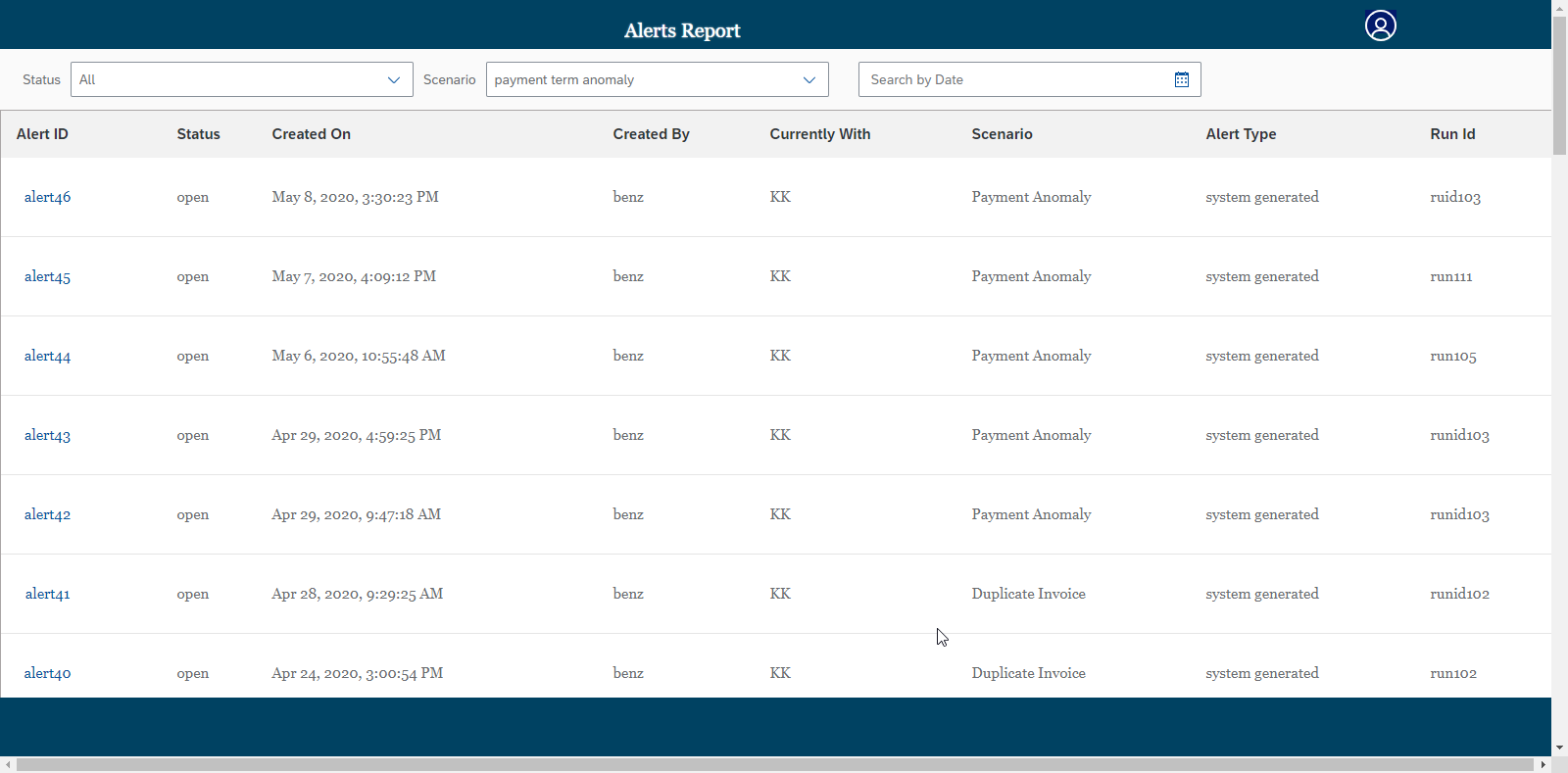
* User can write the notes which will help is future investigation.
* Notes inserted when user input the text and click on submit button.
* Notes contains
* Name of the user created the notes
* Notes
* Date and time of notes created
* Notes creation is logged and displayed in the event details page.
  + 1. **Result Tab**



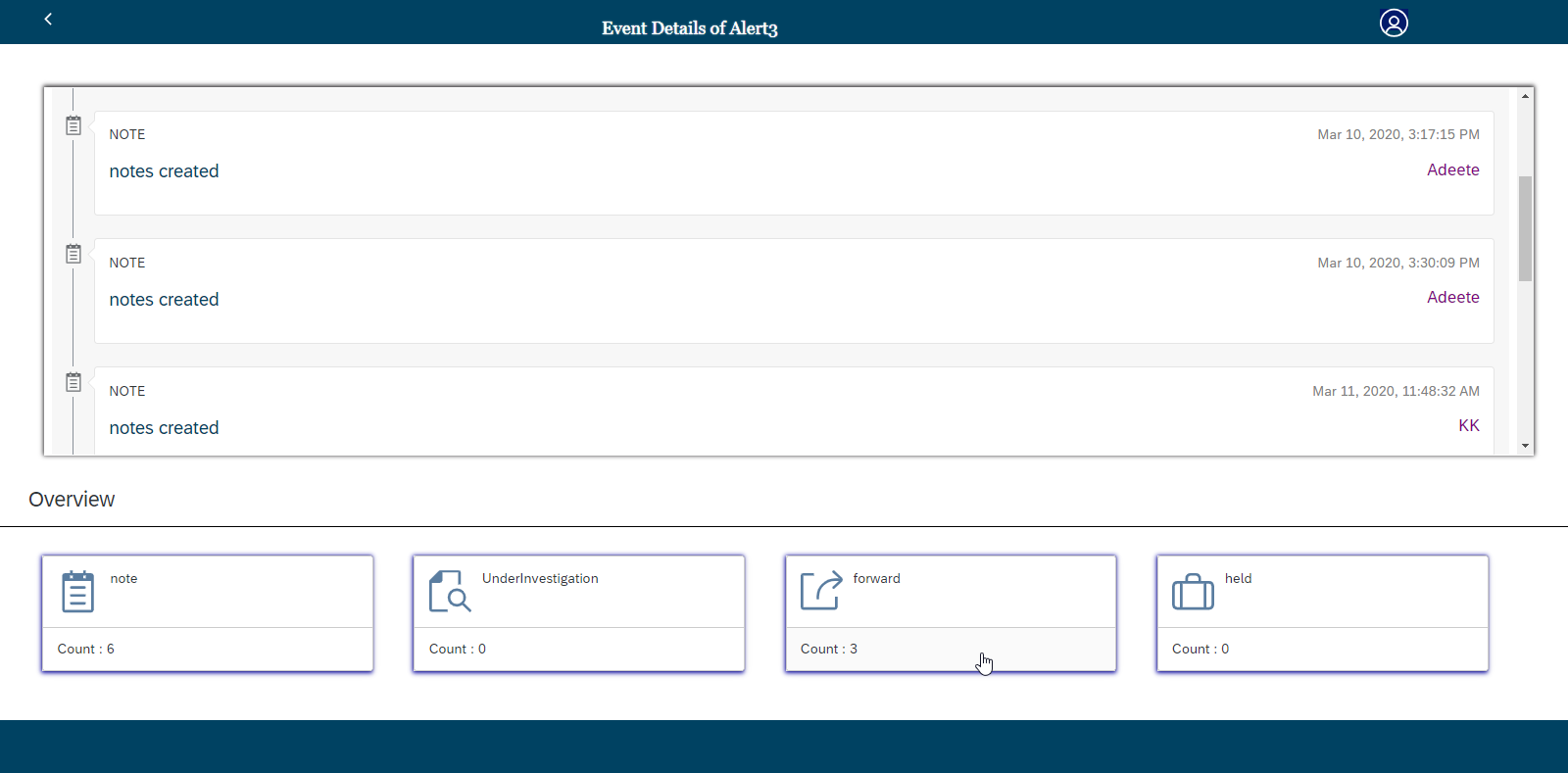
* Result tab display the records which can be anomalies.
* For scenario duplicate invoice user can mark the duplicates and submit to database.
* Show/hide duplicated used as a filter to hide already marked records.
  + 1. **Forward**



* User can forward the alert to one or more recipients.
* User can attach the file while forwarding
* Forwarded event is logged and displayed in the event details page along with recipient names.
  + 1. **Alert Report**



* This screen allows the user to view all the alerts and its details in one place.
* The details available here would be the following:
  + Alert ID
  + Status
  + Created On
  + Created By
  + Currently With
  + Scenario
  + Alert Type
  + Run ID
* Filters (Status, Scenario, and Date) used to search particular alerts.
* Alert IDs are linked to event details page. User can redirect to particular alert event details page by clicking on its alert ID.
  + 1. **Event Details page**



* Event details page will give timeline of operation done for the given alert such as

Notes created forwarded etc.

* Filters (Notes, Held, Forwarded and Under Investigation) used to search particular alerts event.