

Fraud Detection System in Retail Management

Submitted By

Anushree Juthani

17MCEI05



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INSTITUTE OF TECHNOLOGY
NIRMA UNIVERSITY

AHMEDABAD-382481

May 2019

Fraud Detection System in Retail Management

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in CSE(Information Network and Security)

Submitted By

Anushree Juthani

(17MCEI05)

Guided By

Dr. Swati Jain



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INSTITUTE OF TECHNOLOGY
NIRMA UNIVERSITY
AHMEDABAD-382481

Certificate

This is to certify that the major project entitled ”**Fraud Detection System in Retail Management**” submitted by **ANUSHREE JUTHANI (17MCEI05)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination.

Dr. Swati Jain
Associate Professor,
Department of CSE,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Sharada Valiveti
Coordinator M.Tech-CSE(INS),
Department of CSE,
Institute of Technology,
Nirma University, Ahmedabad

Dr. Madhuri Bhavsar
Professor and Head,
CE/IT Department
Institute of Technology,
Nirma University, Ahmedabad.

Dr Alka Mahajan
Director,
Institute of Technology
Nirma University, Ahmedabad,

Statement of Originality

I, **Anushree Juthani**, Roll. No. **17MCEI05**, give undertaking that the Major Project entitled "**Fraud Detection System in Retail Management**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering (CSE)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student

Date:

Place:

Endorsed by
Dr. Swati Jain
(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Dr. Swati Jain**, Associate Professor, Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for her valuable guidance and continual encouragement throughout this work. The appreciation and continual support she has imparted has been a great motivation to me in reaching a higher goal. Her guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Madhuri Bhavsar**, Hon'ble Head of Computer Engineering/ Information Technology Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. Alka Mahajan**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

Anushree Juthani

17MCEI05

Abstract

This report focuses on the system of fraud detection in retail management. The system helps service providers target on different categories of loss and frauds that can emerge in retails. It also helps in providing business analysis. The backbone of the system is POS which is the central point for detecting fraud transactions and anomalies. Reports are generated which use AI to detect whether the transactions are fraudulent or not. Also it consists of an auditing framework which records all logs of every activity done by the customer or service provider.

The UI framework has been designed using the OJET (Oracle Java Extension Toolkit). REST services are used for data retrieval and processing at every node. Javascript ES6 is used to further optimize and enhance the code for better performance.

Abbreviations

POS	Point of Sale.
DBS	Data Base System.
LP	Loss Prevention.
ELT	Extract Load Transform
FPS	Fraud prevention systems
FDS	Fraud detection systems
SFTP	SSH File Transfer Protocol

Contents

Certificate	iv
Statement of Originality	v
Acknowledgements	vi
Abstract	vii
Abbreviations	viii
List of Figures	xi
1 Introduction	1
1.1 Modules of Fraud Detection System	1
1.1.1 Need of an FDS in retail management	1
1.2 Objective	2
1.3 Scope of work	2
2 Literature Survey	3
2.1 Related Work	3
2.1.1 Case-study 1 : Survey of Retail Industries Losses	3
3 Data Flow and architecture	5
3.1 Overview	5
3.2 Application Workflow and Architecture	6
3.3 Log management and REST services	7
3.4 ELT Data Flow	8
4 Work done	10
4.1 Log Maintenance	10
4.2 Loss Prevention and Fraud Detection Analysis	13
4.3 Auditing framework for increased security	15
4.3.1 Auditing framework implementation	16
4.3.2 Audit Database	18
4.4 Scripts using cURL for secure authorization	19
4.4.1 cURL Implementation	20
4.5 Calendar Maintenance and Configuration	21
4.5.1 Calendar Maintenance	21
4.5.2 Calendar Configuration	24

5	Challenges faced	26
6	Future Scope	27

List of Figures

2.1	Chart Diagram of Retail Losses	3
2.2	Cookies/Session based Authentication/Authorization	4
3.1	Basic Application Architecture	5
3.2	Application Architecture of Loss Prevention System	6
3.3	Log management through REST	8
3.4	ELT flow of Data	9
4.1	Remote view of module	11
4.2	Log maintenance	11
4.3	Configuring Log maintenance	12
4.4	Detailed view of module	13
4.5	Dashboard for fraud detection analysis	14
4.6	Fraud Detection analysis and attributes	15
4.7	Audit Parameters	17
4.8	Audit Data Flow	18
4.9	Audit Framework Implementation	18
4.10	Audit Database Schema	19
4.11	Audit Database Data	19
4.12	cURL Implementation	20
4.13	Calendar Maintenance	21
4.14	Add Calendar	23
4.15	Calendar Configuration	25

Chapter 1

Introduction

1.1 Modules of Fraud Detection System

The advancement of use of technology have induced the performing of transactions through electronic commerce systems.e.g. Credit and debit card transactions, Amazon GO, etc. Sadly, these frameworks are utilized by both real clients and fraudsters. What's more, fraudsters used distinctive ways to deal with break the electronic business frameworks. Extortion anticipation frameworks (FPSs) are inadequate to give satisfactory security to the electronic trade frameworks. Be that as it may, the joint effort of FDSs with FPSs may be viable to anchor electronic trade frameworks. In any case, there are issues and difficulties that block the execution of FDSs, for example, idea float, underpins continuous discovery, skewed dispersion, vast measure of information and so on.[6] This study paper means to give an orderly and complete outline of these issues and difficulties that impede the execution of FDSs.

1.1.1 Need of an FDS in retail management

From recent years, in retail industry 14 % misfortunes are just acquired because of extortion and misfortunes. Client obtaining things from store and online locales and they returned it by organizing a few causes and reason. Client endeavor to play out some misbehavior As an aggregate of 92000+ situations are there that required be tried backward in this manner time plays a pivotal job in losses.[4] Manual Testing was dicult for getting wanted outcomes. Consequently, misfortune and extortion discovery prompts create powerfully reports and distinguish misfortunes and to increment

1.2 Objective

The objective of this report is to get to know and understand in brief how fraud management works effectively for retail management, which frameworks and modules are being used to detect frauds and generate reports dynamically, etc.

1.3 Scope of work

Fraud detection in retail for this application will be able to dynamically generate reports for fraudulent transactions by importing data in different formats as required through an importing engine and using SFTP for secure import and transfer. It will also generate logs of all activities through an auditing framework.

Chapter 2

Literature Survey

2.1 Related Work

2.1.1 Case-study 1 : Survey of Retail Industries Losses

Theft and fraud in retail industries losses in last year referenced in Figure 2.1. Referring to the figure, the major reason for this was dishonesty of employees and vendor fraud being the least amongst them .To handle this much amount of data, system should have edibility and scalability. Cloud gives that functionality so this system will design in such a way so it can move to cloud and serve the need of customers dynamically by auditing and generating prompts and reports.[1]

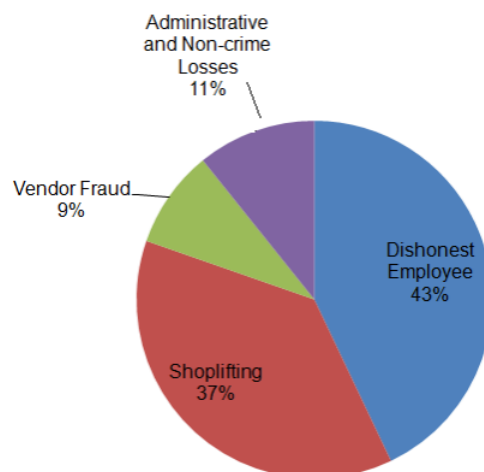


Figure 2.1: Chart Diagram of Retail Losses

Traditional Authorization method

Before JSON web Token was introduced in the market, Enterprises utilized customary session approach for one of a kind distinguishing proof of the client. In session approach server needs to keep up session for every single client and store into its memory or Database framework.[2]

Steps to authorize the user using Session/cookies approach:

1. Client demand for the assets by sending Username and Password
2. Server checks the information sent by client .
3. Server makes a new session and saves it into its database.
4. Server return Session Id (SID) to particular client and store into client cookies.
5. Customer demand next Resource with SID.
6. Server checks Session ID is substantial or not (via looking into DB).
7. Check the client job for approval of mentioned resources.
8. At long last User get mentioned Resource.

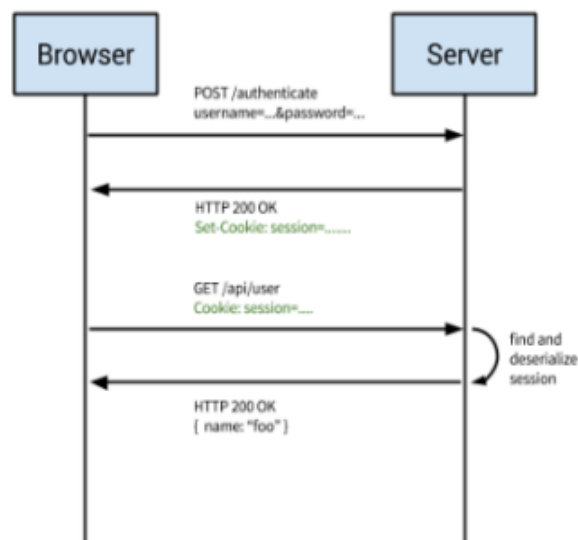


Figure 2.2: Cookies/Session based Authentication/Authorization

Chapter 3

Data Flow and architecture

3.1 Overview

As depicted in figure 3.1, the input data to the application consists of transaction data from any retail store or an ecommerce/inventory sales.

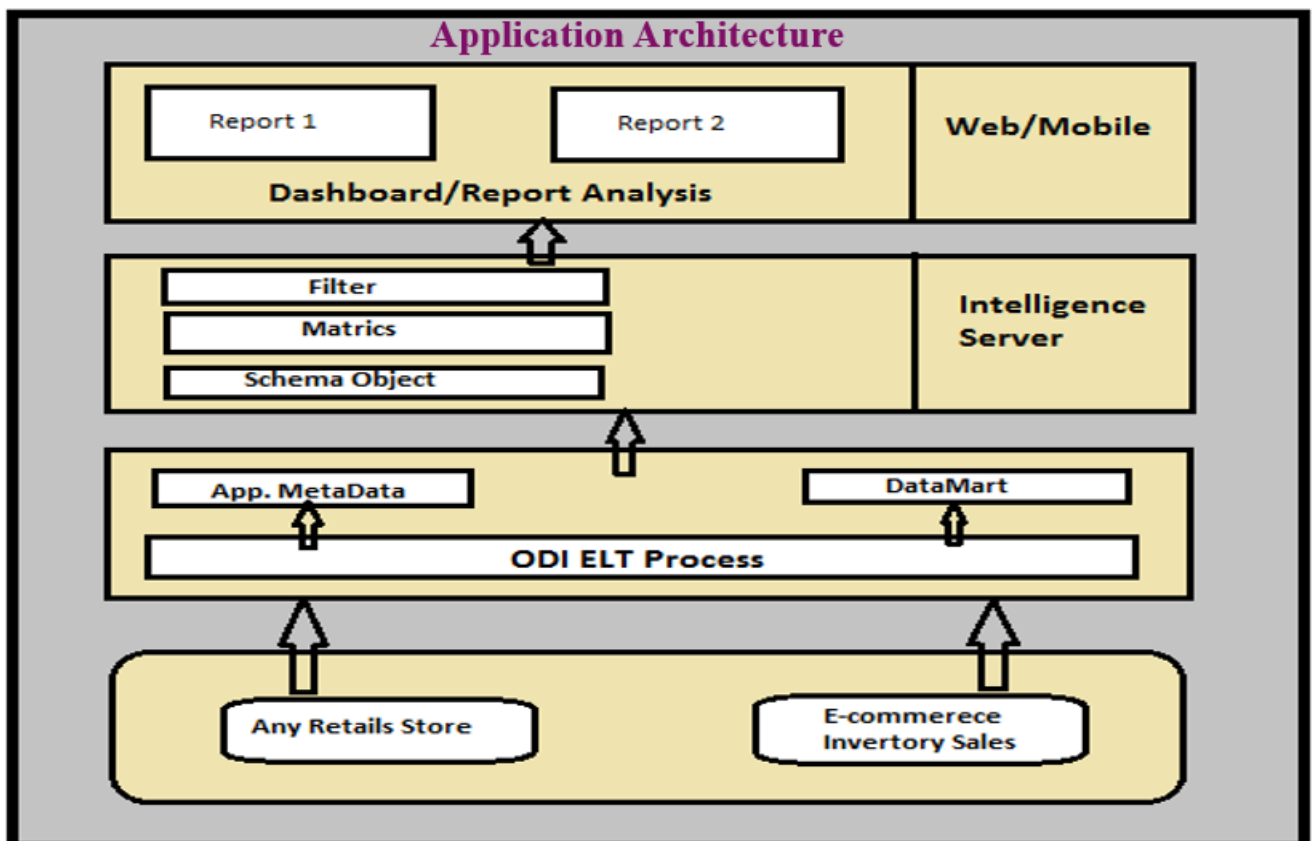


Figure 3.1: Basic Application Architecture

This data undergoes an integration and mining process further which it is categorized

into metadata and data which will further undergo mining indicated by datamart in the figure.

This categorized data undergoes detailed processing through an intelligence server - filtering, matrices, schema objects, etc. The filtering takes place based on various attributes like certain types of transactions, date and time, etc.

Based on this data, reports are generated stating in detail whether the particular data is anomalous or fraudulent. Dashboard generation and updation is done automatically.

3.2 Application Workflow and Architecture

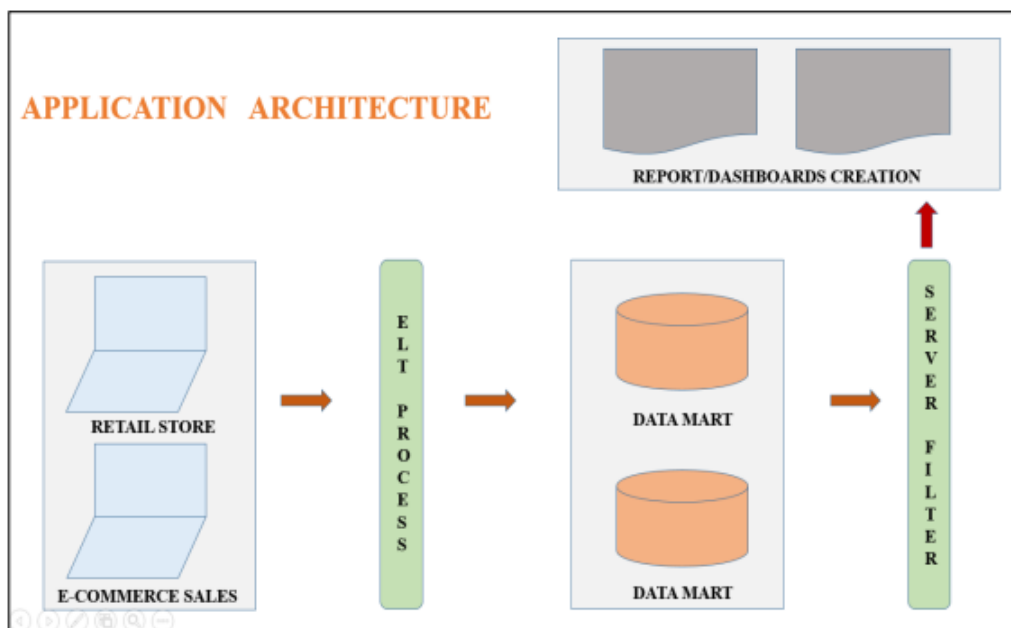


Figure 3.2: Application Architecture of Loss Prevention System

The accompanying focuses plot how the application information recovery is done in the undertaking also, demonstrates the work process of the application:

1. All the client exchange begins from base to up to change over in profitable report.
2. On the application design, the client exchange produced at Any Retail Store and E-business Inventory Sales.

3. All client related exchange are then exchanged to the nearby database of our framework through by means of way of distributed storage and with the procedure of ELT.
4. Transactions are then put away into information bazaar and every single related datum (information about information) will be put away into Application Meta Data.
5. Then, every one of the information will be exchanged to Intelligence Server which prompts take contributions from Data Mart and Application Meta Data.
6. Now, all organized information get put away into Schema Object and all qualities are put away into Metrics and afterward Intelligence Server applies some sort of Filters on information with the goal that significant data get gathered.
7. Filtered data at that point prompts age of reports/dashboards based on given past information. Dashboard additionally actualized here as it is nothing, it is only an accumulation of different reports.
8. Then, produced reports and dashboards are accessible for showcase at different gadgets for example Portable and Web

3.3 Log management and REST services

Log management is a part of the application architecture and flow for fraud detection and detailed anomaly analysis. Log management is a part of the application architecture and flow for fraud detection and detailed anomaly analysis[3].

Referring to the previous architecture flow,

- the data from the POS is transformed into a certain format and loaded onto the external ETL server. This data is imported into the import engines through the web services.
- This existing system can be integrated with other POS systems for e.g. the XStore system as stated in figure. This data further undergoes processing in the Import Management system in which log analysis is done using our application.

- REST services play an important role in this flow and the SSH File Transfer Protocol(SFTP) is used for secure file transfer.
- The log analysis is done using various data patterns and the anomalous log files are stored in the central cloud(UIStorage).

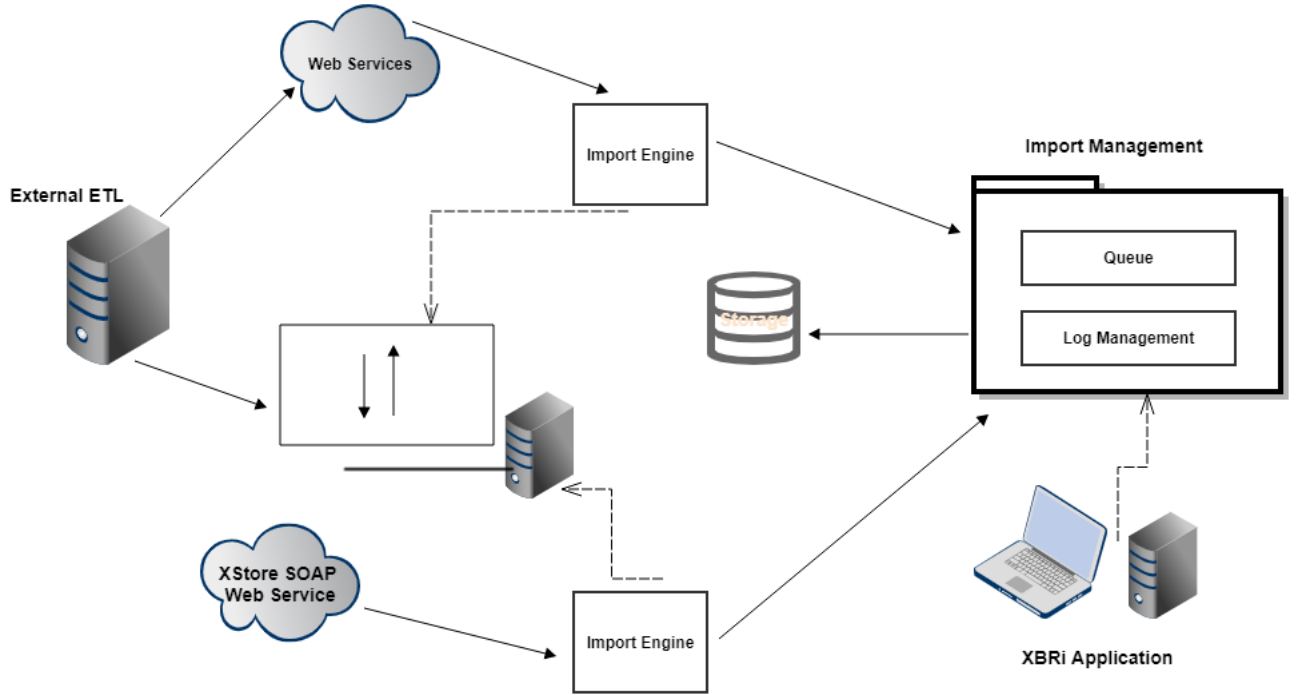


Figure 3.3: Log management through REST

3.4 ELT Data Flow

ELT is a technique which compresses data. It results into data extraction from various formats of data into usable and useful format and then results into data transformation and compression. Data transformation results into storing of data into metrics form and compression of data (only those values that are useful are kept, otherwise discarded).ELT has good performance only when the useful and extracted data machine is used in remote system, like a cloud installation, data appliance and Hadoop cluster ELT Data flow take major 4 steps for completion as follows:[4]

1. In this step, it will result into data extraction from the transactions done by client
2. Secondly, it will result into data delivery on cloud storage through a SFTP protocol.
3. The data stored in cloud storage is dumped into a local database of remote system.
4. The data transferred in above step now undergoes transformation to a usable format

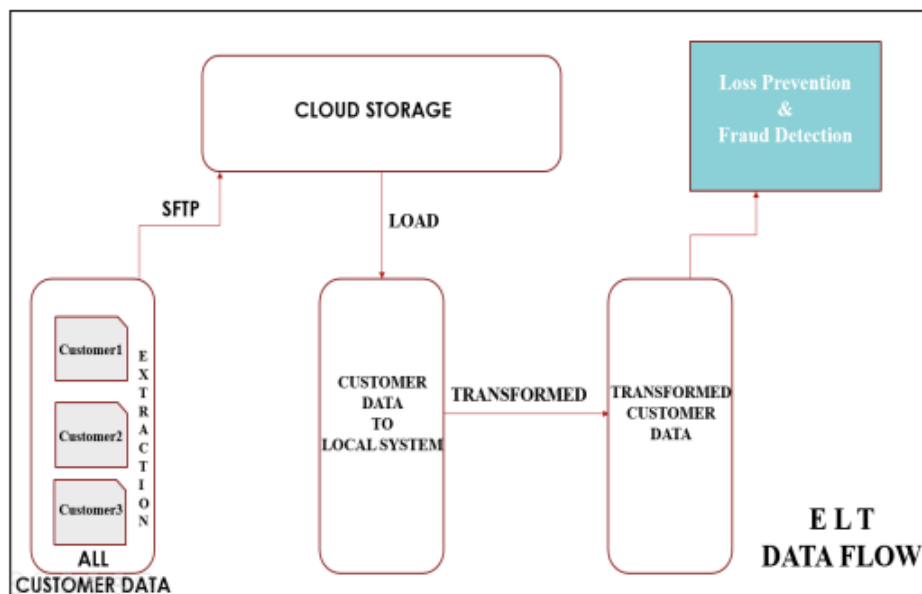


Figure 3.4: ELT flow of Data

Chapter 4

Work done

4.1 Log Maintenance

This module helps in different operations performed on logs:

- 1.Add
- 2.Delete
- 3.View
- 4.Download

There are 2 types of logs:-

- 1.File Logs
- 2.Table Logs

File Logs are mostly auditing files maintained for further purpose. Table Logs are maintained by high-end systems which undergo XTP(Extreme Transaction Processing).The figures ahead show a view of the module, how the module of log maintenance and analysis works

Figure 4.1 shows a remote view of the module.The logs can be categorized according to country and further according to state and cities as per the availability of systems. This system is managed by the security administrator. The figure shows the available systems in India by default. On clicking the **Show Logs** button, it will be directed to the Log Details module which will show the log details for that particular system.

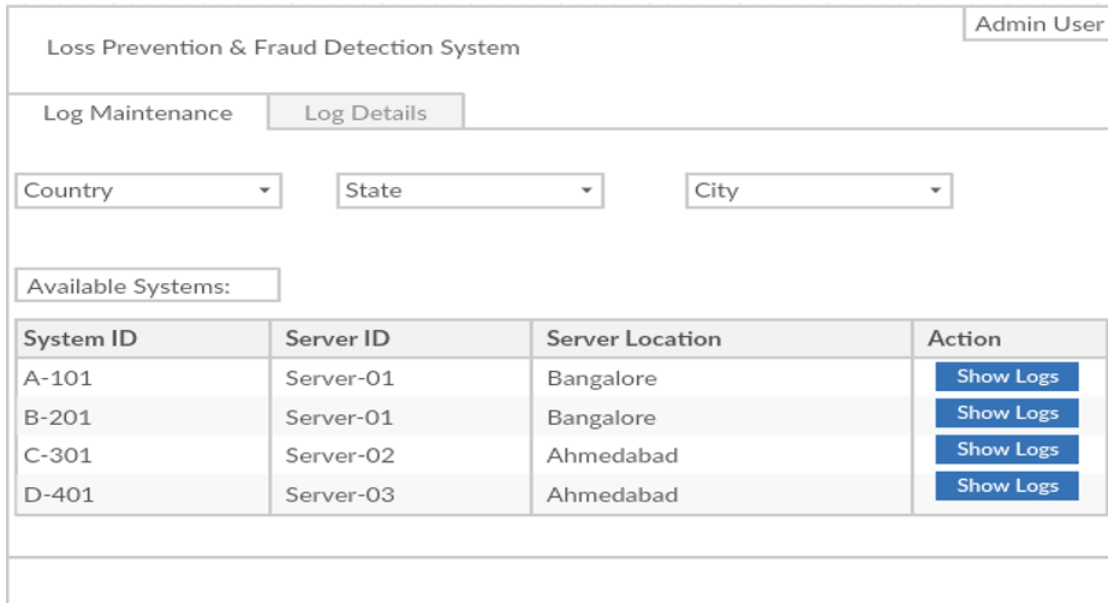


Figure 4.1: Remote view of module

Figure 4.2 shows how log maintenance actually works. There is an admin machine which can access all the systems that maintain the logs. On each POS system, this application is deployed for log maintenance. This system maintains and transfers logs to the admin machine (ETL server) for further log analysis. It maintains logs of the machines working under its accessibility.

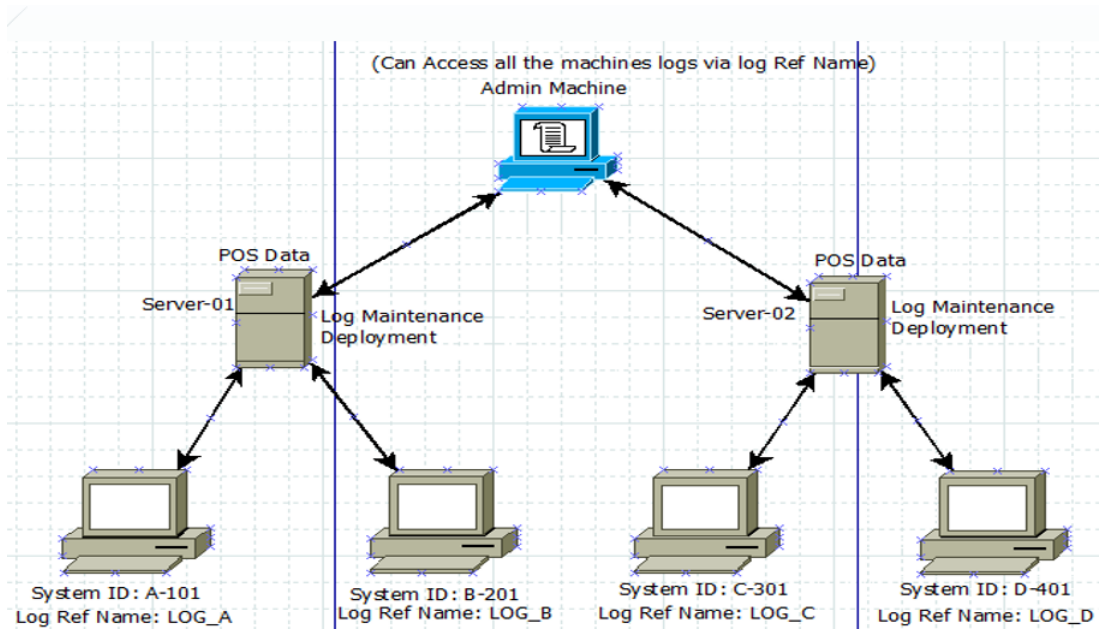


Figure 4.2: Log maintenance

Figure 4.3 shows the configuration of log maintenance module. This feature is avail-

able only in the central admin machine. The required fields are:

- 1.SystemID
- 2.ServerID
- 3.Log Reference Name
- 4.Log Description
- 5.Log Reference Location

The SystemID and ServerID together form the primary key for a particular system.

Loss Prevention & Fraud Detection System Admin User

Log Configuration

Add Log :

System ID :

Server ID :

Log Ref. Name :

Log Description :

Log Ref Location :

Figure 4.3: Configuring Log maintenance

Figure 4.4 shows the log details tab of the application. It displays the **system ID** as to which system it belongs, the **Log date** when the log was created, and the log files under that system according to the date. Selecting any log file and then clicking on the **Show Log** button displays the log. This application has a specific feature that if the selected log file to show is greater than 5 MB, it will display a warning dialog box as to inform the user that it will take time to display the file due to overhead and greater size of the file. Also the admin or the user has an option to download the log file among the following 3 formats:

- 1.PDF
- 2.Excel sheet
- 3.Word or text document

The download feature allows the admin to refer to the file in future instead of to run the application every time.

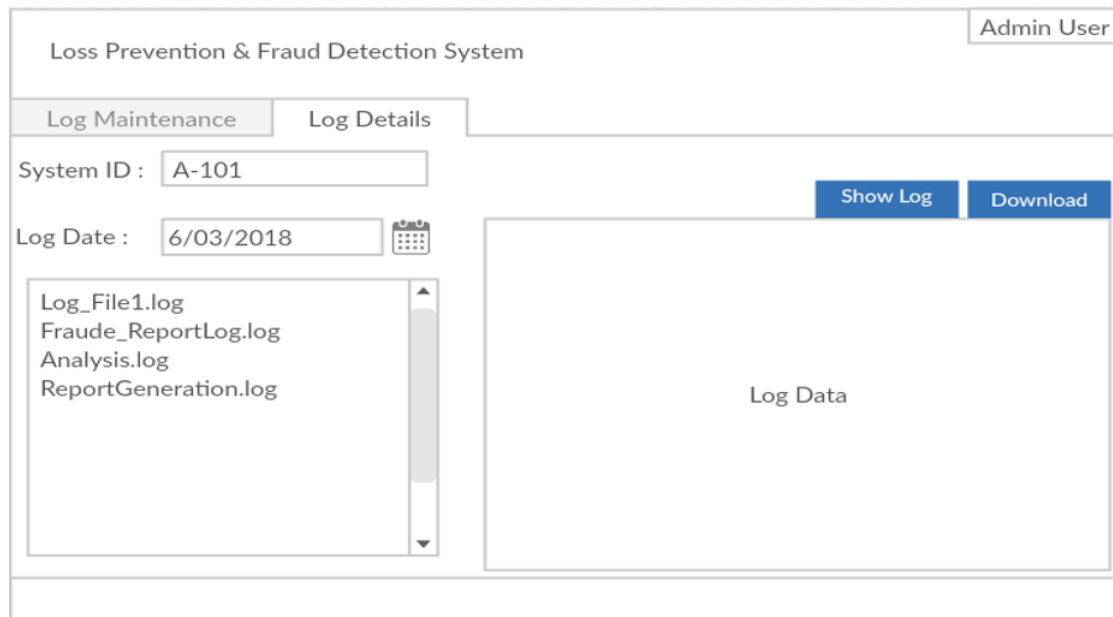


Figure 4.4: Detailed view of module

4.2 Loss Prevention and Fraud Detection Analysis

Figure 4.5 shows a model example of how fraud detection is done for industries on annual premises. It results into fraud detection based on some criteria i.e. extraction from the data given by client and then follows to give report results. Figure 4.6 shows some of the attributes of detection of fraud and its analysis. As shown in figure 4.6, some of the attributes of fraud detection are:

- web traffic over a system
- credit card tokens
- JSON Web token for verification
- violation of accessibility privileges
- number of transactions

Report generation can be done in 2 ways:

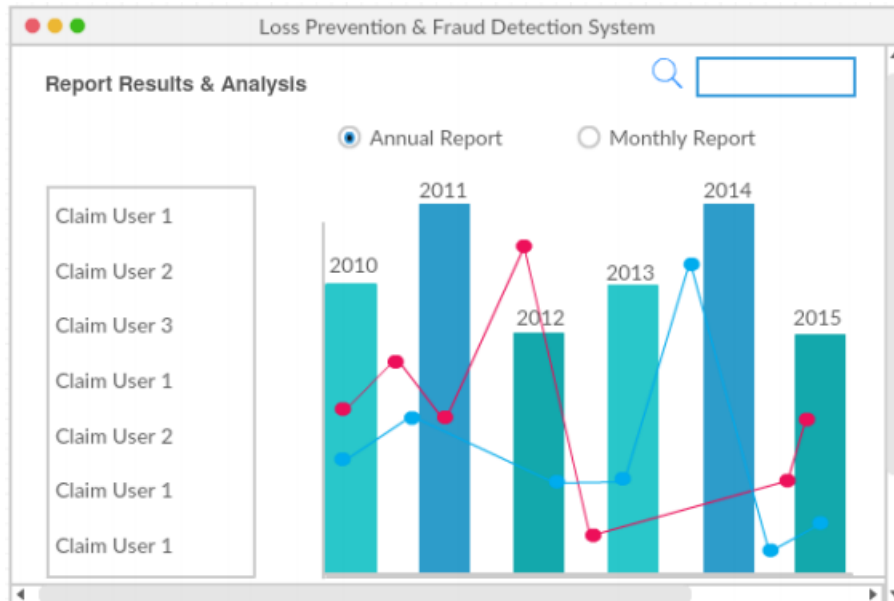


Figure 4.5: Dashboard for fraud detection analysis

- Monthly : On month to month premise framework can just recognize extortion just for those companies which have high deals as high client exchange is required for producing reports. Here and there for organizations which have lower deals, framework neglects to identify extortion on month to month premise
- Yearly : On yearly premise framework prompts produce reports for immense information exchanges. Thus, it will turn out to be extremely simple for framework to distinguish misrepresentation and misfortunes. Organizations which have less deals additionally get advantage by yearly revealing

The report generation provides an overview of the fraud detection analysis over a large scale. The analysis of minute details of fraud detection can be done only by the security administrator of the system. The reports are mailed to the respective clients weekly, monthly, every six months, or once a year as per the selection of the client.

The client is notified of a possible threat of fraud so that the client can take immediate countermeasures to protect from threat. In case of severe fraud detection and failure to notify the client about the danger, the client's system is automatically disabled with prior written agreements from the client.

Thus, report generation is an important aspect for fraud detection as it not only notifies of fraud but also analyzes in detail the pattern of fraud so that we can take countermeasures accordingly.

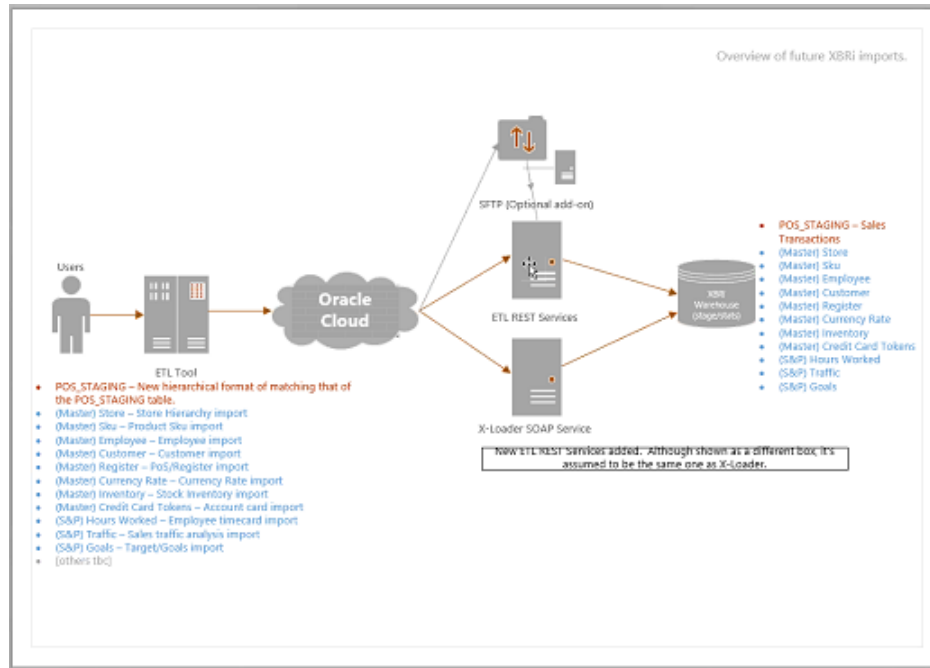


Figure 4.6: Fraud Detection analysis and attributes

4.3 Auditing framework for increased security

Due to ever increasing use of technologies in applications, the threat from cyber-attacks is significant and continuously evolving. Enterprises and organizations consider internal auditing for increased security as a third line of defense.

Application give the interface among Users and delicate information along these lines application's appropriate control on data stream are on top need. To shield those information from client or watch out for their activity we require an edge work in such a way so it can screen all required activity from user. Purpose of the information reviewing is utilized to catch the occasion or on the other hand task on the application information, it includes profiling of client information. A few information of the application are delicate and confined to the administrator clients as it were. It is helpful in future to follow the adjustment and access of private information

There are two ways to audit an application:

- Data storage in a simple file format
- Data storage in a remote database

In this project, the data that is audited are stored in a remote database rather than a simple file. There are many definitions that are required to be specified in our local database to store the data that is audited. Parameters are mentioned below:

- UserId : Always unique, depicts which user was using or performing operations.
- Date Time : The exact time at which the user was using the application.
- Access Data : A separate log file is formed, which consists of the data the user created, read or edited in a modified format.
- Complexity/Severity : Among 3 options:
 - Low - Read
 - Medium edit, read(for specific type of data)
 - High-(create,edit,delete)
- Operations - CRUD(create, read, update, delete)
- Modules
- Name of module being used. E.g. Log Maintenance

Below is the pictorial representation of flow and different parameters of audit data:

4.3.1 Auditing framework implementation

REST services are used for this framework implementation. Many parallel requests will start for serving output for requested resource.

- Select one application and start its execution from App UI
- Request will start for the module application resource requested.

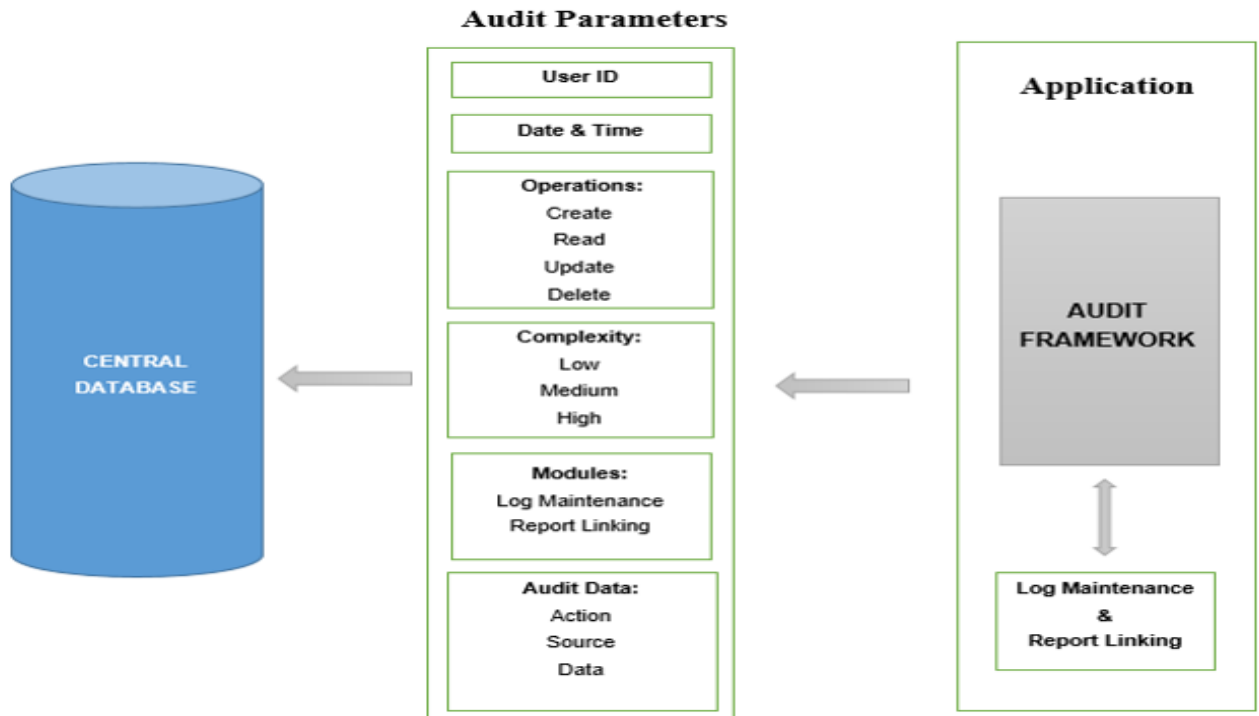


Figure 4.7: Audit Parameters

- State parameters which are compulsory to start request and REST request will reach to the application server.
- Before displaying the output to App UI, auditing framework is accessed for auditing.
- After the data is audited, the corresponding output will be displayed to User for the resource requested.

Figure 4.8 shows the flow diagram of audit data framework:

The following snap shows how auditing is implemented for Log Maintenance module. Various REST services are used for such implementation. REST annotations as below are used for the implementation:

1. GET - to get a resource object for e.g. to get an existing node object or log object.
2. PUT - to update an existing resource object
3. POST - to create a new resource object for e.g. creating a new node object, creating a new log file entry and so on
4. DELETE - to delete an existing log file entry

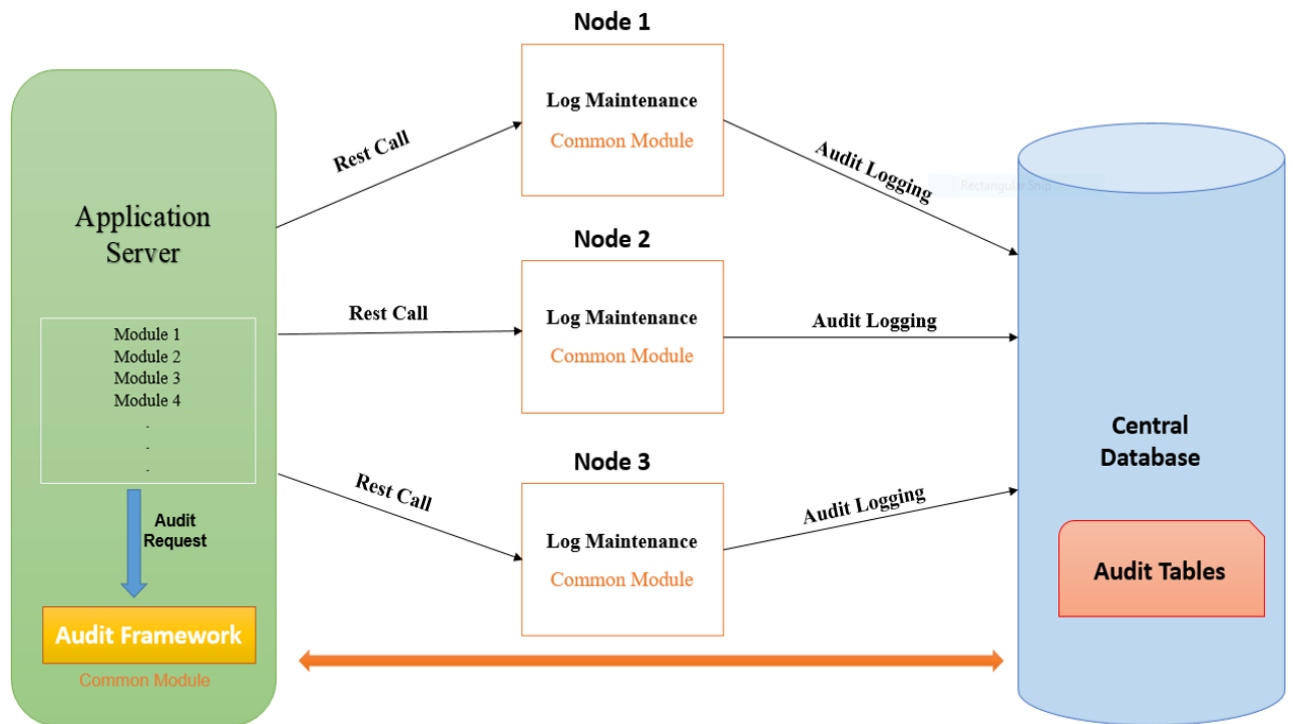


Figure 4.8: Audit Data Flow

No	Scenario	Webservices	Severity	Category	Module
1	Create Node	addNodeName	Medium	Create	v1/nodeManager
2	Delete Node	deleteNodeName	Medium	Delete	v1/nodeManager
3	Create Log Entry (Type: File)	addLogFileData	Medium	Create	v1/logviewer
4	Create Log Entry (Type : Table)	addLogTableData	Medium	Create	v1/logviewer
5	Update Log Entry (Type: File)	addLogFileData	Medium	Update	v1/logviewer
6	Update Log Entry (Type : Table)	addLogTableData	Medium	Update	v1/logviewer
7	Delete Log Entry (Type: File)	deleteFileLog	Medium	Delete	v1/logviewer
8	Delete Log Entry (Type : Table)	deleteTableLog	Medium	Delete	v1/logviewer
9	Load LogEntry (Type: File)	getFileLogData	Medium	Read	v1/logviewer
10	Load LogEntry (Type : Table)	getTableLogData	Medium	Read	v1/logviewer
11	Download LogEntry (Type: File)	getTableLogData	High	Read	v1/logviewer
12	Download LogEntry (Type : Table)	getTableLogData	High	Read	v1/logviewer

Figure 4.9: Audit Framework Implementation

4.3.2 Audit Database

Based on the REST services implemented above, an audit database is implemented which shows the various actions performed by each user along with the timestamps and the type of action performed.

Figure 4.10 shows the database schema for the implementation of auditing for Log Maintenance module.

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	USER_ID	VARCHAR2 (128 BYTE)	No	(null)	1	(null)
2	LOG_DATE	DATE	No	(null)	2	(null)
3	LOG_TIME	VARCHAR2 (20 BYTE)	No	(null)	3	(null)
4	CATEGORIES	VARCHAR2 (20 BYTE)	No	(null)	4	(null)
5	SEVERITY	VARCHAR2 (20 BYTE)	No	(null)	5	(null)
6	MODULE	VARCHAR2 (20 BYTE)	No	(null)	6	(null)
7	LOG_DATA	CLOB	Yes	(null)	7	(null)

Figure 4.10: Audit Database Schema

Figure 4.11 shows how the audit log data is stored into the database.

USER_ID	LOG_DATE	LOG_TIME	CATEGORIES	SEVERITY	MODULE	LOG_DATA
ajuthani	12-OCT-18	06:14:22	delete	high	v1/logviewer	{ "logName": "scu1", "logType": "file", "groupName": "scu1", ... }
MUKOTHAN	12-OCT-18	06:14:22	read	high	v1/logviewer	{ "fileName": "temp_5mbfile.log", "logName": "Xbro_demo", ... }
MUKOTHAN	12-OCT-18	06:14:23	read	medium	v1/logviewer	{ "fileName": "temp_5mbfile.log", "logName": "Xbro_demo", ... }
atugugup	29-JUN-18	06:32:07	create	medium	v1/logviewer	{ "logName": "POSLOG Pending Queue Import", "logType": "f...", ... }
ajuthani	11-JUL-18	12:06:38	read	medium	v1/logviewer	{ "fileName": "PRO_EVENTLOG", "logName": "log name", "logT...", ... }
ajuthani	18-JUL-18	11:20:02	read	medium	v1/logviewer	{ "fileName": "PRO_EVENTLOG", "logName": "Xbro Log Table", ... }
ajuthani	18-JUL-18	11:21:08	read	high	v1/logviewer	{ "fileName": "PRO_EVENTLOG", "logName": "Testing Shivani...", ... }
ajuthani	30-AUG-18	06:13:31	read	medium	v1/logviewer	{ "fileName": "temp_5mbfile.log", "logName": "Xbro_demo", ... }
ajuthani	30-AUG-18	06:13:38	read	medium	v1/logviewer	{ "fileName": "temp_5mbfile.log", "logName": "Xbro_demo", ... }
ajuthani	30-AUG-18	06:19:15	read	medium	v1/logviewer	{ "fileName": "Timesheet Entry.log", "logName": "Xbro_dem...", ... }
ajuthani	30-AUG-18	06:24:34	read	medium	v1/logviewer	{ "fileName": "Timesheet Entry.log", "logName": "Xbro_dem...", ... }
ajuthani	30-AUG-18	06:28:08	read	medium	v1/logviewer	{ "fileName": "Timesheet Entry.log", "logName": "Xbro_dem...", ... }
ajuthani	30-AUG-18	06:30:11	read	medium	v1/logviewer	{ "fileName": "Timesheet Entry.log", "logName": "Xbro_dem...", ... }
ajuthani	30-AUG-18	06:31:55	read	medium	v1/logviewer	{ "fileName": "Timesheet Entry.log", "logName": "Xbro_dem...", ... }
ajuthani	30-AUG-18	06:33:49	read	medium	v1/logviewer	{ "fileName": "Timesheet Entry.log", "logName": "Xbro_dem...", ... }
ajuthani	30-AUG-18	06:36:10	read	medium	v1/logviewer	{ "fileName": "Timesheet Entry.log", "logName": "Xbro_dem...", ... }
ajuthani	30-AUG-18	06:40:46	read	medium	v1/logviewer	{ "fileName": "Timesheet Entry.log", "logName": "Xbro_dem...", ... }
ajuthani	30-AUG-18	06:40:56	read	medium	v1/logviewer	{ "fileName": "temp.log", "logName": "Xbro_demo", "logType": "...", ... }
ajuthani	30-AUG-18	06:41:03	read	medium	v1/logviewer	{ "fileName": "temp - Copy.log", "logName": "Xbro_demo", "...", ... }
ajuthani	30-AUG-18	06:41:06	read	medium	v1/logviewer	{ "fileName": "temp - Copy (6).log", "logName": "Xbro_dem...", ... }
ajuthani	30-AUG-18	06:41:10	read	medium	v1/logviewer	{ "fileName": "temp - Copy.log", "logName": "Xbro_demo", "...", ... }
ajuthani	30-AUG-18	06:41:28	read	medium	v1/logviewer	{ "fileName": "temp_5mbfile.log", "logName": "Xbro_demo", "...", ... }
ajuthani	30-AUG-18	06:41:40	read	medium	v1/logviewer	{ "fileName": "temp - Copy.log", "logName": "Xbro_demo", "...", ... }
atugugup	29-JUN-18	06:29:48	create	medium	v1/logviewer	{ "logName": "POSLOG Failure Import File", "logType": "fi...", ... }
atugugup	29-JUN-18	06:29:52	create	medium	v1/logviewer	{ "logName": "POSLOG archive Import File", "logType": "fi...", ... }
atugugup	29-JUN-18	06:30:02	create	medium	v1/logviewer	{ "logName": "POSLOG Pending Queue Import", "logType": "F...", ... }

Figure 4.11: Audit Database Data

4.4 Scripts using cURL for secure authorization

- cURL stands for Client URL Request Library. It is a tool to transfer data from or to a server, using one of the supported protocols (HTTP, HTTPS, FTP, FTPS, SCP, SFTP, TFTP, DICT, TELNET, LDAP or FILE). The command is designed to work without user interaction. It offers proxy support, user authentication, FTP uploading, HTTP posting, SSL connections, cookies, file transfer resume, etc.

The main objective of selecting cURL scripts was:

- Faster and automatic operations
- Less overhead
- Better authorization token verification

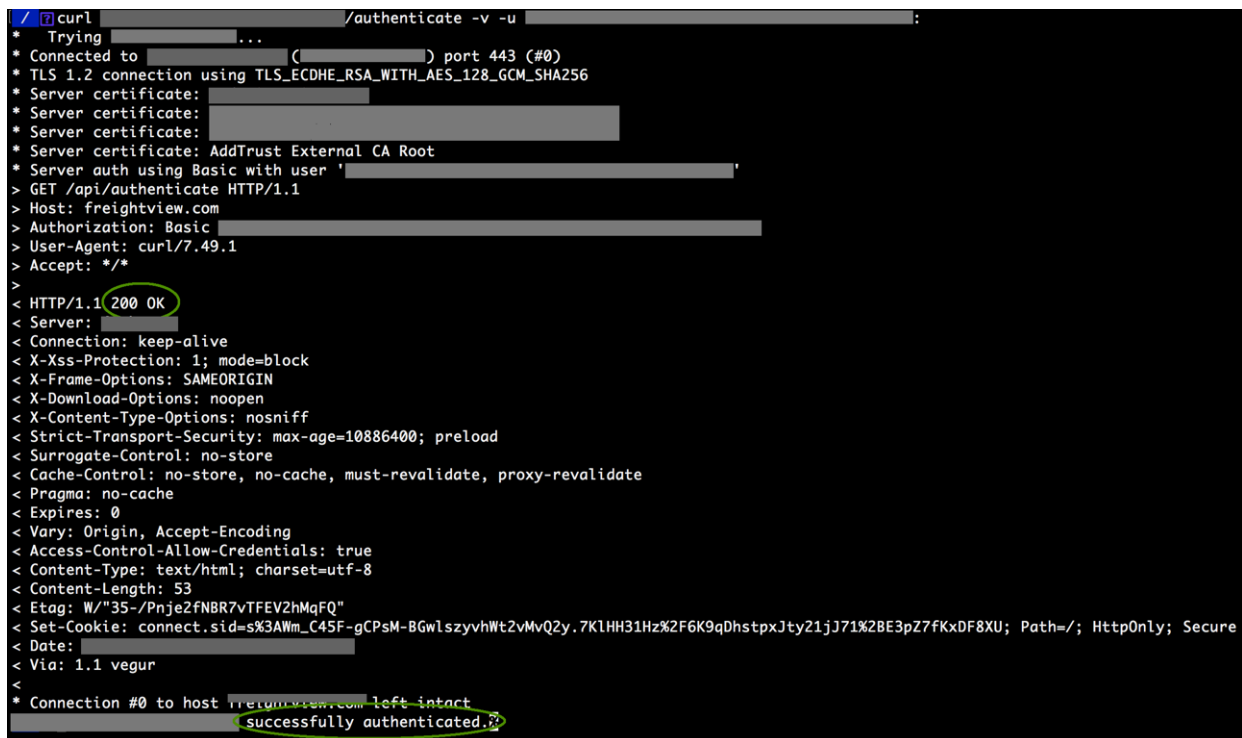
4.4.1 cURL Implementation

```
curl -f -H $AUTHORISATION -H $CONTENT_TYPE request POST $URL -d $GRANT_TYPE $USERNAME & $PASSWORD & $SCOPE
```

where

- AUTHORISATION:- token for authorization which will change every time
- CONTENT_TYPE:- application/json, xml, plaintext html, etc

Figure 4.12 shows the cURL implementation for authorization.



```
curl [redacted] /authenticate -v -u [redacted] :
* Trying [redacted] ...
* Connected to [redacted] ([redacted]) port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
* Server certificate: [redacted]
* Server certificate: [redacted]
* Server certificate: [redacted]
* Server certificate: AddTrust External CA Root
* Server auth using Basic with user "[redacted]"
> GET /api/authenticate HTTP/1.1
> Host: freightview.com
> Authorization: Basic [redacted]
> User-Agent: curl/7.49.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: [redacted]
< Connection: keep-alive
< X-Xss-Protection: 1; mode=block
< X-Frame-Options: SAMEORIGIN
< X-Download-Options: noopen
< X-Content-Type-Options: nosniff
< Strict-Transport-Security: max-age=10886400; preload
< Surrogate-Control: no-store
< Cache-Control: no-store, no-cache, must-revalidate, proxy-revalidate
< Pragma: no-cache
< Expires: 0
< Vary: Origin, Accept-Encoding
< Access-Control-Allow-Credentials: true
< Content-Type: text/html; charset=utf-8
< Content-Length: 53
< Etag: W/"35-/Pnje2fNBR7vTFEV2hMqFQ"
< Set-Cookie: connect.sid=s%3AWm_C45F-gCPsM-BGwlszyvhWt2vMvQ2y.7K1HH31Hz%2F6K9qDhstpxJty21jJ71%2BE3pZ7fKxDF8XU; Path=/; HttpOnly; Secure
< Date: [redacted]
< Via: 1.1 vegur
<
* Connection #0 to host [redacted] left intact
[redacted] successfully authenticated.?
```

Figure 4.12: cURL Implementation

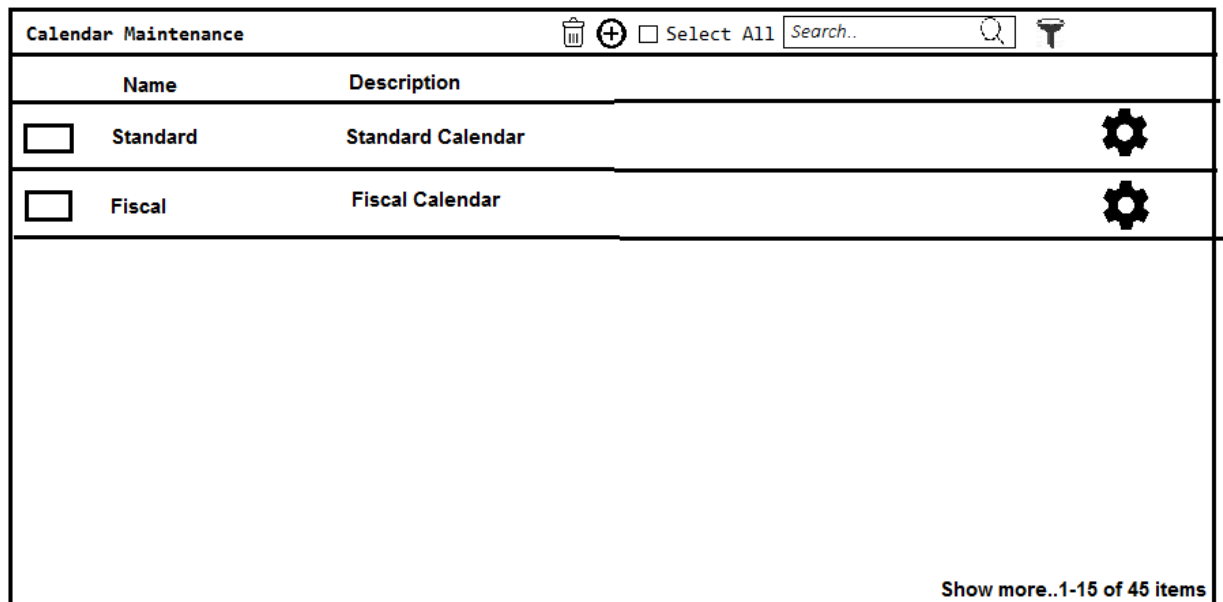
4.5 Calendar Maintenance and Configuration







4.5.1 Calendar Maintenance

The modules of Calendar Maintenance and configuration are developed in consequence to the limitations of Report Generation module for fraud detection analysis. In the Report Generation Module, the reports are generated for detailed analysis and fraud detection based on this analysis. The limitation in this module was that the user had to load all the reports for all types of calendars(Standard, Fiscal, Custom, and Rolling) to select a particular report for a particular type of calendar.

The user in the above scenario couldn't select the reports based on time. For e.g. if the user wanted the reports for only last month, he couldn't do so in the above scenario. So, I worked on the Calendar Maintenance and Calendar Configuration modules which will further be integrated with the Report Generation module which will enable the user to select reports based on the already defined dates added into the database.

Figure 4.13 shows a snap of the entry point of the calendar maintenance module.



Calendar Maintenance		 	<input type="checkbox"/> Select All	<input type="text" value="Search.."/>		
Name	Description					
<input type="checkbox"/> Standard	Standard Calendar					
<input type="checkbox"/> Fiscal	Fiscal Calendar					

Show more..1-15 of 45 items

Figure 4.13: Calendar Maintenance

As shown in figure 4.13, the list of calendars are displayed on the index screen. The functionalities provided on this screen are :

- Delete icon- deletes one or more selected calendars
- Add icon- navigates to create calendar page
- Select All checkbox - allows to select all calendars at once for a particular operation.
- Search- search a particular calendar which is case insensitive
- Filter- Sorting based on Ascending order and Descending order.
- Settings button- For each calendar, displays a menu which shows-
 1. Edit
 2. Delete-Read only
 3. View - Read only

On the index screen, only the calendar name and its description are shown. In the edit mode, the detailed view is shown. By doubleclicking on the calendar on the index screen, it is navigated to the edit mode.

The UI designs for the edit, view and delete is the same but different models are created based on the operation and functionality. Various design patterns were used for designing the framework too.

Figure 4.14 shows the add calendar UI design.

In the creation of the calendar, the fields are:

- Name of calendar(Primary Key)
- Description
- Type of calendar-4 options are there:
 - Standard
 - Fiscal
 - Rolling
 - Custom

Create Calendar

Save
 Cancel

Name*

Description

Type*

Standard
 Fiscal
 Rolling
 Custom

Dates

Name	Type	Selection	Start	End	Action

Figure 4.14: Add Calendar

- Dates- The fields are:
 - Name - unique for a given calendar
 - Type - Options like:
 - * DAY_ABSOLUTE
 - * DAY_RELATIVE
 - * WEEK_ABSOLUTE
 - * WEEK_RELATIVE
 - * MONTH_ABSOLUTE
 - * MONTH_RELATIVE
 - * QUARTER_ABSOLUTE
 - * QUARTER_RELATIVE

The name and type of calendar fields are compulsory to save a calendar. In the dates types options, there are two types- Absolute and Relative. Absolute relates to

the beginning of the current year and the dates are calculated for positive values only. Relative relates to the current date and the dates are calculated based on positive as well as negative values. So for the past calculations, Relative is used.

For the edit calendar mode, the name of the calendar field as well as the type of calendar dropdown is disabled. The only fields editable are the dates for which I have used composite components. Each date has a edit and delete button with it. The delete button works such that even if there is one date added for a particular type of calendar, the calendar type dropdown will be automatically disabled. Only when all the dates are deleted or haven't been added, the type dropdown will be enabled.

4.5.2 Calendar Configuration

The Calendar Configuration module is mainly used for setting the fiscal calendar configuration values by the admin user. These values include:

- Fiscal year
- Fiscal start date
- Fiscal end date
- Number of weeks in the fiscal year
- Next known 53 week year
- Start day of week
- Calendar type for fiscal

Figure 4.15 shows the Calendar Configuration design.

The values for configuration are stored in cloud storage. These values are then integrated into the Calendar Maintenance module for the fiscal year date calculations. The values used are :

- fiscal year
- fiscal year start date
- fiscal calendar type

Calendar Configuration		Save	Cancel								
Fiscal Calendar											
year	<table border="1"> <tr><td>2013</td></tr> <tr><td>2014</td></tr> <tr><td>2015</td></tr> <tr><td>2016</td></tr> </table>	2013	2014	2015	2016	calendar type	<table border="1"> <tr><td>4-5-4</td></tr> <tr><td>4-4-5</td></tr> <tr><td>5-4-4</td></tr> <tr><td>4-4-4</td></tr> </table>	4-5-4	4-4-5	5-4-4	4-4-4
2013											
2014											
2015											
2016											
4-5-4											
4-4-5											
5-4-4											
4-4-4											
start date	<input type="text"/>	start day of week	<table border="1"> <tr><td>Sunday</td></tr> <tr><td>Monday</td></tr> <tr><td>Tuesday</td></tr> <tr><td>Wednesday</td></tr> </table>	Sunday	Monday	Tuesday	Wednesday				
Sunday											
Monday											
Tuesday											
Wednesday											
end date	<input type="text"/>										
number of weeks	<input type="text"/>										
next known 53 week	<input type="text"/>										

Figure 4.15: Calendar Configuration

- fiscal cal start day of week

These values are integrated for date calculations into the dates pop-up UI.

Chapter 5

Challenges faced

- The development of UI using Oracle JET was a big challenge as it was completely new for me. Different Oracle JET components had to be used and understanding and using their properties according to the requirements.
- Some of the UI requirements were such that the existing components cannot be used. So we had to develop composite components as per the need. The designing and coding for that was a challenge in itself.
- Using REST services with different annotations for retrieving the resource objects was a challenge as it required encoding of the URLs which was a challenging task.
- Integration of the two modules-Calendar Maintenance and Calendar Configuration was a difficult task as REST calls had to be made single-handedly for both of them. Retrieving the resource objects and using them into the UI using different data-structures was a challenging task.

Chapter 6

Future Scope

- The future scope is based on the shortcomings and limitations of the existing application. The Log maintenance module needs to be more performance-tested and developed accordingly.
- It is too slow when the user loads it for the first time as it also loads all the log files along with it. If the user tries to load a large log file with size greater than 100 MB, then the application crashes sometimes.
- So, instead of storing the log files on a standalone server, the log files can be stored on a cloud which will not only be more lightweight but also more secure and more efficient.
- The Calendar Maintenance module will need more enhancements like supporting different types of calendars for fiscal calendar like 4-4-4, 4-5-5, etc. Each calendar created will have support for these type of calendars as per the need of the user.

Bibliography

- [1] J. Hillmer, R. Jones, C. Gessner, C. Johnston, K. Lewis, and S. Deshpande, System and method for detecting fraudulent transactions,” Mar. 30 2004. US Patent 6,714,918
- [2] A. Kukic, Cookies vs. tokens: The definitive guide,” 2016. US Patent
- [3] B. Goodman, Loss prevention and sales productivity cloud services,” 2017. US Patent
- [4] Tobias Wchner, Alexander Pretschner, ”Data Loss Prevention Based on DataDriven Usage Control”, Software Reliability Engineering (ISSRE) 2012 IEEE 23rd International Symposium on, pp. 151-160, 2012.
- [5] Bell, T. & Carcello, J. (2000). A Decision Aid for Assessing the Likelihood of Fraudulent Financial Reporting. *Auditing: A Journal of Practice and Theory* 10(1): 271-309.
- [6] Bolton, R. & Hand, D. (2002). Statistical Fraud Detection: A Review (With Discussion). *Statistical Science* 17
- [7] Shavlik, J. & Shavlik, M. (2004). Selection, Combination, and Evaluation of Effective Software Sensors for Detecting Abnormal Computer Usage. *Proc. of SIGKDD04*