FPT Education

TRƯỜNG ĐẠI HỌC FPT

Capstone Project Report

Final Report

May, 2025

Topic: Developing Alert and Monitoring Functions for Suspicious Activities on SAP System

Student name (ID):

Khuat Duy Hieu - HE171387

Le Viet Anh - HE172133

Nguyen Minh Hieu - HE161850

Nguyen Hoai Nam - HE160187

Nguyen Phuc Long - HE170623

Supervisors:

Nguyen Xuan Nui - Mentor FPT

Bui Le Minh - Mentor FPT

Do Thanh Loc - Mentor FSOFT

I. Introduction **4**
1. General Information **4**
1.1 Potential Consequences of Solely Using SM20: 5
1.2 Project Motivation 6
2. The Team 6
3. Organization Structure 8
4. Scope of Work 10
4.1 In-Scope Activities 10
4.2 Out of Scope 12
4.2.1 Out of Project Scope 12
4.2.2 Out of Product Scope 12
5. Final Report Structure 13
II. Product Statements **13**
1. Product Analysis 13
1.2 How the App Solves Existing Challenges 14
1.3 Key Features 14
1.3.1. Dynamic Audit Log Viewer 15
1.3.2. Integrated Case Management System 15
1.3.3. Real-Time Dashboard Visualizations 15
1.3.4. External Service Integration (via Node.js API) 15
1.4 Conclusion 16
2. As-Is 16
2.1 Manage SM20 (Security Audit Log Viewer): 16
2.2 Limitations of Current Features: 17
2.3 Send emails by SOST (SAPconnect Send Requests): 17
2.4 Limitations of Current Features: 18
3. To be (After Project Implementation) 18
3.1. Production 18
3.1.1. Logs Analyzer – SAP System Log Analysis 19
Key Features: 19
1. Main Interface 19
2. Advanced Filtering Options 20
3. Log Detail View 20
4. Case Creation 20
5. Export Functionality 21
Benefits: 21
3.3.2 Cases Management – Centralized Security Case Handling 22
Key Features and Functionality: 22
1. Case Overview Dashboard 22
2. Detailed Case View 23
3. Comments and Activity History 24
4. Filtering and Search Capabilities 24
Benefits: 24
3.3.3 Emails – Security Notifications via External API 24

Key Features and Functionality 25
Benefits 26
3.3.4 Dashboard – Centralized Security Intelligence and Monitoring 26
Key Features and Functionality 26
Benefits 29
4. Project Management 29
4.1. Training Plan – Listing and Purpose 29
4.1.1 Blueprint (converted to To-Be Process): 29
4.1.2 Configuration Notes: 29
4.1.3 Functional Specification: 30
4.1.4 Technical Specification: 30
4.1.5 Unit Test Evidence: 30
4.1.6 Test Scenario: 30
4.1.7 User Manual: 31
4.1.8 UAT (Acceptance Test) Sign-off: 31
4.1.9 Thesis Defense: 31
4.2. Project Management Plan 31
Key Phases and Dates: 33
III. SAP Configurations **34**
1. Enterprise Structure Definition 34
2 Business Process Configuration 34
2.1 Flag Management for Checking Algorithms 34
2.2 Threshold Configuration for Brute-force Attack Diagnostics 34
2.3 External Base API URL Management 34
3. Suspicious Events Diagnostic Algorithms 34
3.1 Brute-force Attacking 34
3.2 User logon from new IP Address 35
3.3 User locked on client after numerous tries 37
4 Master Data 41
5 Technical Configuration via TCode ZG1CFG 41
5.1 Custom Configuration Table (Z-Table) 41
5.2 Custom Transaction Code (TCode ZG1CFG) 41
IV. SAP Coding **42**
1. Application Architecture 42
2. Functional/Technical Specifications 43
2.1 Logs Analyzer 43
2.1.1 Logs Analyzer 43
2.1.2 Audit Log Detail 44
2.1.3 Audit Logs Create Case 46
2.2 Cases Managements 47
2.2.1 Audit Logs Cases Management 47
2.2.2 Audit Logs Case Detail 48
2.3 Dashboard 49
2.3.1 System Security Dashboard 49

2.4 Emails 50
2.4.1 Audit Logs Send Notifications 50
3. Reports 52
4. Deployment 56
4.1. SAP Fiori 56
4.2. SAP ABAP 57
4.3. NodeJS Backend 58
V. Notification System **59**
1. API Specifications for Sending Notifications 59
1.1 New Case Notification 59
1.2. Case Status Change 62
1.3. Change Assignee 64
1.4. New Comment 66
1.5. Weekly report 68
1.6. Month report 68
2. Sending Notification via API 69
Node.js Email Sending Example: 69
3. Expected Notifications Result 71
Testing with Postman 81
VI. Implementation & Testing **82**
1. Implementation 82
1.1. SAP Process Execution 82
1.2. Installation Guides 85
1.3. User Manual 91
2. Testing 92
2.1. Unit Test 92
Test Results Report 92
2.2. System Test 93
System Test Summary 93
VII. Appendix **94**

I. Introduction

1. General Information

In modern enterprise environments, SAP systems serve as the digital backbone for critical business operations, including finance, logistics, procurement, and human resources. Due to the sensitive nature of the data and processes they manage, SAP platforms are increasingly becoming attractive targets for both external cyber threats and insider attacks.

Security monitoring within SAP traditionally relies heavily on the Security Audit Log Viewer (transaction SM20), which captures system activities such as login attempts, authorization checks, and critical configuration changes. However, using SM20 alone for security log management introduces significant risks and limitations:

- Delayed Incident Detection:
  SM20 is a manual log viewer; it requires periodic human inspection. Suspicious activities, such as brute-force login attempts or privilege escalations, may remain undetected for hours or days, giving attackers a dangerous time advantage.
- Data Overload and Noise:
  Audit logs in SM20 often contain huge volumes of routine or low-risk entries, making it difficult to quickly identify critical security threats without intelligent filtering or prioritization.
- Lack of Real-Time Alerts:
  SM20 does not generate real-time notifications. Without automated alerts, security teams depend on scheduled reviews, leading to delayed responses and increased risk of damage.
- No Structured Incident Management:
  The native SAP interface does not provide case creation, tracking, or collaboration workflows, resulting in fragmented and inefficient incident response processes.
- Limited Integration Capabilities:
  SM20 logs are isolated within the SAP GUI environment, making it difficult to integrate log information with external systems such as SIEM tools (e.g., Splunk, QRadar) or modern collaboration platforms (e.g., Microsoft Teams, email systems).

1.1 Potential Consequences of Solely Using SM20:

If SAP security monitoring relies exclusively on SM20 without enhancements:

- Threats may go undetected, allowing unauthorized access, data theft, or financial fraud.
- Regulatory violations could occur, exposing the organization to fines (e.g., GDPR, SOX non-compliance).
- Operational disruptions may happen due to delayed threat response, leading to system downtime or data corruption.
- Increased workload and burnout among security analysts, who must manually sift through irrelevant log data.

- Damage to company reputation in the event of publicized breaches caused by slow detection and response.

1.2 Project Motivation

Recognizing these challenges, our project — "Developing Alert and Monitoring Functions for Suspicious Activities on SAP Systems" — was initiated to enhance SAP system security posture. By leveraging SAP SM20 logs as the foundation and layering on real-time monitoring, automated detection, external integrations, and a user-friendly Fiori dashboard, the project aims to transform SAP security monitoring into a proactive, responsive, and intelligent defense mechanism.

2. The Team

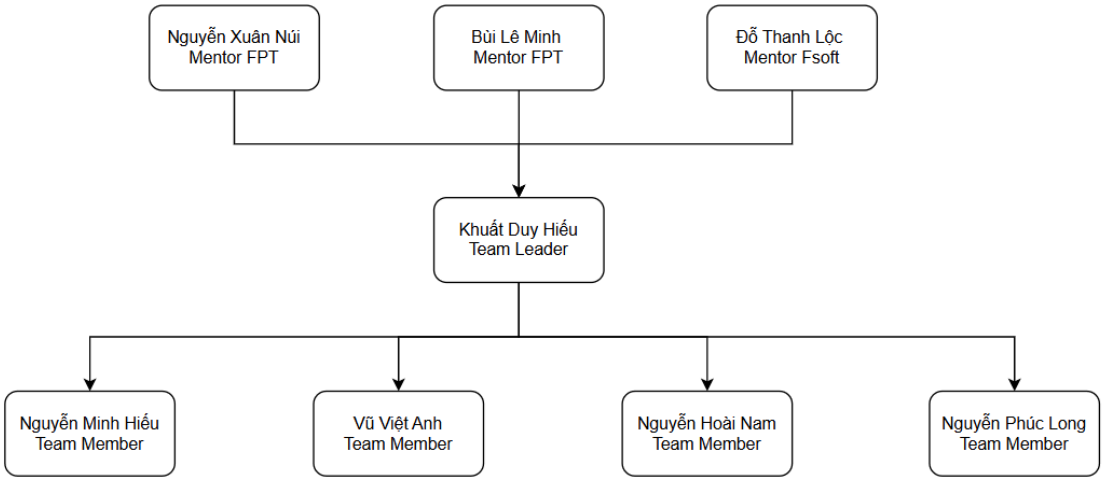
The project's success is a result of teamwork, expertise, and close collaboration between functional and technical teams. The project team includes SAP security experts, developers, and testers responsible for configuration, coding, testing, and documentation.

Mentors:

- **Nguyễn Xuân Núi** – Mentor FPT
- **Bùi Lê Minh** – Mentor FPT
- **Đỗ Thanh Lộc** – Mentor FSOFT

Team Members:

- **Khuất Duy Hiếu** – Team Leader
- **Nguyễn Phúc Long** – Team Member
- **Nguyễn Minh Hiếu** – Team Member
- **Nguyễn Hoài Nam** – Team Member
- **Lê Việt Anh** – Team Member



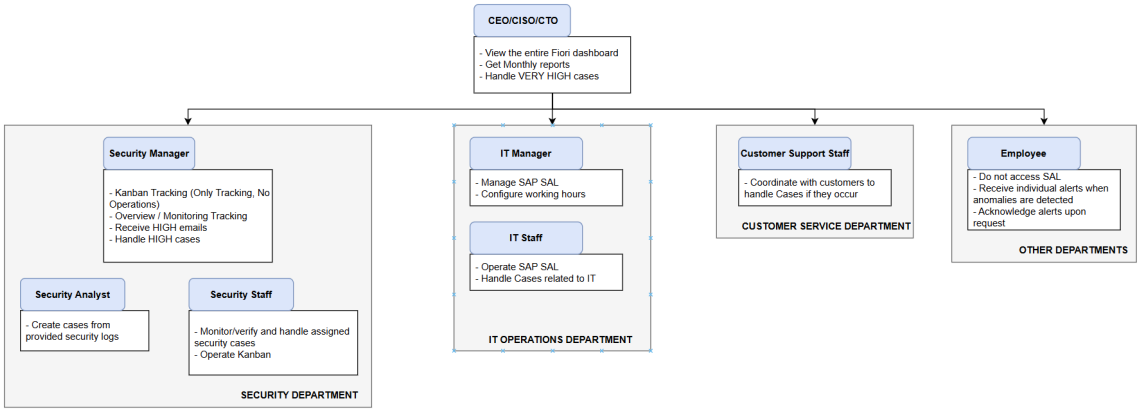
D~Do; R~Review; S~Support; I~Informed; <blank>- Omitted

Responsibility	**HieuKD HE171387**	**HieuNM HE161850**	**LongNP HE170623**	**AnhLV HE172133**	**NamNH HE160187**
Project Planning & Tracking	**D**	R	R	R	R
Business Requirement Workshop	**S**	**S**	**D**	**R**	**R**
Tobe Process	**R**	**S**	**D**	**I**	**I**
Functional Specification	**R**	**S**	**D**	**R**	**R**
Configuration	**D**	**I**	**R**	**I**	**I**

Technical Specification	**S**	**S**	**D**	**R**	**R**
Coding	**D**	**D**	**D**	**D**	**D**
User Manual	**R**	**R**	**D**	**I**	**I**
System Testing	**R**	**D**	**R**	**R**	**R**
Final Report	**I**	**D**	**D**	**I**	**S**

---

3. Organization Structure



The organization structure represents the roles and responsibilities of different teams and staff members involved in managing and handling suspicious activities within the SAP system. The structure is divided into four main departments: **Security Department**, **IT Operations Department**, **Customer Service Department**, and **Other Departments**. Here's a breakdown of the responsibilities for each role:

- **CEO/CISO/CTO (Top Management)**:
    - View the entire Fiori dashboard, which provides real-time visibility of all system activities.

    - Receive monthly reports to monitor overall system performance and security.

    - Handle **HIGH** security cases, indicating critical security incidents that require executive intervention.

- **Security Manager**:
    - **Descriptions**: Security Department (Main team responsible for monitoring and managing security-related cases)

        - Oversee the Kanban Tracking system, which provides a high-level overview of ongoing cases (Note: Only tracking, no direct operations).

        - Monitor and track security activities in real-time.

        - Receive **HIGH** priority email alerts and respond accordingly.

        - Handle **HIGH** cases, which are security issues requiring immediate attention and resolution.

- **Security Analyst**::
    - **Descriptions**: Security Department (Main team responsible for monitoring and managing security-related cases)

        - Create cases based on security logs provided by the SAP system.

        - Analyze and identify potential threats or suspicious activities from the logs.

- **Security Staff**:
    - **Descriptions**: Security Department (Main team responsible for monitoring and managing security-related cases)

        - Monitor, verify, and handle the security cases assigned by the security manager.

        - Operate the Kanban system for tracking cases and ensuring they are being addressed in a timely manner.

- **IT Manager**:
    - **Descriptions:** IT Operations Department (Responsible for the operation and configuration of the SAP system):

        - Manage the SAP Security Audit Logs (SAL) to ensure continuous and accurate monitoring of the system.

        - Configure working hours and schedules for the IT staff involved in system management.

- **IT Staff**:
    - **Descriptions:** IT Operations Department (Responsible for the operation and configuration of the SAP system).:
        - Operate the SAP SAL system and handle any IT-related cases that arise.
        - Ensure the system is running smoothly and troubleshoot any technical issues related to security or performance.
- **Customer Support Staff**:
    - **Descriptions:** Customer Service Department (Handles communication and coordination with customers regarding security issues)
        - Coordinate with customers to address and resolve any security-related cases that affect them.
        - Provide support in managing cases and act as the liaison between the customer and the technical teams.
- **Employee**:
    - **Descriptions:** Other Departments (General employees with limited interaction with the security system).
        - Employees do not have access to the Security Audit Logs (SAL) and are not involved in the security system's operations.
        - They will receive individual alerts when anomalies or suspicious activities are detected in their areas.
        - They acknowledge alerts upon request, ensuring that they are aware of any security threats that might involve them.

4. Scope of Work

4.1 In-Scope Activities

This project focuses on developing an Alert and Monitoring System for suspicious activities within the SAP environment, based on Security Audit Logs (SAL).

The solution includes the following main components:

- **Real-time SAP Log Monitoring:**
  Monitoring of critical security events such as failed logins, unauthorized access, password changes, and user lockouts via SAP transaction SM20.

- **Automated Detection Algorithms:**
  Intelligent algorithms are implemented to detect suspicious behaviors, including:
    - **Brute-force attack attempts**
    - **Logins from new or unknown IP addresses**
    - **Excessive failed login attempts leading to user lockout**
    - **Password changes performed by users**

- **These algorithms are dynamic and adjustable through configurable system thresholds.**

- **Automated Notification System:**
  Upon detection of suspicious activities, the system sends automated email alerts and Microsoft Teams messages via a decoupled Node.js server connected externally to SAP.

- **Dashboard Interface (SAP Fiori):**
  A modern, user-friendly SAP Fiori application is provided for:
    - **Viewing real-time alerts and security events**
    - **Managing and tracking security cases**
    - **Visualizing event patterns and case statuses through interactive dashboards**

- **Case Management:**
  The system includes structured case management, allowing users to:
    - **Create cases directly from suspicious logs**
    - **Assign and track incident statuses (Open, In Progress, Review, Closed)**
    - **Manage evidence attachments**
    - **Maintain a history of investigation activities and comments for collaboration**

- **External Service Integration:**
  The solution integrates with external services:
    - **Email SMTP servers for notification delivery**
    - **Geolocation APIs for enhanced IP address analysis**
    - **Designed through an independent Node.js backend, separated from the SAP ABAP environment for scalability and flexibility.**

4.2 Out of Scope

4.2.1 Out of Project Scope

(*Activities not covered by the project team*)

- **Implementation Services:**
  The project does not include final deployment to production systems, Go-Live support, or performance optimization.
- **Operations and Maintenance:**
  The project team is not responsible for ongoing system operations, 24/7 monitoring, incident response, or maintaining Node.js servers and SAP systems after delivery.

4.2.2 Out of Product Scope

(*Functionalities and features not delivered by the solution*)

- **Automatic Remediation:**
  The product detects and alerts about suspicious activities but does not automatically take action (e.g., blocking users, disabling accounts).
- **Machine Learning-Based Detection:**
  The system does not utilize AI/ML models for predictive security threat detection in this version.
- **Mobile-Optimized Interface:**
  The SAP Fiori applications are optimized for desktop and tablet use only; mobile device support is not included.
- **Multi-language Support:**
  The solution is delivered only in English. Localization or multi-language interfaces are not provided.
- **Real-time Event Streaming:**
  The system does not integrate real-time event streaming platforms (e.g., Kafka, AWS

Kinesis) for live security event processing.

- **External SIEM Integration:**
  The product does not integrate with external SIEM (Security Information and Event Management) solutions like Splunk, QRadar, or Elastic Stack.

- **Two-way User Communication Workflow:**
  The notification system is one-way only (email/Teams alert). It does not manage user acknowledgments, reminders, or escalation workflows based on user responses.

5. Final Report Structure

The structure of this report is as follows:

- **Introduction**: Overview of the project, objectives, and team members.
- **Product Statements**: Analysis of the product's necessity, features, and limitations of the current system.
- **SAP Configurations**: Overview of SAP configurations and their relevance to the project.
- **SAP Coding**: Description of the coding elements and technical specifications.
- **Notifications System**: The Node.js server uses **Nodemailer** to send email notifications.
- **Implementation & Testing**: Process of implementing the system, followed by a detailed testing phase.

II. Product Statements

1. Product Analysis

Large enterprises operating on SAP platforms increasingly face complex security threats, including unauthorized access attempts, privilege misuse, and fraudulent transactions. While SAP provides native logging tools such as **SM19** (Security Audit Configuration) and **SM20** (Audit Log Review), these tools present significant limitations in modern enterprise environments:

- **Lack of real-time detection**: SM19/SM20 require manual inspection and do not offer proactive alerts.
- **Limited automation**: Analysts must manually search for anomalies across massive volumes of raw logs.

- **Poor prioritization**: The absence of severity classification makes it difficult to distinguish between routine events and potential security incidents.

- **Minimal integration**: Built-in tools are tightly bound to SAP GUIs, making cross-system collaboration and external service integration difficult.

To address these pain points, the **"Developing Alert and Monitoring Functions for Suspicious Activities on SAP System"** application was developed. This custom-built solution augments SAP's security capabilities by automating the **detection, classification, alerting, and management** of suspicious activities.

1.2 How the App Solves Existing Challenges

Challenge	Solution Provided by the App
Manual log inspection	Real-time log viewer with advanced filtering, sorting, and categorization
No alerting or escalation	Automated case creation and email notification on suspicious activities
Inability to prioritize risks	Severity levels (Low, Medium, High) assigned to log events and cases
Poor visibility into system activity	Interactive dashboard showing trends, volume, and types of events
Weak integration with external tools	Node.js API enables seamless external communication and scalable email delivery

---

1.3 Key Features

1.3.1. Dynamic Audit Log Viewer

- Centralized interface for analyzing security audit logs
- Filters by user, severity, event type, time range
- Sortable columns and detailed log view with context
- Option to export filtered logs for offline analysis

1.3.2. Integrated Case Management System

- Converts flagged logs into actionable investigation cases
- Tracks status through Open, In Progress, Review, and Closed stages
- Includes role-based assignment, evidence attachment, and activity logging
- Facilitates internal collaboration and accountability

1.3.3. Real-Time Dashboard Visualizations

- Displays system-wide event trends by time and type
- Breaks down medium- and high-severity counts per hour or per day
- Log type distribution analysis
- Summary panels showing open, in-progress, and closed cases
- Supports filterable time ranges and manual refresh

1.3.4. External Service Integration (via Node.js API)

- Sends email alerts to relevant personnel upon:
    - Case creation
    - Case status updates
    - Assignee changes
- Operates independently of SAP ABAP, improving speed and scalability

- Ensures better responsiveness to emerging security incidents

---

1.4 Conclusion

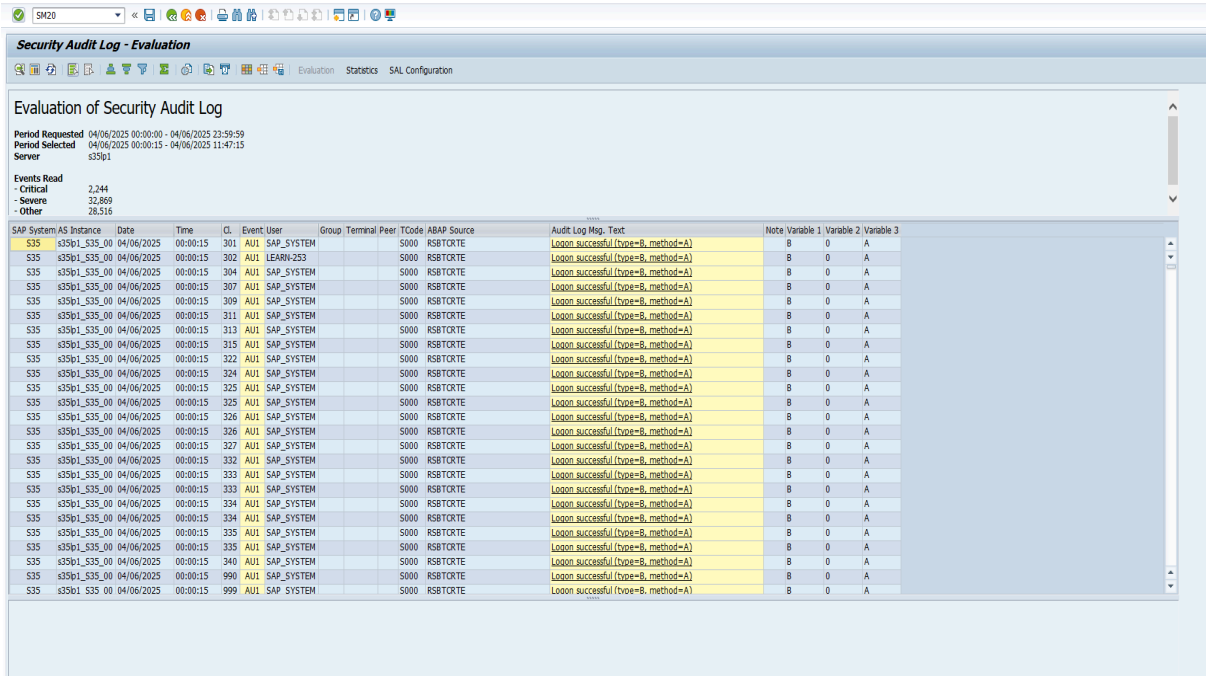
This application significantly enhances the SAP environment's native logging and monitoring capabilities by providing a modern, automated, and integrated solution. It enables faster threat detection, improved incident response, and greater operational transparency—all critical factors in securing enterprise SAP systems.

2. As-Is

In the current SAP system, there are standard features available for auditing and notification purposes, such as:

2.1 Manage SM20 (Security Audit Log Viewer):

Allows users to view security-related log entries captured by the system, including logins, failed attempts, and sensitive actions.



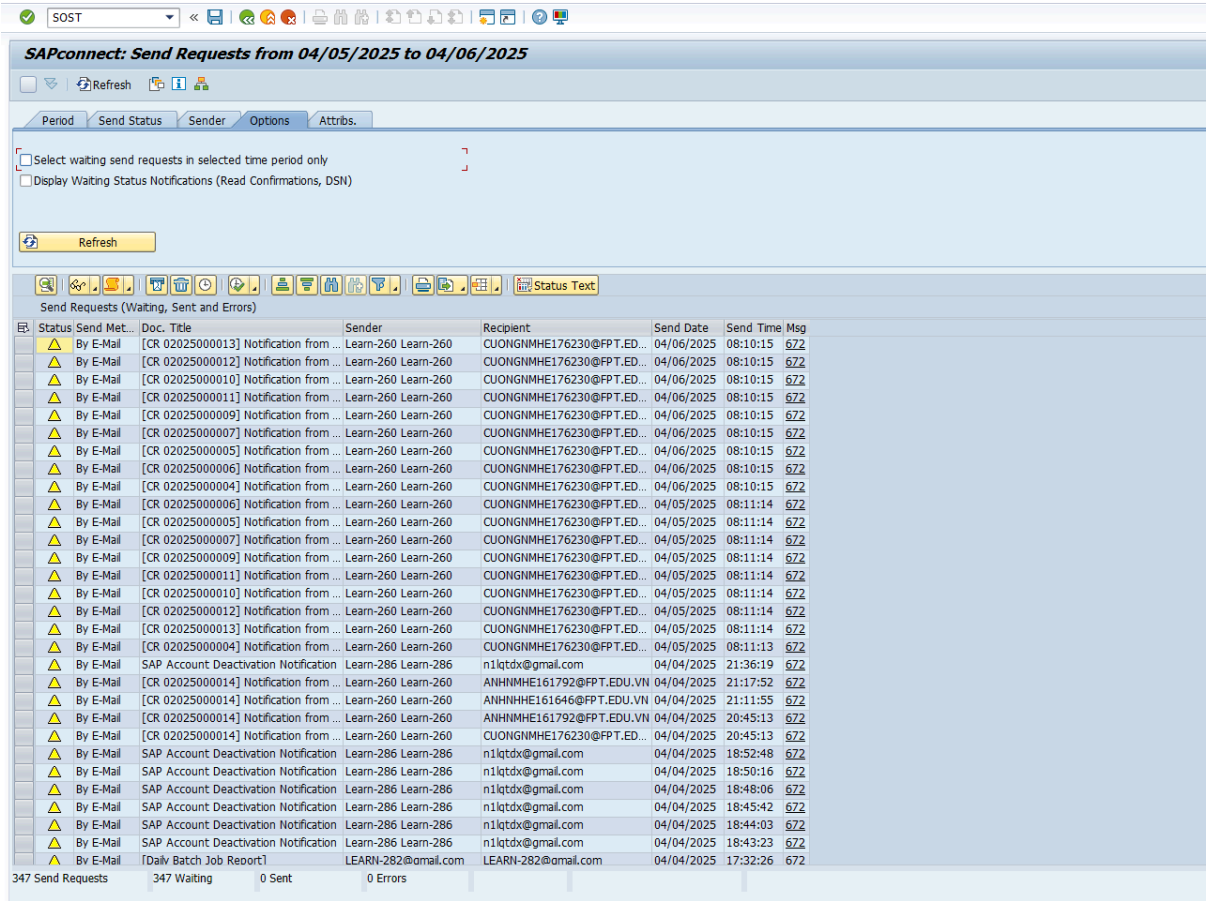
2.2 Limitations of Current Features:

Although SM20 provides access to security logs, it is primarily designed for generic auditing and lacks flexibility for business-specific filtering and visualization. It often displays an overwhelming amount of raw data, including irrelevant logs, making it difficult for users to focus on critical events that align with internal policies or compliance rules.

➔ **Why Improve?:** To make audit logs more useful and relevant, it's necessary to customize filtering, categorization, and display logic to suit the specific auditing criteria of the business. This includes role-based views, dynamic filters, and user-friendly visualizations.

2.3 Send emails by SOST (SAPconnect Send Requests):

Manages outbound email communication within the SAP system, displaying the status of sent emails.



Status	Send Met...	Doc. Title	Sender	Recipient	Send Date	Send Time	Msg
	By E-Mail	[CR 02025000013] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/06/2025	08:10:15	672
	By E-Mail	[CR 02025000012] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/06/2025	08:10:15	672
	By E-Mail	[CR 02025000010] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/06/2025	08:10:15	672
	By E-Mail	[CR 02025000011] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/06/2025	08:10:15	672
	By E-Mail	[CR 02025000009] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/06/2025	08:10:15	672
	By E-Mail	[CR 02025000007] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/06/2025	08:10:15	672
	By E-Mail	[CR 02025000005] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/06/2025	08:10:15	672
	By E-Mail	[CR 02025000006] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/06/2025	08:10:15	672
	By E-Mail	[CR 02025000004] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/06/2025	08:10:15	672
	By E-Mail	[CR 02025000006] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/05/2025	08:11:14	672
	By E-Mail	[CR 02025000005] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/05/2025	08:11:14	672
	By E-Mail	[CR 02025000007] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/05/2025	08:11:14	672
	By E-Mail	[CR 02025000009] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/05/2025	08:11:14	672
	By E-Mail	[CR 02025000011] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/05/2025	08:11:14	672
	By E-Mail	[CR 02025000010] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/05/2025	08:11:14	672
	By E-Mail	[CR 02025000012] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/05/2025	08:11:14	672
	By E-Mail	[CR 02025000013] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/05/2025	08:11:14	672
	By E-Mail	[CR 02025000004] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/05/2025	08:11:13	672
	By E-Mail	SAP Account Deactivation Notification	Learn-286 Learn-286	n1lqtdx@gmail.com	04/04/2025	21:36:19	672
	By E-Mail	[CR 02025000014] Notification from ...	Learn-260 Learn-260	ANHNMHE161792@FPT.EDU.VN	04/04/2025	21:17:52	672
	By E-Mail	[CR 02025000014] Notification from ...	Learn-260 Learn-260	ANHNHHE161646@FPT.EDU.VN	04/04/2025	21:11:55	672
	By E-Mail	[CR 02025000014] Notification from ...	Learn-260 Learn-260	ANHNMHE161792@FPT.EDU.VN	04/04/2025	20:45:13	672
	By E-Mail	[CR 02025000014] Notification from ...	Learn-260 Learn-260	CUONGNMHE176230@FPT.ED...	04/04/2025	20:45:13	672
	By E-Mail	SAP Account Deactivation Notification	Learn-286 Learn-286	n1lqtdx@gmail.com	04/04/2025	18:52:48	672
	By E-Mail	SAP Account Deactivation Notification	Learn-286 Learn-286	n1lqtdx@gmail.com	04/04/2025	18:50:16	672
	By E-Mail	SAP Account Deactivation Notification	Learn-286 Learn-286	n1lqtdx@gmail.com	04/04/2025	18:48:06	672
	By E-Mail	SAP Account Deactivation Notification	Learn-286 Learn-286	n1lqtdx@gmail.com	04/04/2025	18:45:42	672
	By E-Mail	SAP Account Deactivation Notification	Learn-286 Learn-286	n1lqtdx@gmail.com	04/04/2025	18:44:03	672
	By E-Mail	SAP Account Deactivation Notification	Learn-286 Learn-286	n1lqtdx@gmail.com	04/04/2025	18:43:23	672
	By E-Mail	[Daily Batch Job Report]	LEARN-282@gmail.com	LEARN-282@gmail.com	04/04/2025	17:32:26	672

2.4 Limitations of Current Features:

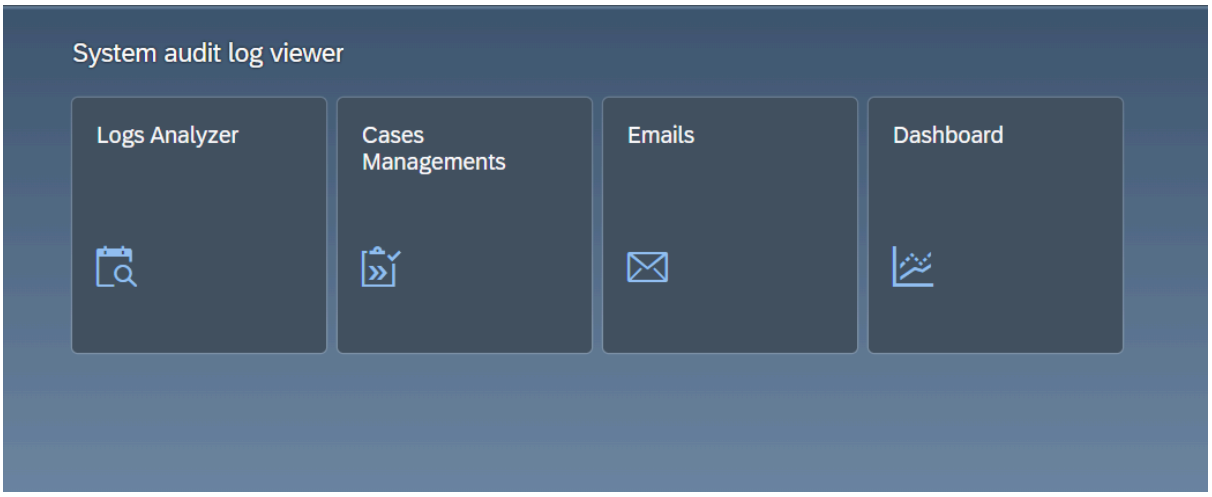
SOST handles the sending of email notifications from within SAP, but it is a one-way communication system. It does not support monitoring or processing responses from users. Additionally, it lacks built-in support for reminders or follow-up messages, which are essential for businesses that require user acknowledgment or compliance actions.

➔ **Why Improve? :** To meet modern communication needs, businesses require two-way communication with users, including:

- Tracking responses from recipients.
- Sending reminders or escalations if no response is received within a defined time.
- Customizable templates and content based on event type or severity.

3. To be (After Project Implementation)

3.1. Production



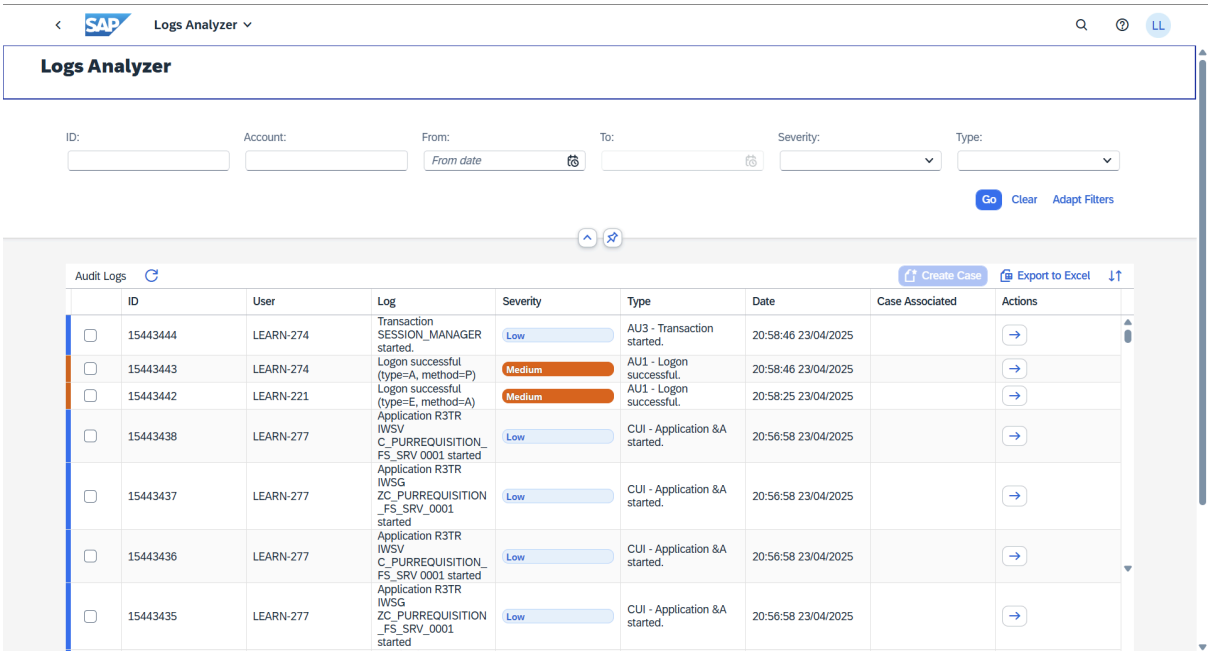
Catalog ID	User Account	Role Code	Accessible Modules
LEARN-253	zg1.salv.gd	GD	Dashboard, Emails
LEARN-254	zg1.salv.tpan	TPAN	Dashboard, Case Management, Emails
LEARN-255	zg1.salv.nvan	NVAN	Dashboard, Case Management
LEARN-256	zg1.salv.nvan	NVAN	Dashboard, Case Management
LEARN-257	zg1.salv.nvpt	NVPT	Dashboard, Logs Analyzer

Access Summary by Module

- **Dashboard**: Accessible by all users
- **Logs Analyzer**: Accessible by NVPT
- **Case Management**: Accessible by TPAN, and NVAN
- **Emails**: Accessible by GD and TPAN

3.1.1. Logs Analyzer – SAP System Log Analysis

Overview:
Logs Analyzer is a dedicated tool that enables system administrators to monitor, analyze, and investigate SAP system audit logs in real time. This application helps track user behavior and detect suspicious or unauthorized activities, enhancing the overall security posture of the system.



Key Features:

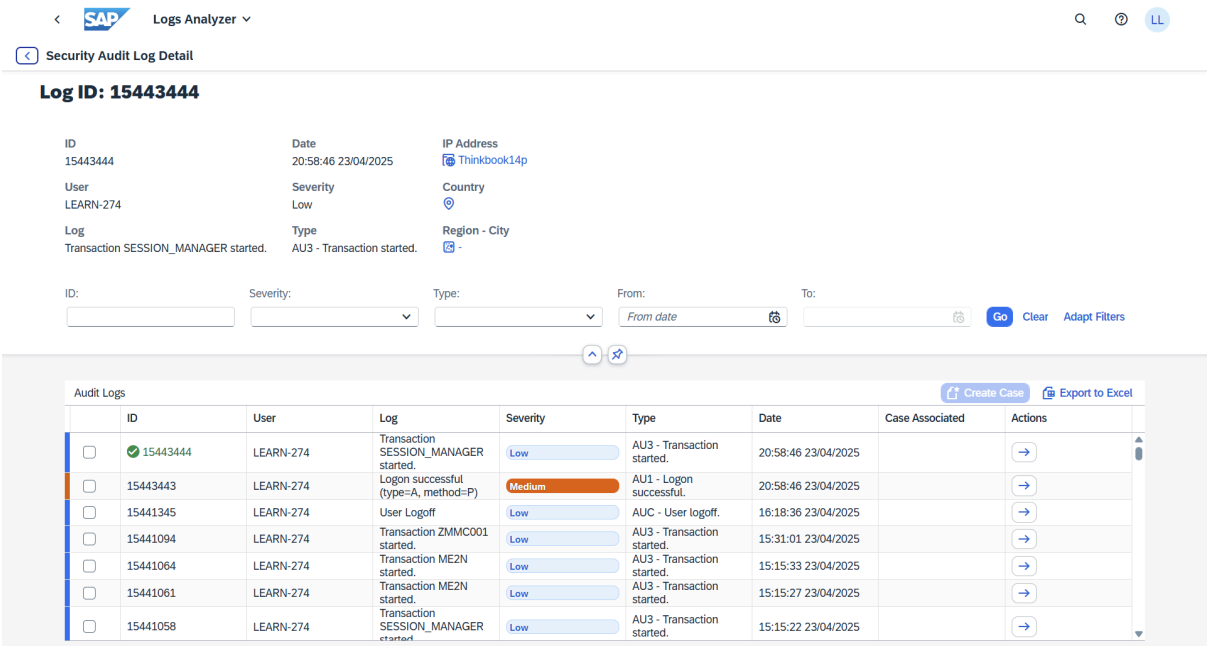
1. Main Interface

- Accessible from the **System Audit Log Viewer** dashboard.
- Displays a comprehensive audit log table with columns such as ID, User, Log, Severity, Type, and Date.

- Each entry represents a specific action or event recorded in the system.

2. Advanced Filtering Options

- Filters include: ID, Account, Severity, Event Type, and Date Range.

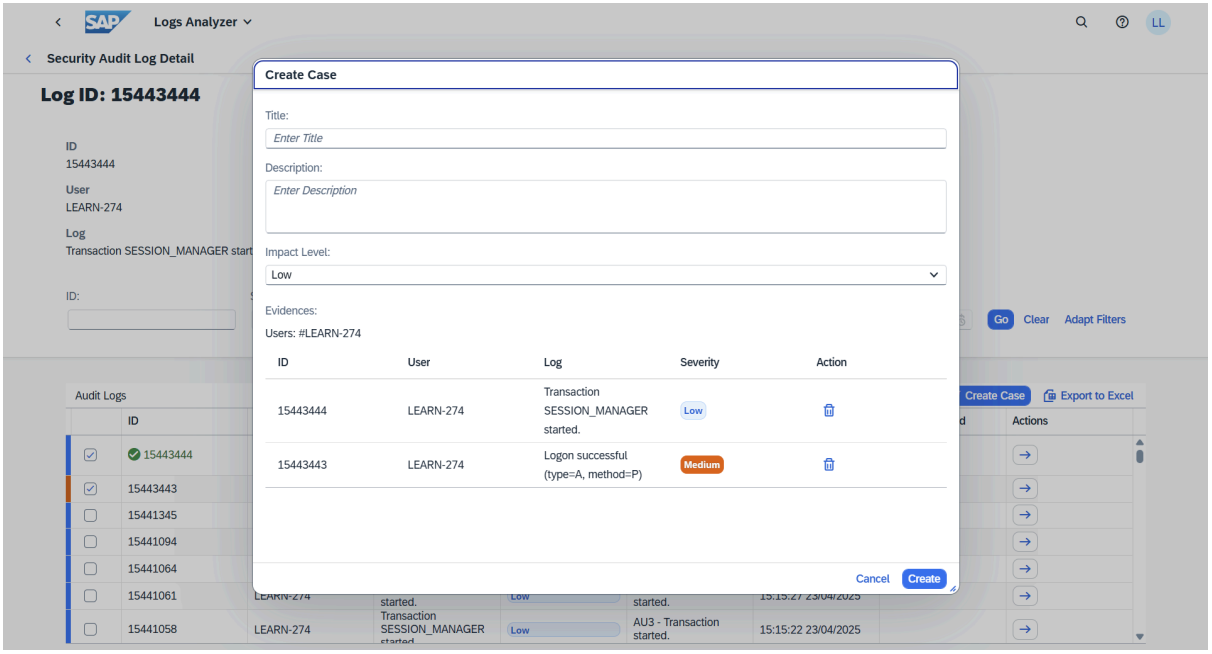


- Enables administrators to narrow down logs to identify:
    - Failed login attempts
    - Suspicious logins (e.g., method = P, type = A)
    - Transaction starts and other system interactions

3. Log Detail View

- Clicking the arrow icon in the Actions column opens a detailed view of the selected log:
    - Shows all relevant metadata including User, Log Description, Date, Severity, Type, IP Address, and Device Name.
    - Displays related logs in chronological order to understand user behavior across a session.

4. Case Creation



- Admins can select suspicious logs and click **"Create Case"** to initiate a security case directly from the logs.
- The case form includes:
    - Title and Description
    - Impact Level selection (e.g., Low, Medium, High)
    - Attached Evidences (selected log entries)
- This helps streamline incident handling and ensures proper tracking and documentation.

5. Export Functionality

- Logs (filtered or all) can be exported to Excel for offline analysis or reporting purposes.

Benefits:

- Enables **proactive security monitoring** and **early threat detection**.
- Reduces investigation time by providing clear visibility into user activities.
- Supports audit and compliance by maintaining an accessible trail of security-relevant events.

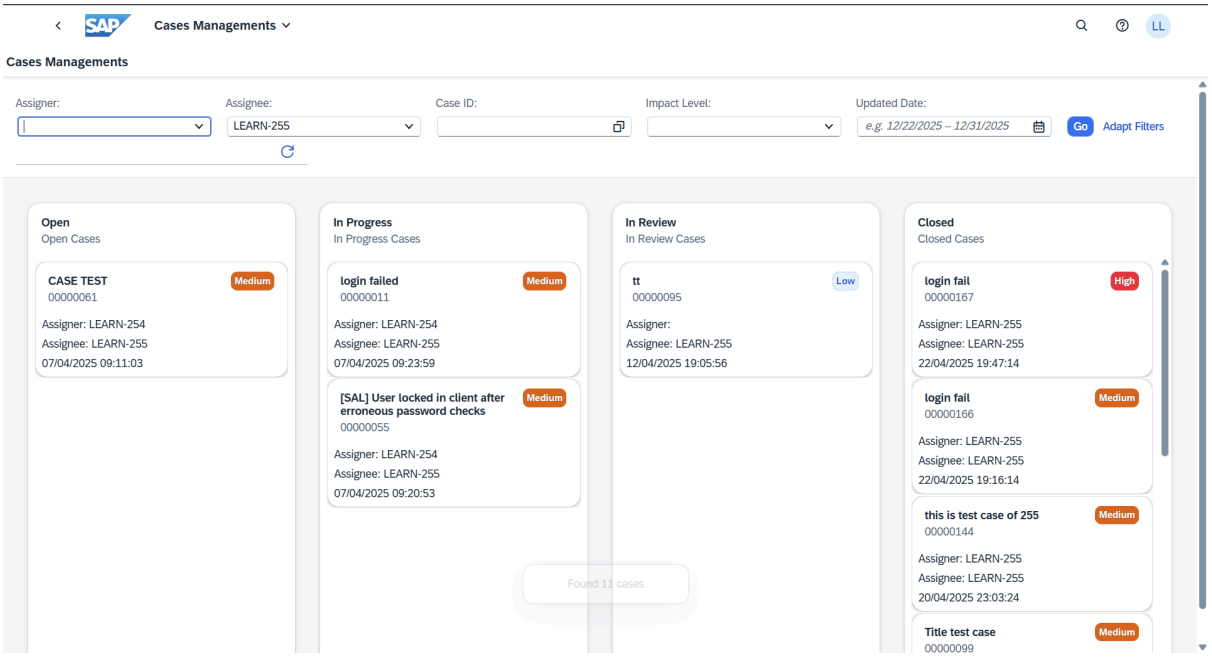
3.3.2 Cases Management – Centralized Security Case Handling

Overview:

The Cases Management module is designed to support the end-to-end lifecycle of security incidents detected through audit logs in the SAP system. It enables administrators to create, track, and manage investigation cases derived from suspicious log entries. This structured approach ensures that every event of concern is properly documented, investigated, and resolved.

Key Features and Functionality:

1. Case Overview Dashboard

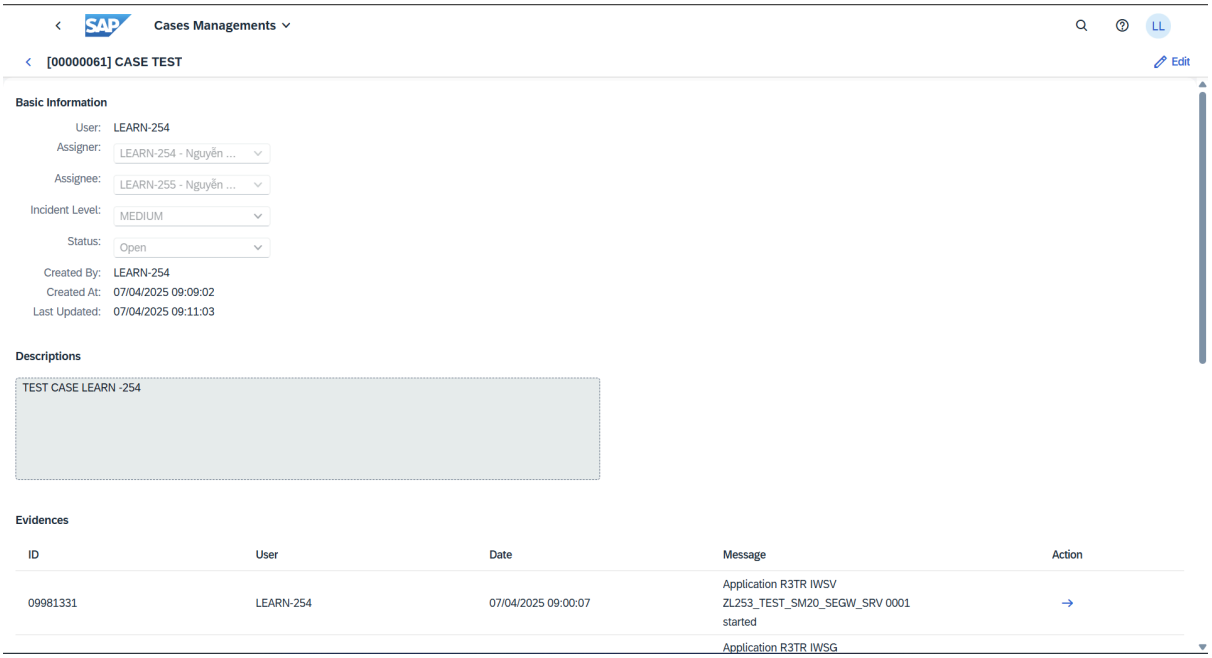


Cases are categorized into four main statuses:

- Open
- In Progress
- In Review
- Closed

Each case card includes the case ID, title, severity level, assigner, assignee, and last updated timestamp. This Kanban-style view allows for an intuitive understanding of case progression and team workload.

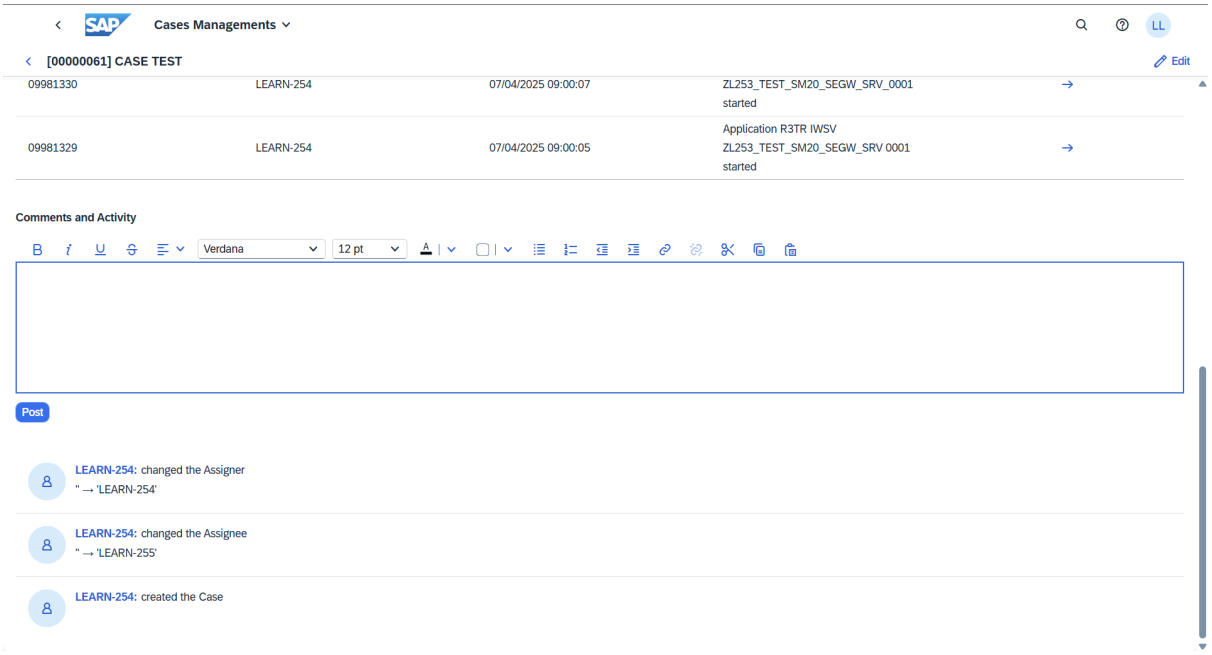
2. Detailed Case View



When accessing a case, users can view:

- Basic information including assigner, assignee, severity (impact level), status, creation and update timestamps
- A descriptive section explaining the context or findings of the case
- A list of linked audit logs (evidences), providing full details such as log ID, user, date, and event message. Each evidence item is traceable back to the source log for further verification.

3. Comments and Activity History



The module supports collaborative case handling with a comment section for internal discussions and investigation notes. All changes made to a case—such as updates to assignees or status—are automatically recorded in a chronological activity log to maintain traceability.

4. Filtering and Search Capabilities

Advanced filters allow users to search by assigner, assignee, case ID, impact level, and updated date range. This is particularly useful for managing large volumes of cases and focusing on high-priority items.

Benefits:

- Ensures structured and accountable management of security incidents
- Enhances team collaboration during investigations
- Improves visibility into ongoing and resolved cases
- Provides a clear audit trail for compliance and review purposes

3.3.3 Emails – Security Notifications via External API

Overview:
The **Emails module** is responsible for sending automated notifications to administrators and security teams regarding key security events occurring within the SAP system. These notifications ensure timely awareness and enable fast response to potential threats.

Key Features and Functionality

1. Real-Time Alerts

Email notifications are triggered by system events such as:

- Case status updates (e.g., Open → Closed)
- New case creation
- Case assignments or reassignments

Each email includes essential metadata:

- Case ID and name
- Old status and new status
- Date and time of the change
- Sender and recipient information

2. External Email Delivery Architecture

Instead of relying on SAP's internal ABAP mailing service, the system uses a **Node.js-based API** to send emails through an external server. This design offers several advantages:

- Fully decouples SAP from the mailing process
- Avoids SAP system resource limitations
- Enables use of modern mailing services (e.g., SMTP providers, email APIs)

3. Web-Based Email Viewer

The Emails module also includes an inbox-like UI within the SAP system:

- Displays a list of all notifications with timestamps and recipient emails
- Provides a search function to filter by subject or recipient

- Shows structured email content with key case details and action buttons

4. Structured Email Format
Each email is formatted with a consistent and clean layout, containing:

- Notification type (e.g., status change, new case)
- Case information (ID, name, old/new status)
- Direct access link to the relevant case detail page in the SAP system

---

Benefits

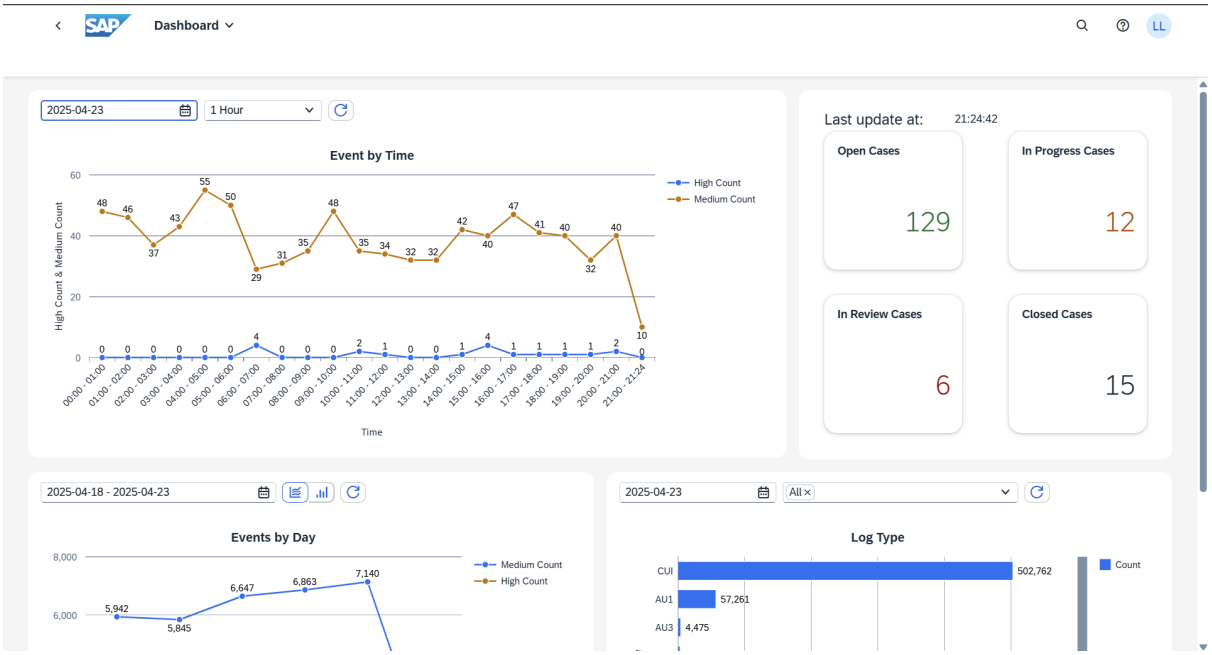
- Operates independently of SAP's built-in mail infrastructure
- Enables **faster and more reliable email delivery**
- Easily **scalable and maintainable** using modern backend technology (Node.js)
- Supports **custom templates and localization**
- Promotes real-time collaboration among incident response teams

3.3.4 Dashboard – Centralized Security Intelligence and Monitoring

Overview:
The Dashboard module provides a real-time, centralized view of all security-related activities within the SAP system. By aggregating data from logs and case management, it delivers critical insights through dynamic charts and key performance indicators (KPIs), allowing administrators to detect trends, assess threats, and respond efficiently.

Key Features and Functionality

1. Visual Timeline of Security Events



- The “Event by Time” chart displays counts of medium and high-severity events across hourly intervals.

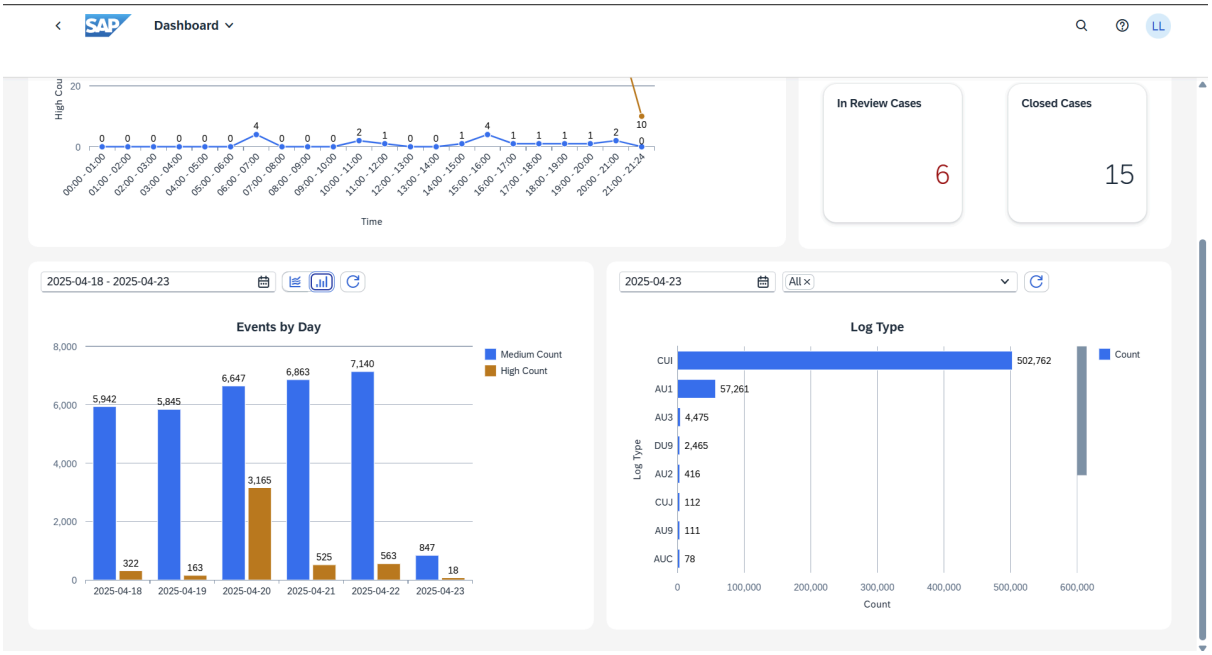
- Enables teams to identify spikes, patterns, or unusual activity times that may indicate targeted or coordinated threats.

2. Daily Event Overview

- The “Events by Day” chart tracks the volume of logged events per day, categorized by severity level.

- Supports trend analysis over configurable date ranges, helping teams plan for peak loads or recurring incidents.

3. Log Type Breakdown



- A horizontal bar chart categorizes logs by type (e.g., CUI, AU1, AU3) and volume.
- Helps prioritize which systems or functionalities are generating the most critical or frequent activities.

4. Case Summary Panels

- Real-time counts of:
    - Open Cases
    - In Progress Cases
    - In Review Cases
    - Closed Cases
- These metrics help management assess workflow status, backlog, and investigation effectiveness at a glance.

5. Custom Date Filtering and Refreshing

- Time filters enable selection of specific days, ranges, or real-time updates.
- Users can refresh the dashboard manually to ensure the most current data is displayed.

Benefits

- Provides a **single source of truth** for all audit-related metrics and security statuses.
- Enhances **situational awareness** through live data visualization and filtering.
- Enables **faster response** by clearly surfacing high-impact events and overdue investigations.
- Reduces reliance on manual tracking or external reporting tools by integrating everything into one interface.

1.

Limitations:

- The solution will not modify the underlying SAP security log structure, meaning the original log formats and content are not altered. The app will only work with the existing SAP log data.
- Changes to SAP's user roles and authorization systems are not part of this project scope. The application will not handle user management or modify access control features in SAP.
- This solution is focused exclusively on security monitoring and alerts. Non-security-related SAP functionalities are beyond the scope of this development.

4. Project Management

4.1. Training Plan – Listing and Purpose

4.1.1 Blueprint (converted to To-Be Process):

- **Purpose**: This document outlines the expected processes (To-Be) that the system will follow after development. It helps the team understand how the system will operate post-deployment and serves as the foundation for configuring and building the system's features in line with user requirements.
- **Usage**: It is used as a guideline for the development of features and for providing direction during the actual deployment phase.

4.1.2 Configuration Notes:

- **Purpose**: This document contains detailed notes about system configurations, including setup instructions for SAP components like SM19, SM20, ZG_LOG, and

SAP Fiori. It serves as a guide to ensure the components are configured correctly for the project.

- **Usage**: Used by developers and technical staff for configuring the system properly and avoiding errors during implementation.

4.1.3 Functional Specification:

- **Purpose**: This document outlines the system's functional requirements, detailing each feature and how it will operate. It acts as a key document for developers to understand user needs and build the corresponding features accordingly.

- **Usage**: Used during programming and development to ensure that the features are implemented according to the approved requirements.

4.1.4 Technical Specification:

- **Purpose**: This document outlines the technical aspects of the system, including technologies used, database structure, and how system components will interact. It ensures developers understand the architecture and design for implementing the system.

- **Usage**: Used by developers to build the system and ensure compatibility between software and hardware components.

4.1.5 Unit Test Evidence:

- **Purpose**: This document includes the results from unit testing, confirming that individual components of the system work correctly. It validates that each module of the system functions as expected before integration with the larger system.

- **Usage**: Used to prove that the individual functions of the system have been tested and verified to work before system integration.

4.1.6 Test Scenario:

- **Purpose**: This document describes the test scenarios that the system will undergo to identify potential bugs or issues. It ensures that the system is tested in realistic, practical scenarios to confirm its proper behavior.

- **Usage**: Used to design test cases and evaluate how the system behaves under different situations during testing.

4.1.7 User Manual:

- **Purpose**: The user manual provides instructions for end-users on how to use the system. It includes steps for interacting with the system, such as how to filter logs, manage cases, and receive notifications.
- **Usage**: Used to support users in becoming familiar with the system and optimizing their interaction with the platform.

4.1.8 UAT (Acceptance Test) Sign-off:

- **Purpose**: The AUT sign-off document is used to confirm that the system has met all agreed-upon requirements and functions. It acts as official confirmation from the customer that the system is ready for deployment and use.
- **Usage**: This document is signed by the customer after the system has passed the acceptance testing phase, signifying the formal acceptance of the system.

4.1.9 Thesis Defense:

- **Purpose**: This document assists the student in preparing for their thesis defense. It summarizes the development process, methodology, and outcomes. Students must present the results and justify the technical and functional decisions made throughout the project.
- **Usage**: Used when the student is defending their project in front of a panel to demonstrate the success and validity of their work.

4.2. Project Management Plan

No.	Task Description	Duration	Planned Date (From)	Planned Date (To)
1	**Preparation + Explore**			
1.1	Project Kick-Off Meeting	1 day	17-Jan	17-Jan

1.2	Study & Clarify Requirement	3 days	20-Jan	22-Jan
1.3	Preparing Presentation Slides for Workshop	2 days	22-Jan	24-Jan
1.4	**Business Requirement Workshop**	5 days	27-Jan	31-Jan
1.4.1	Functional	3 days	27-Jan	29-Jan
1.4.2	Technical	2 days	30-Jan	31-Jan
1.5	**Exploration (Requirement Analysis)**	13 days	3-Feb	14-Feb
	AS-IS analysis	7 days	3-Feb	10-Feb
	Investigate requirement & SAP system	4 days	3-Feb	6-Feb
	Create the draft function list	3 days	7-Feb	9-Feb
	Conduct Workshop	1 day	10-Feb	10-Feb
	Propose solution (TOBE Process)	6 days	11-Feb	14-Feb
	Design solution for Audit System	3 days	11-Feb	13-Feb
	Create and Update Tobe Process	3 days	12-Feb	14-Feb
1.6	**Blueprint Finalization**	9 days	17-Feb	21-Feb
1.6.1	Create Blueprint	4 days	17-Feb	20-Feb
1.6.2	Review Blueprint (3 times maximum)	3 days	21-Feb	23-Feb
1.6.3	Sign-off Blueprint	2 days	24-Feb	25-Feb
2	**Realization**			
2.1	**Functional Specification Document (FSD)**	16 days	1-Mar	15-Apr
2.3.1	Functional Specification	4 days	1-Mar	4-Mar
2.3.2	Technical Specification	4 days	5-Mar	8-Mar
2.3.3	Review Functional Specification	3 days	9-Mar	11-Mar
2.3.4	Review Technical Specification	3 days	12-Mar	14-Mar
2.4	**Development**	29 days	15-Apr	21-Apr

	Coding & Unit Test	15 days	15-Apr	30-Apr
	Dashboard (Frontend)	5 days	15-Apr	19-Apr
	Dashboard (Backend)	5 days	20-Apr	24-Apr
	Auto-detect Algorithms	3 days	21-Apr	23-Apr
	Log Table (Frontend)	3 days	23-Apr	25-Apr
	Log Table (Backend)	3 days	26-Apr	28-Apr
	Cases (Backend + Frontend)	5 days	29-Apr	3-May
	Fix bug	Ongoing	3-May	5-May
2.5	**Testing**			
2.5.2	Create Test Scenario	2 days	1-May	2-May
2.5.3	System Test	5 days	3-May	7-May
2.5.4	Bug Fix & Re-test	4 days	8-May	11-May
2.6	**User Manual**	7 days	12-May	19-May
3	**UAT (User Acceptance Testing)**			
3.1	UAT	5 days	20-May	24-May
3.2	UAT Sign Off	2 days	25-May	26-May
4	**Defense (Go-live)**			
4.1	Thesis Defense	2 days	7-May	8-May

Key Phases and Dates:

1. **Preparation + Explore**: Focus on understanding requirements, preparing presentations, and conducting workshops with the client. Expected to be completed by **31-Jan**.

2. **Realization**: This phase covers functional and technical documentation, development, and testing, with key tasks spanning from **1-Mar** to **21-Apr**.

3. **UAT**: The system will undergo User Acceptance Testing (UAT) in **May**, followed by sign-off.

4. **Defense (Go-live)**: The final defense of the project, which includes project delivery and client sign-off, will take place on **7th May – 8th May**.

III. SAP Configurations

1. Enterprise Structure Definition

In this project, the customization of SAP enterprise structures is **not applicable**.
The solution is designed to operate **independently** from the organizational hierarchy. It does not rely on structures such as Plant, Company Code, or Sales Area.
This architectural decision allows the report and monitoring tools to function seamlessly across the entire system landscape, regardless of how the enterprise structure is defined. As a result, deployment is simplified, and there is no requirement for organizational dependency alignment.

2 Business Process Configuration

Although the solution is **focused on system operation monitoring** rather than supporting business transactions, several critical technical configurations have been introduced to enhance flexibility and performance.

2.1 Flag Management for Checking Algorithms

The system incorporates dynamic flags to control the activation of specific checking algorithms. Administrators can enable or disable particular monitoring features without modifying the program source code, allowing quick and agile adjustments based on operational needs.

2.2 Threshold Configuration for Brute-force Attack Diagnostics

A configurable **threshold value** has been implemented for DDOS attack detection algorithms.
This configuration helps fine-tune the system's sensitivity to abnormal traffic patterns, minimizing false positives while maintaining strong security posture.

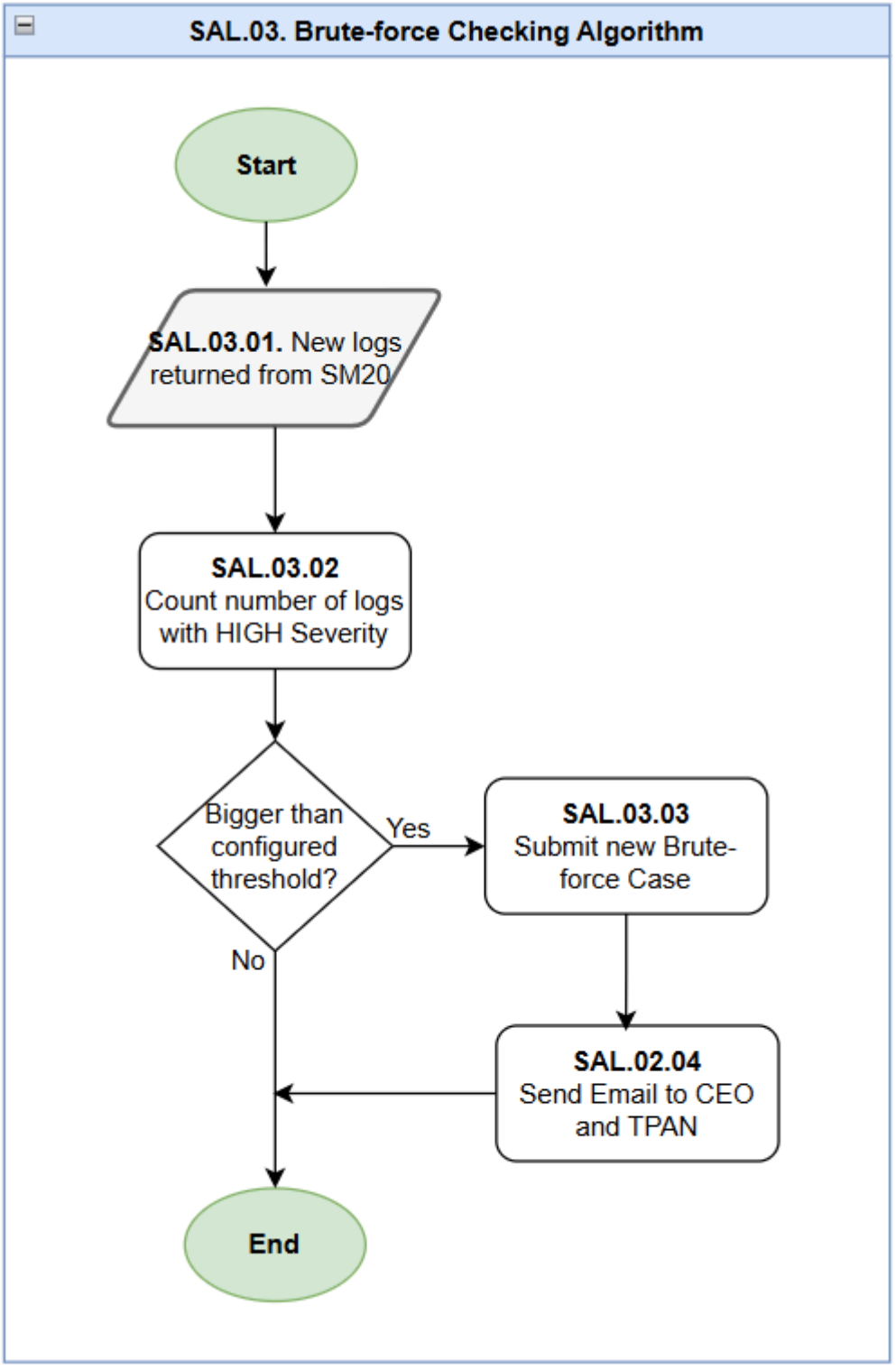
2.3 External Base API URL Management

The system provides a flexible setting for the **External Base API URL**, allowing integration with external monitoring platforms or external alerting services.
This setting ensures seamless external communication without hardcoding URLs into ABAP programs, promoting maintainability and scalability.

3. Suspicious Events Diagnostic Algorithms

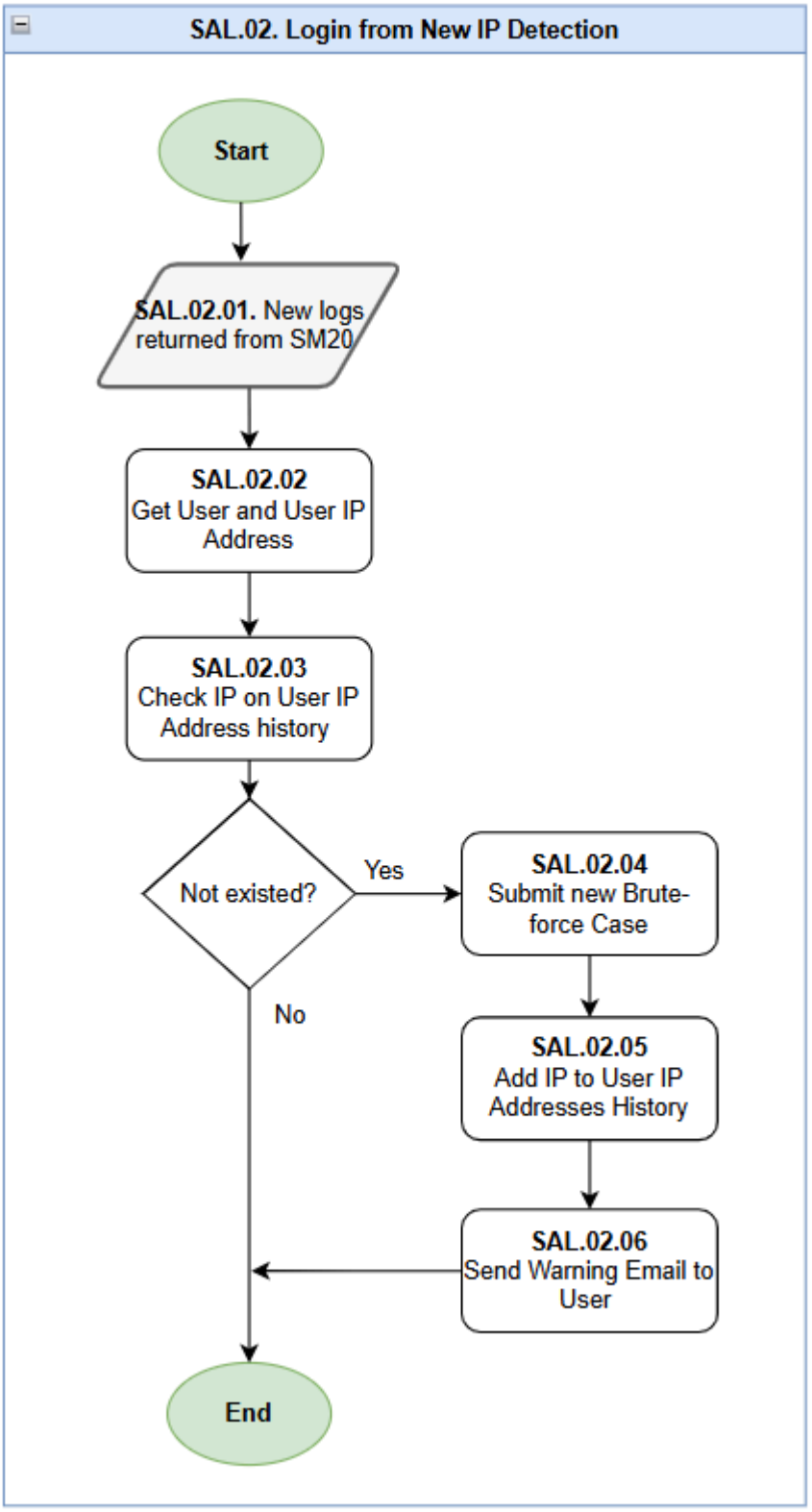
3.1 Brute-force Attacking

The algorithm will check the number of logs with HIGH severity count in a time (15 minutes). If the number exceed the number configured in the System, an Alert will be raised



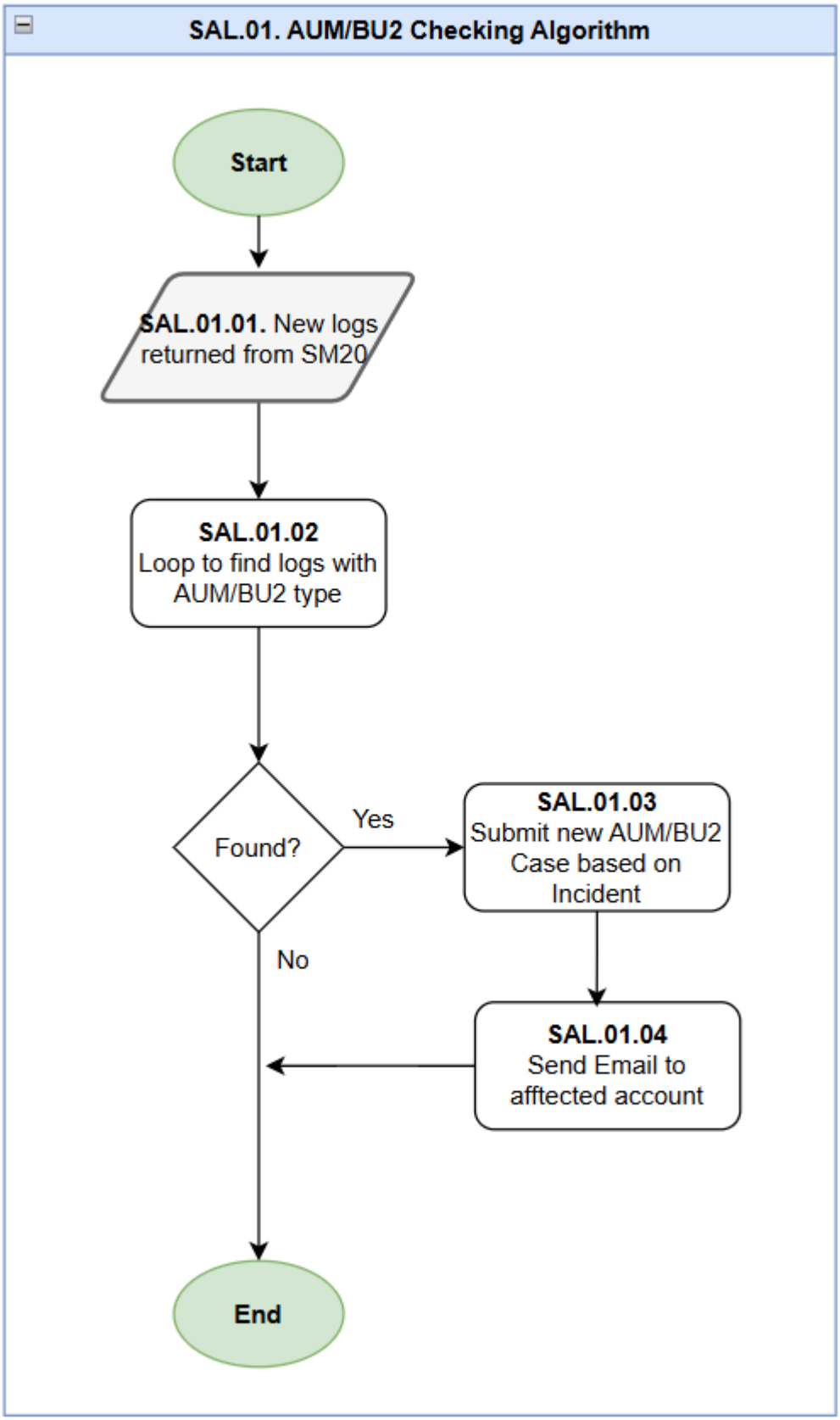
3.2 User logon from new IP Address

The algorithm will try to get the IP where the user logged in. If this IP is new and not in the history of previously logged in user IPs, an alert will be raised


SAL.02. Login from New IP Detection
Start
SAL.02.01. New logs returned from SM20
SAL.02.02
Get User and User IP Address
SAL.02.03
Check IP on User IP Address history
Not existed?
Yes
No
SAL.02.04
Submit new Brute-force Case
SAL.02.05
Add IP to User IP Addresses History
SAL.02.06
Send Warning Email to User
End

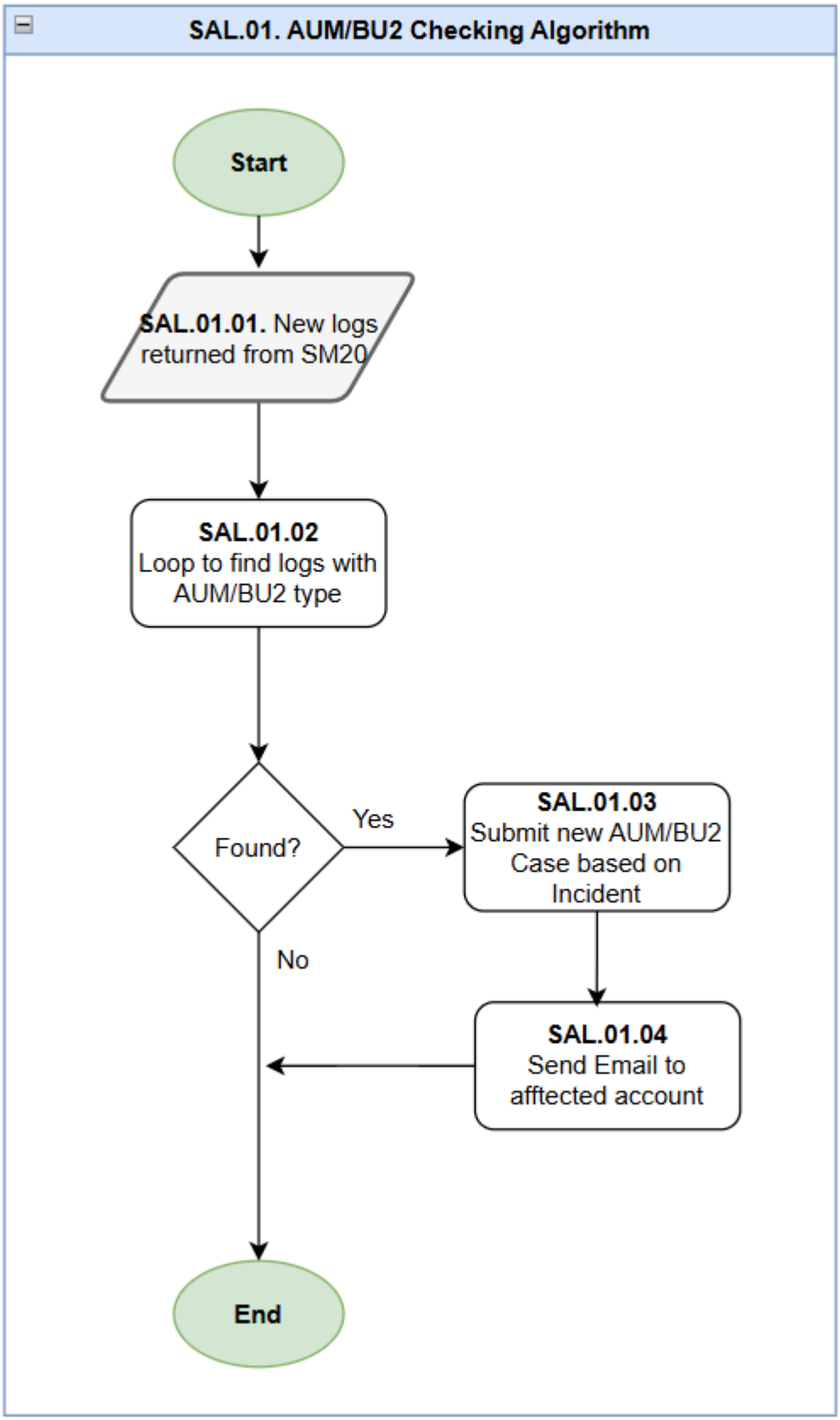
3.3 User locked on client after numerous tries

The algorithm will check if the user account has been locked temporarily on the client due to numerous tries. If the incident happen, an alert will be raised


SAL.01. AUM/BU2 Checking Algorithm
Start
SAL.01.01. New logs returned from SM20
SAL.01.02
Loop to find logs with AUM/BU2 type
Found?
Yes
No
SAL.01.03
Submit new AUM/BU2 Case based on Incident
SAL.01.04
Send Email to afftected account
End

3.4 User password changed on client

The algorithm will check if the user account password has been changed on the client. If the action happen, an warning email will be sent to user


SAL.01. AUM/BU2 Checking Algorithm
Start
SAL.01.01. New logs returned from SM20
SAL.01.02
Loop to find logs with AUM/BU2 type
Found?
Yes
No
SAL.01.03
Submit new AUM/BU2 Case based on Incident
SAL.01.04
Send Email to afftected account
End

4 Master Data

Master data management is not required for this project.
The report operates purely on **system logs**, **background job executions**, and **technical events**, without needing traditional SAP master data such as Customers, Vendors, or Materials.
This independence simplifies both initial implementation and ongoing data governance processes.

5 Technical Configuration via TCode ZG1CFG

To achieve **high flexibility**, **easy maintainability**, and **operational agility**, a dedicated technical configuration mechanism was developed.

5.1 Custom Configuration Table (Z-Table)

A **custom Z-table** was created to store all configurable parameters used by the report and monitoring programs.
This table eliminates hardcoding of critical values inside ABAP code and enables easy updates through standard SAP table maintenance functions.
Typical configurable parameters include:

- Monitoring algorithm flags
- Thresholds for system health indicators
- External API URLs
- Notification settings

5.2 Custom Transaction Code (TCode ZG1CFG)

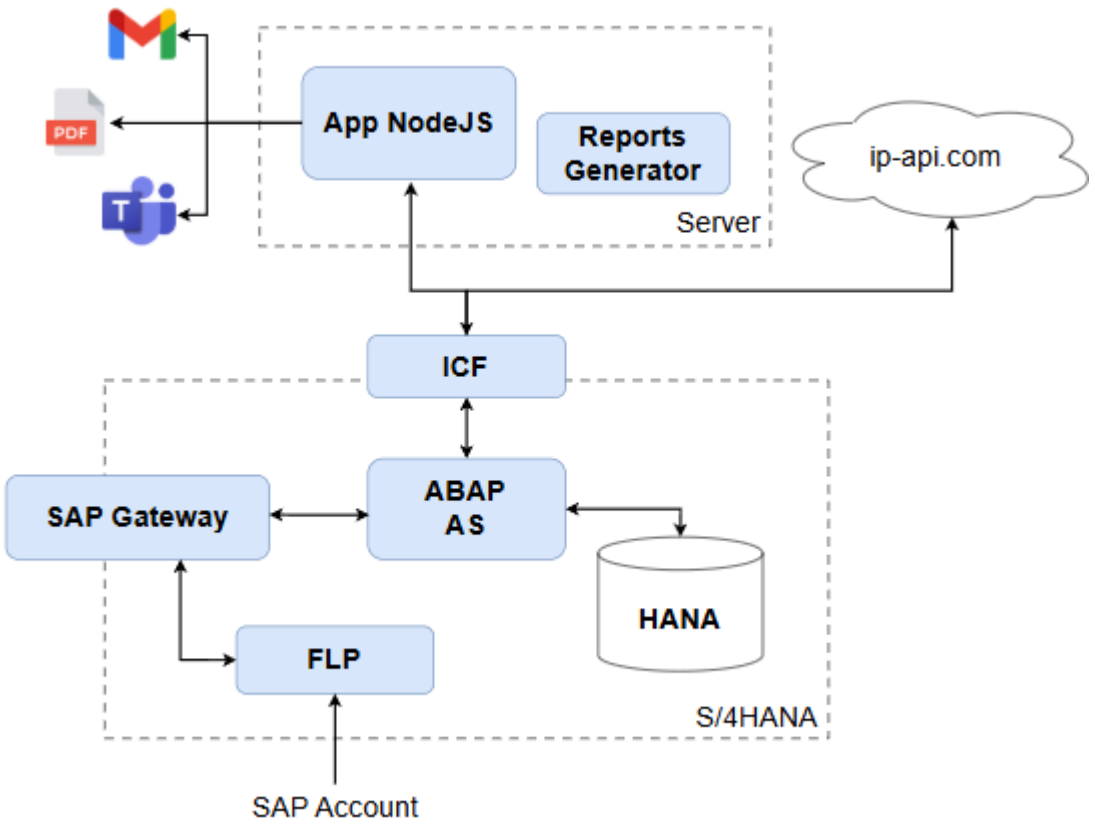
A custom **Transaction Code ZG1CFG** was developed to maintain the configuration table via a user-friendly SAP screen.
Key features of TCode ZG1CFG include:

- **Real-time parameter updates** without changing the source code
- **No need for transport requests** for routine configuration changes
- **Access control** to ensure that only authorized administrators can modify system settings

This approach embodies the modern SAP development best practice of **separating configuration from code**, leading to faster deployment cycles, lower maintenance risk, and higher system flexibility.

IV. SAP Coding

1. Application Architecture



The system integrates SAP S/4HANA with an external Node.js server.

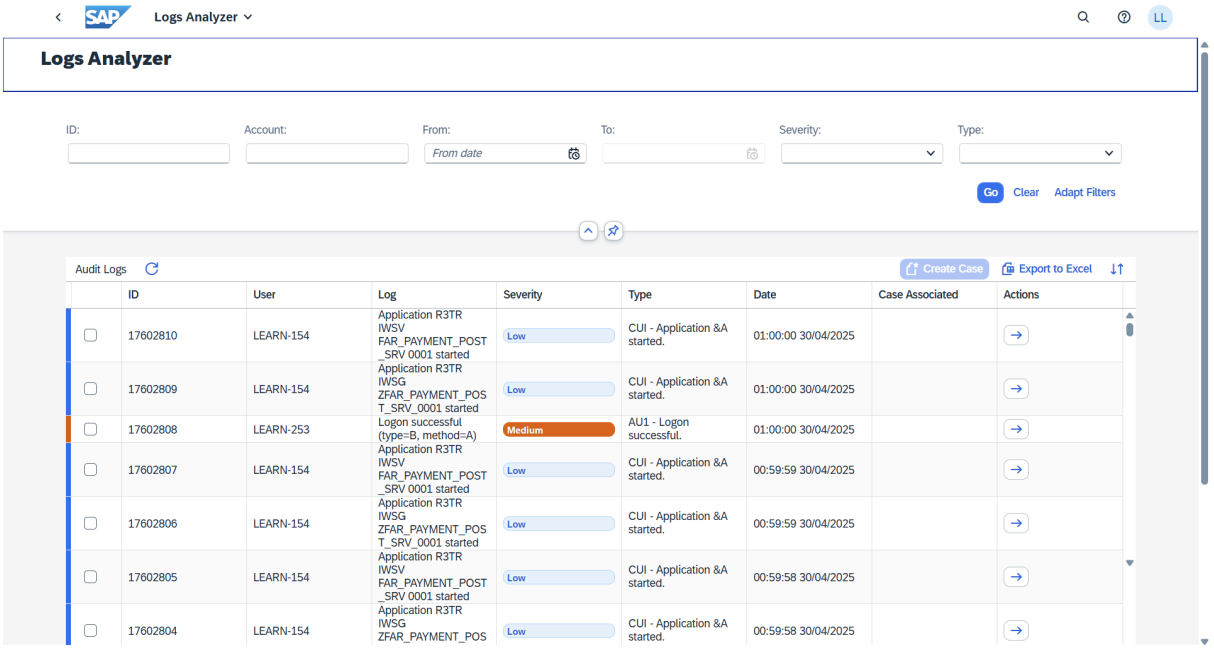
- **Frontend:** SAP Fiori Launchpad (FLP) allows users to access apps via SAP Gateway using their SAP accounts.
- **Backend (SAP):** ABAP Application Server (ABAP AS) handles core business logic and data operations via HANA DB. External communications go through ICF (Internet Communication Framework).
- **Backend (Node.js):** A standalone Node.js server connects to ABAP via ICF to retrieve SAP data and integrates with services like Gmail and Microsoft Teams.
- **External API:** The Node.js server also consumes external services such as ip-api.com to obtain IP-based geolocation data.

This architecture enables seamless integration between SAP systems and external platforms for enhanced business processes and notifications.

2. Functional/Technical Specifications

2.1 Logs Analyzer

2.1.1 Logs Analyzer



- **Function ID: ZG1_LOGTABLE.**
- **Function Name: *Logs Analyzer***
- **Purpose**: This function provides a real-time dashboard for system audit logs, offering users a clear and actionable overview of security-related events. It enables monitoring, alerting, and follow-up actions for suspicious log activities recorded from SAP SM20.
- **Functional Specification (FS)**
  - Display all logs or filter by user-defined conditions.
  - "Fetch" button to retrieve new logs from SM20.
  - On detecting a **password change**, system sends: Email, Teams message.
  - Allow selection of multiple suspicious logs (same user, unused in other cases) to open function ***CREATE_CASE***.
  - Double-click or click row arrow → open function ***Security Audit Log Detail.***
- **Technical Specification (TS)**
  - **Frontend (SAP Fiori UI5)**

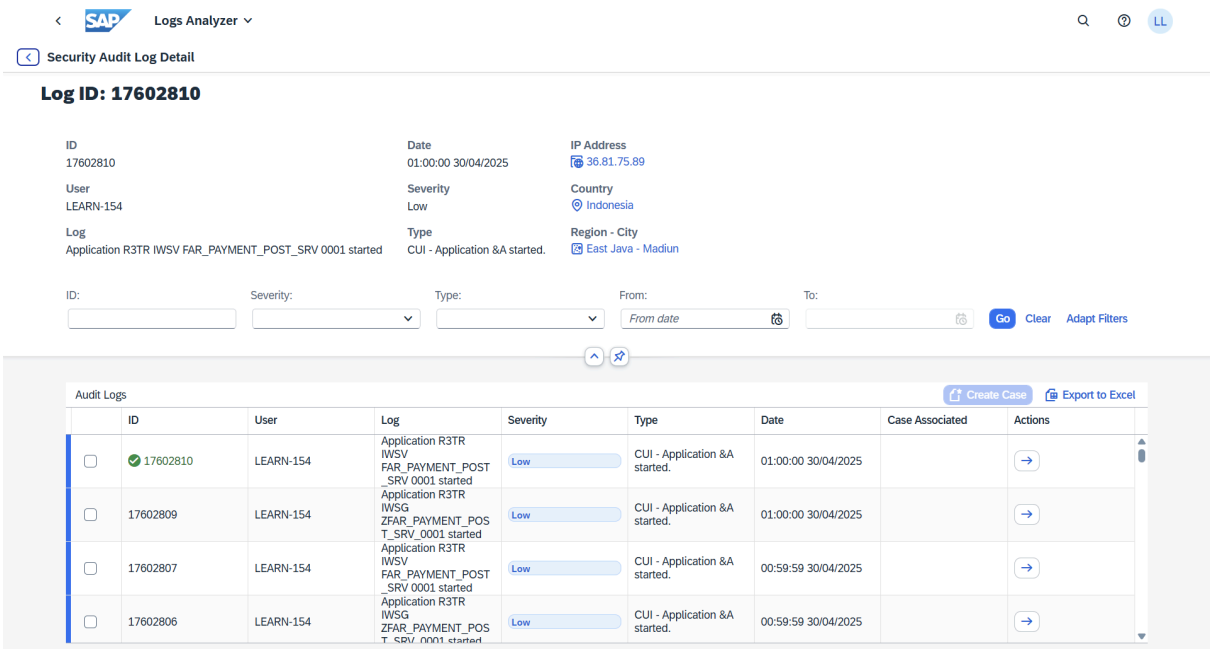
1. Logs table with filters and multi-select.

2. "Fetch" button triggers data reload.

3. Case creation via double-click or button.

- **Backend (SAP ABAP OData)**
  1. **SUBMIT FUNCTION zg1_salv –** Fetch newest data from SM20 and perform suspicious events analysys**.**
  2. **SALOGSET_GET_ENTITYSET -** fetch filtered logs from **ZL253_SALOG**
  3. **SALOGSET_GET_ENTITY (log_ID) -** to get entry from ZL253_SALOG table
  4. Detect password changes → trigger email & Teams alert via ICF.
- **External (Node.js Server)**
  1. Receives ICF call from SAP**.**
  2. Sends**:**
     a. **Email via Gmail API.**
     b. **Teams alert via Microsoft Teams API.**
  3. Optional geolocation using **sapbe.naucomnha.com.**

2.1.2 Audit Log Detail



- **Function ID**: **ZG1_LOGTABLE**

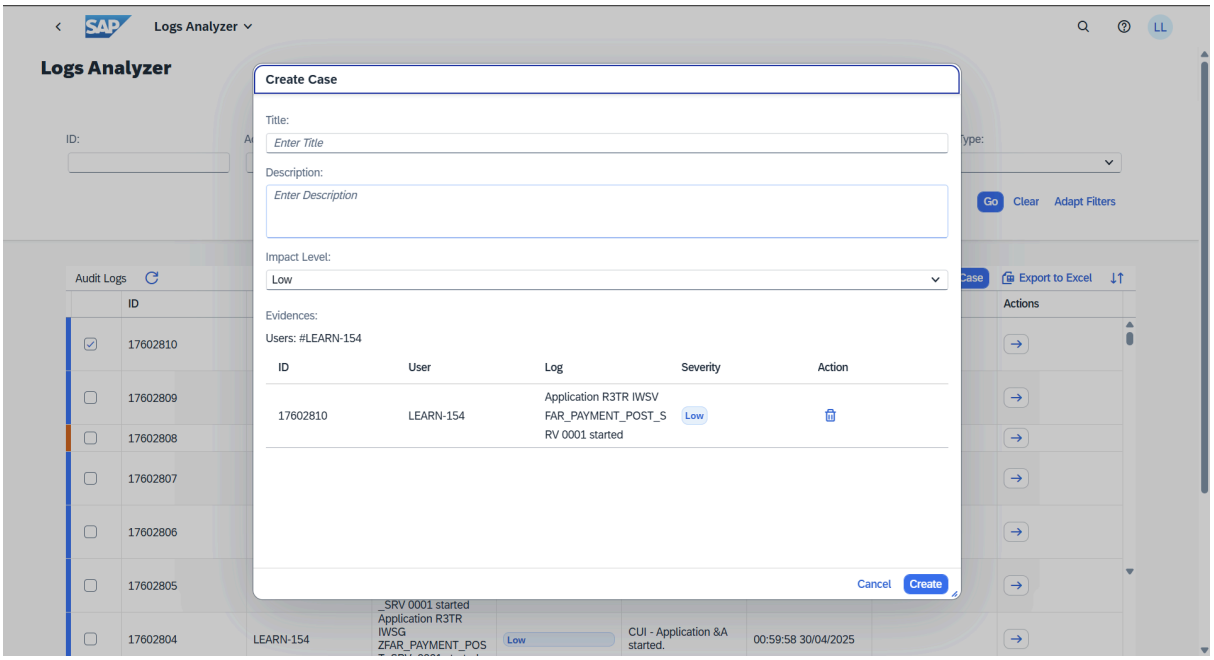
- **Function Name**: ***Security Audit Log Detail***

- **Purpose:** Provides detailed insights into a specific audit log entry, including related user logs, and supports case creation and export for further analysis or reporting.

- **Functional Specification (FS)**
  - Display general information of a selected log from ***Logs Analyzer***.
  - Show all logs related to the same user, with optional filter conditions.
  - Allow selection of multiple logs for case creation.
  - Enable export of displayed logs to **Excel**.
  - Open ***CREATE_CASE*** when creating a case based on selected logs.

- **Technical Specification (TS)**
  - **Frontend (SAP Fiori UI5)**
    1. Display log details (user, timestamp, event type, severity).
    2. Show related logs with filters (e.g., user, date).
    3. Multi-select logs to trigger case creation via modal **(*CREATE_CASE*).**
    4. Export log table to Excel format for download.
  - **Backend (SAP ABAP OData)**
    1. **METHOD SALOGSET_GET_ENTITYSET** – to fetch relevant logs from ZL253_SALOG table.
    2. **METHOD SALOGSET_GET_ENTITY (user_id=...)** – retrieve related user logs.

2.1.3 Audit Logs Create Case



- **Function ID**: **ZG1_LOGTABLE**

- **Function Name**:***Create Case***
- **Purpose**: Enable users to create a security case from selected suspicious logs. This feature provides tools to evaluate, describe, and submit a new case with appropriate severity and context.on predefined rules for logging events that require attention.

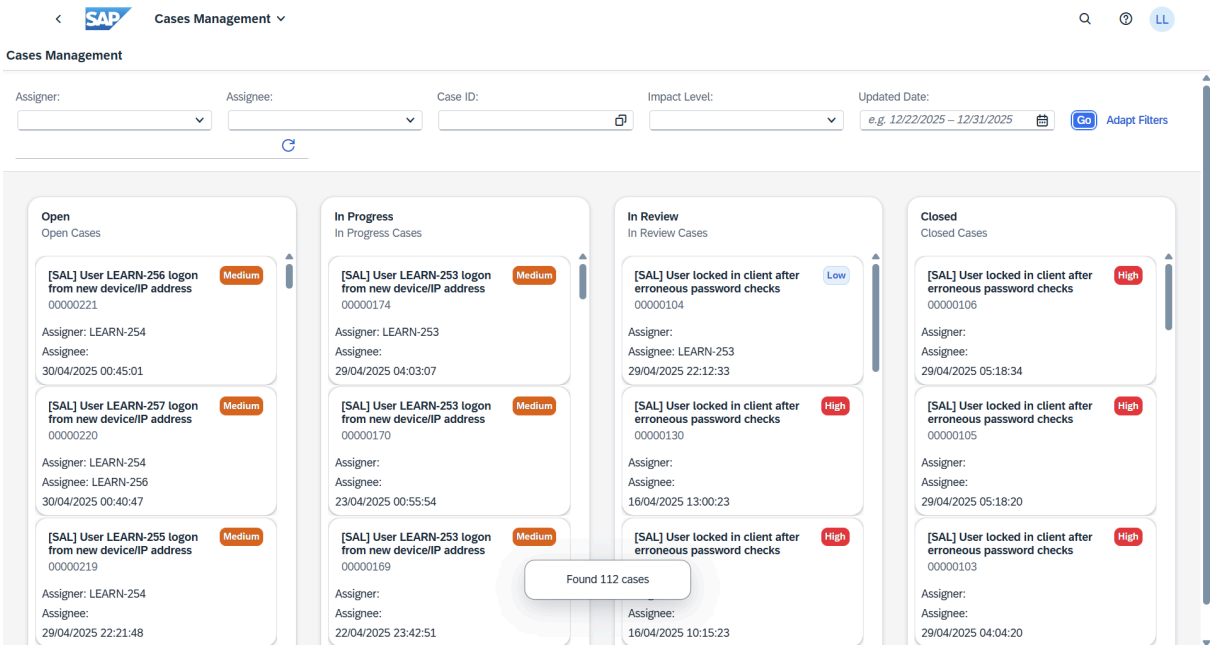
- **Functional Specification (FS)**
  - Display list of selected logs in the Evidences table.
  - Allow users to:
    1. Evaluate and remove logs (close modal if none left).
    2. Input **Title** and **Description** for the case.
    3. Select **Incident Impact Level**.
  - Submit data to create a new case based on provided inputs and selected logs.
- **Technical Specification (TS)**
  - **Frontend (SAP Fiori UI5)**
    1. Render list of logs as evidences.
    2. Provide input fields for: Title, Description, Impact Level (dropdown or radio buttons).
    3. Auto-close modal if all logs are removed.
    4. Trigger case creation API on submit.
  - **Backend (SAP ABAP OData)**
    1. **Method EVIDENCESET_CREATE_ENTITY**

a. Accepts log list, title, description, and impact level.

b. Validates input and inserts new record into **ZL253_CASES**.

c. Links selected logs as evidences to the case

2.2 Cases Managements



2.2.1 Audit Logs Cases Management

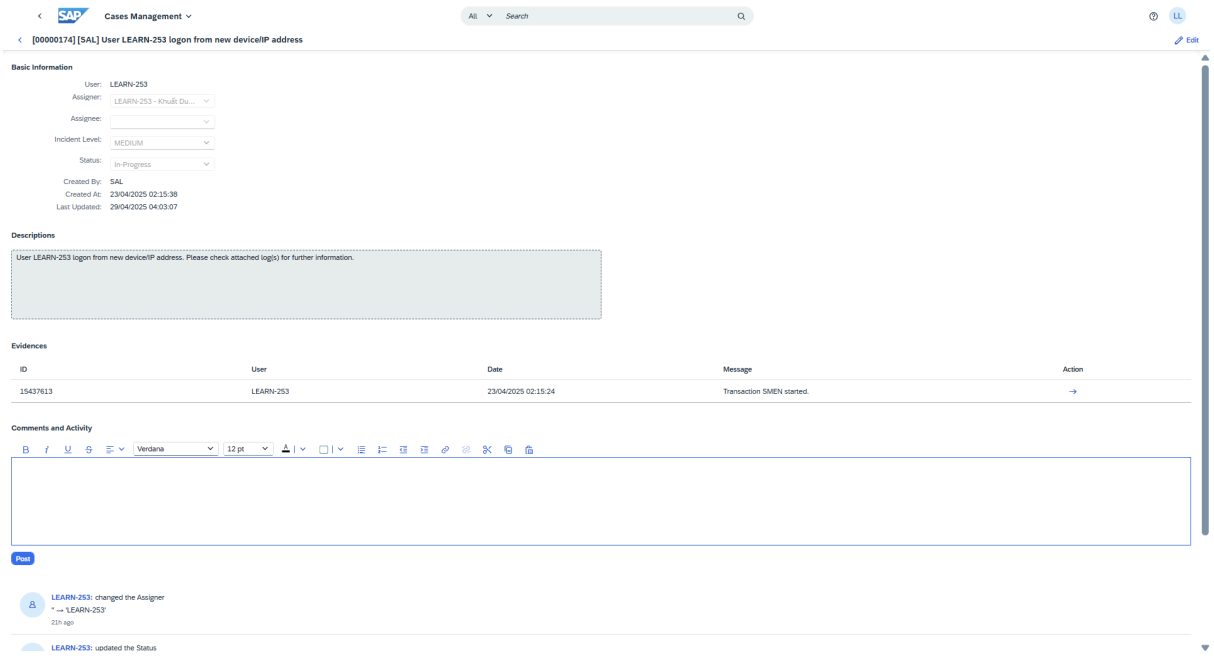
- **Function ID**: **ZG1_CASETABLE**
- **Function Name**: **Cases Managements**
- **Purpose**: Provides an overview and tracking system for all security cases. Supports filtering, viewing, and navigating to detailed case statuses. Enables users to monitor progress and manage lifecycle of cases.
- **Functional Specification (FS)**
  - Display all cases with statuses: Open, In-Progress, In-Review, Re-Open, and Closed.
  - Allow filtering by: Case ID, Status, Created Date (From/To)
  - Fetch and display cases from table ZL253_CASES.
  - Enable navigation to ZSALV_CASES_DETAILS by clicking on a case.
- **Technical Specification (TS)**
  - **Frontend (SAP Fiori UI5)**
    1. Collect filters via input form.
    2. On "Fetch", call OData service **/ZSALV_CASES_SRV/GET_CASES.**
    3. Display results in SmartTable with dynamic sorting/filtering.

4. Navigate to ***Security Case Details*** when user clicks on a case row or tab.

- **Backend (SAP ABAP OData)**
  1. **OData /CaseSet** – receive filters, query **ZL253_CASES** table.

  2. Data includes case metadata such as ID, status, creation date, and assigned user.

2.2.2 Audit Logs Case Detail



- **Function ID**: **ZG1_CASETABLE**
- **Function Name**: ***Security Case Details***
- **Purpose**: Provide full visibility into individual security cases derived from audit logs. Enables users to review, update, and collaborate on case-related data such as evidences, impact, and comments.
- **Functional Specification (FS)**
  - Display case details using **case_id** from previous screen.

  - Show related evidences and comments.

  - Allow updates to**:** Impact level, Status, Evidences
- **Technical Specification (TS)**
  - **Frontend (SAP Fiori UI5)**
    1. Display full case info (status, priority, assignee).

    2. Allow dropdown updates (e.g., Impact Level, Status).

    3. Handle evidence list actions: add/remove/update.

4. Manage comment section (add, edit, delete).

5. Support attachment uploads via UI.

- **Backend (SAP ABAP OData)**
  1. **GET/ZSALV_CASE_SRV/GET_CASE_DETAIL(case_id=...)** – fetch case info.

  2. **Method CASESET_UPDATE_ENTITY** – update impact level or status.

  3. **Method SALOGSET_GET_ENTITY** – modify evidences.

  4. **Method CASECMTSET_CREATE_ENTITY** – add new comments or attachments.

2.3 Dashboard

2.3.1 System Security Dashboard

- **Function ID: ZG1_DASHBOARD_1**
- **Function Name**: **System Security Dashboard**
- **Purpose**: This dashboard gives a high-level overview of system security by visualizing audit log patterns and case statuses. It helps monitor KPIs, identify trends in security events, and evaluate system behavior over time.
- **Functional Specification (FS)**
  - Event By Day: Show count of Medium and High severity events per day (from **ZL253_SALOG**).

  - Log Type: Show distribution of log entries by log type (from **ZL253_SALOG**).

  - Event By Time: Show Medium/High severity events by time range on a selected day.

  - Case Status: Show the total number of security cases by status (from **ZL253_CASES**).

  - Users can click Fetch to reload the data based on selected filters.

- **Technical Specification (TS)**
  - **Frontend (SAP Fiori UI5)**

1. Collects filters: Date From/To, Log Type, and Time Range.

2. Calls OData service /ZSALV_DASHBOARD_SRV/GET_DASHBOARD_DATA.

3. Displays data in visual charts (vizFrame):
    a. **Event By Day:** Daily event count by severity.

    b. **Log Type:** Pie/bar chart showing log type distribution.

    c. **Event By Time:** Hourly breakdown of event severity.

    d. **Case Status:** Summary of cases grouped by status.

- **Backend (OData - SAP)**
    1. **GET /ZSALV_LOG_SRV/SalogTypeCountSet(StartDate, EndDate)**
    → Returns log type counts in date range.

    2. **GET /ZSALV_LOG_SRV/CaseStatusCountSet**
    → Returns total cases grouped by status.

    3. **GET /ZSALV_LOG_SRV/SalogCountSet(StartDateTime, EndDateTime, Step)**
    → Returns event count by severity across time intervals.

    4. **POST /ZSALV_LOG_SRV/TriggerZG1SALV**
    → Triggers background processing for security logs or case summaries.

    5. All responses are in OData-compliant JSON format for visualization.

2.4 Emails

2.4.1 Audit Logs Send Notifications

- **Function ID**: **ZG1_EMAIL**
- **Function Name**: **Emails**
- **Purpose**: This module allows users to view, filter, and analyze automated emails sent by the SALV system. It provides insight into communication related to suspicious activities, including the email content and associated case links.
- **Functional Specification (FS)**
    - Display a list of system-generated emails related to logs and cases.

    - Allow users to apply filters to narrow down email records (by date, case ID, etc.).

    - Click on an email row to:

- **Technical Specification (TS)**
  - **Frontend (SAP Fiori UI5)**
    1. **Render a table of email logs retrieved from the backend.**
    2. **Provide filter fields in the table header for quick search.**
    3. **Handle row click to:**
       a. Show full email content and its template.
       b. Navigate to the detailed case in function ***Security Case Details*** view if associated with a case.
  - **Backend (SAP ABAP OData)**
    1. **GET /ZSALV_EMAIL_SRV/GET_EMAILS**
       a. Fetch email logs from table **ZL253_EMAILS** using input filter parameters.
       b. Return full email content and metadata including related case reference (if any).

Brief Explanation of Each Specification:

- **Audit Log Dashboard**: This is the central feature for visualizing and managing all audit logs in a consolidated interface, providing insights into system events. It is crucial for administrators to get a quick view of the system's security status.

- **Audit Log Detail**: It complements the dashboard by offering a detailed view of individual logs, allowing security analysts and IT staff to investigate specific incidents more deeply.

- **Audit Logs Case Detail**: This module helps manage cases that are generated from suspicious logs, giving users the ability to view the entire case's history and status, making it easier for teams to track and resolve issues.

- **Audit Logs Cases Management**: The case management feature enables efficient tracking of all open cases, ensuring that they are being handled properly. It also allows for escalation and reassignment if necessary.

- **Audit Logs Create Case**: This function streamlines the case creation process by automatically creating cases based on system-generated alerts, reducing the manual workload for security personnel.

- **Audit Logs Send Notifications**: This ensures that relevant stakeholders are alerted promptly via email when suspicious activities are detected, allowing them to respond quickly.

- **System Security Dashboard**: This high-level monitoring tool helps executives and IT managers track overall security trends, manage incidents, and view system performance metrics in real time.

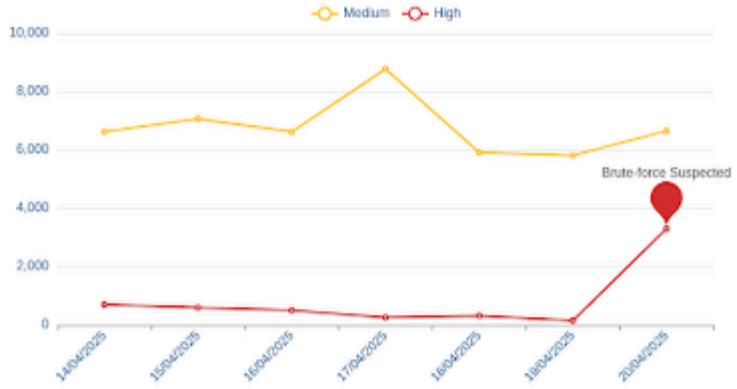
3. Reports

- **Weekly Report**: Generates a summary of suspicious logs and activities, to be sent to the CEO for review.

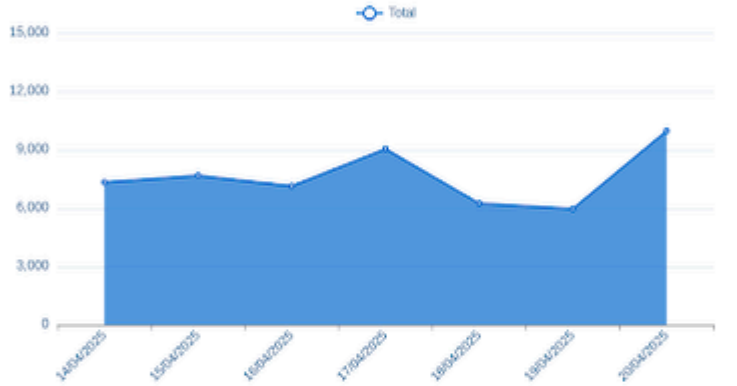
SAL Reports - Week 3, April 2025

Report generated on 29/04/2025 at 20:47:34

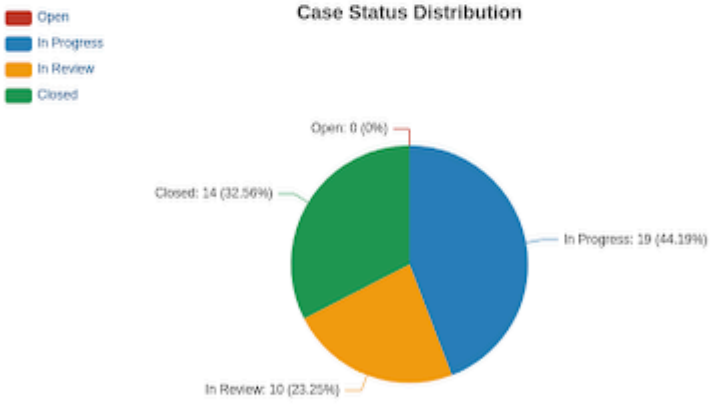
Daily Event Severity Counts



Daily Total Event Counts



Case Status Distribution



Incident Counts by Account

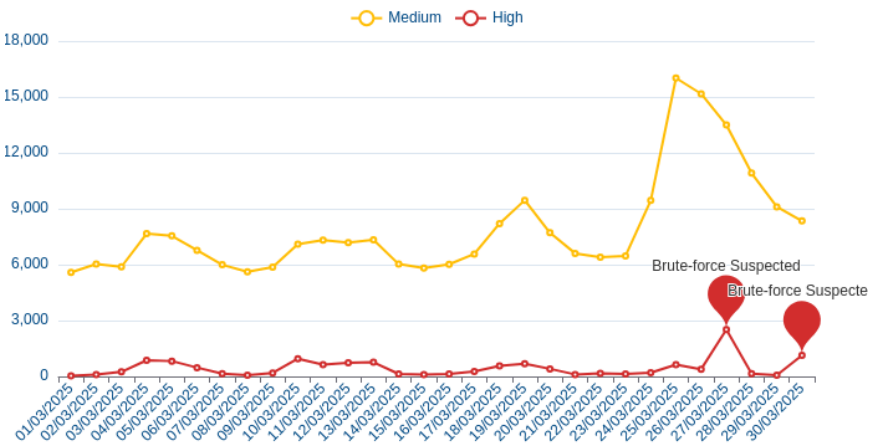
Account	Incident Count
LEARN-120	6
LEARN-060	3
LEARN-062	3
LEARN-156	2
LEARN-145	2
LEARN-817	1

- **Monthly Report**: Provides a detailed analysis of all security-related activities, helping management track system performance and compliance.

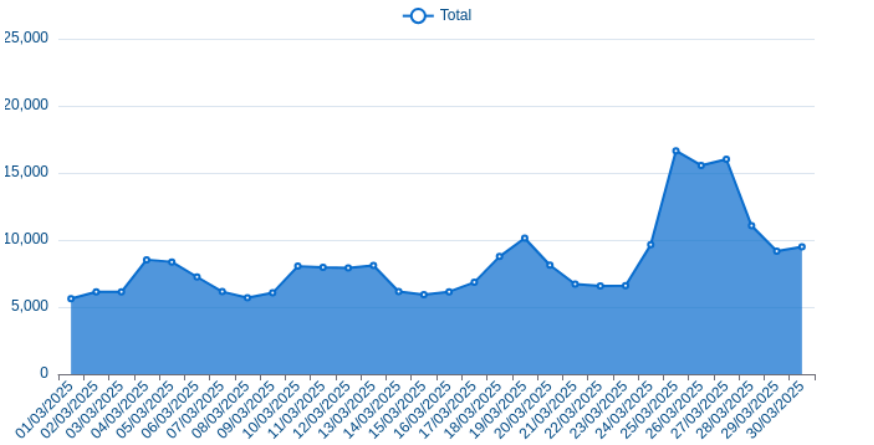
SAL Reports - March 2025

Report generated on 29/04/2025 at 20:41:08

Daily Event Severity Counts



Daily Total Event Counts



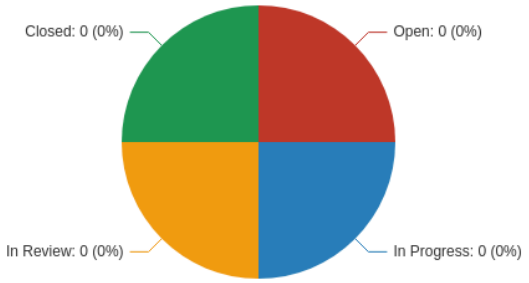
Case Status Distribution

Open

In Progress

In Review

Closed



Incident Counts by Account

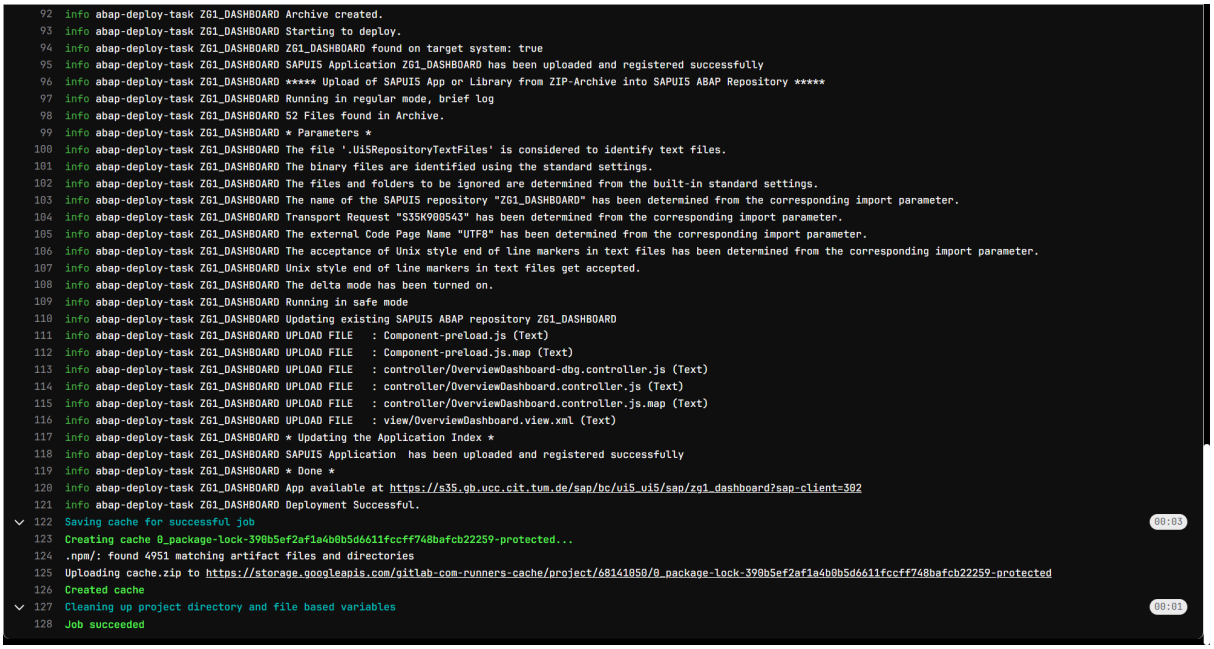
Account	Incident Count

4. Deployment

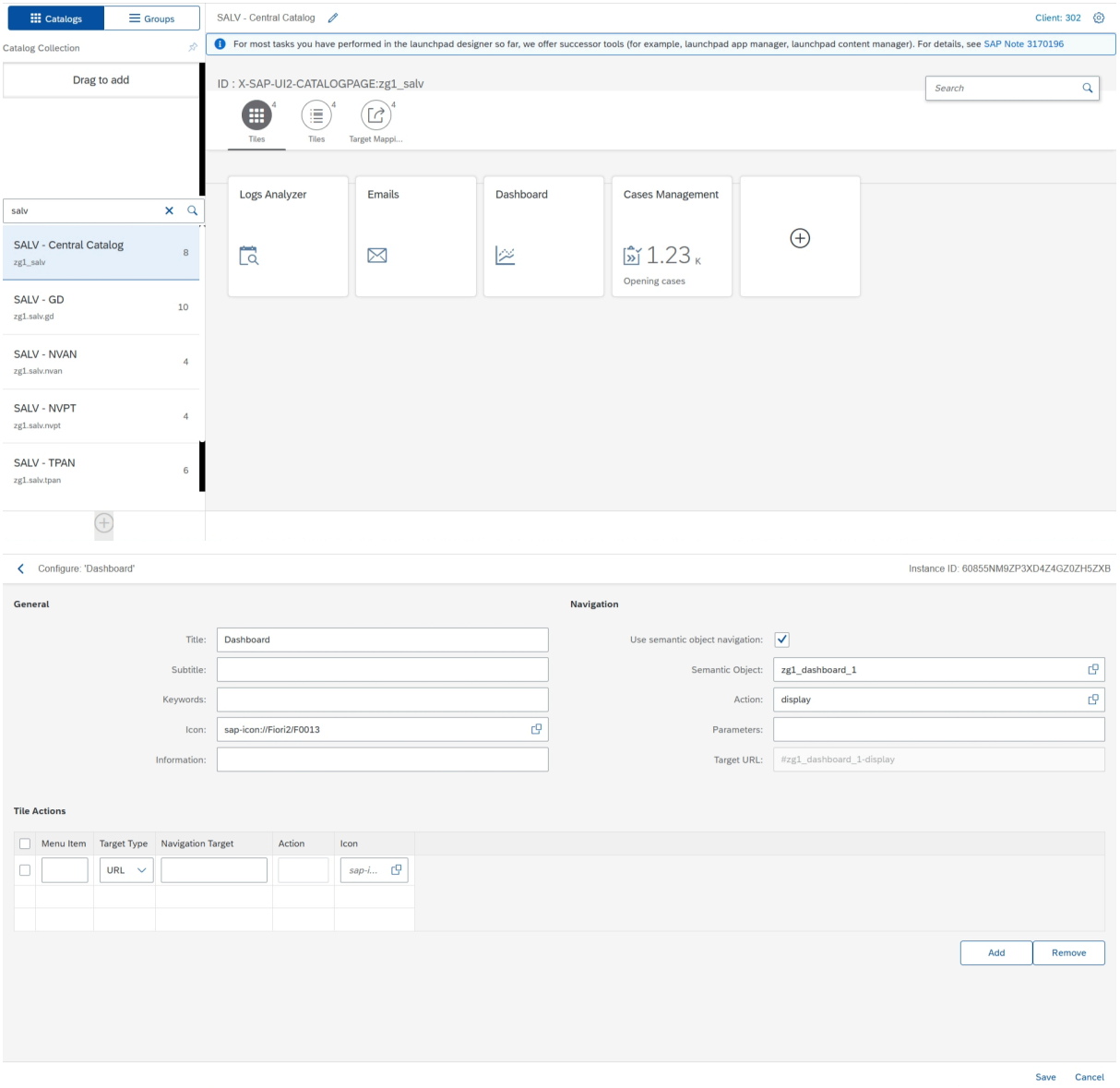
4.1. SAP Fiori

The SAPUI5 Fiori Application will be deployed on SAP BSP, then navigated to SAP FLP using Catalogs as Tiles (Applications). Users with authorizations will be managed by FLP

Run **npm run deploy** inside source code to deploy to BSP



Deploy to FLP using FLP Designer Catalog and Tiles



4.2. SAP ABAP

The ABAP Backend will be deployed on SAP S4/HANA, behind SAP Gateway

Managed automatically by SAP S4HANA. No manual deployment needed

4.3. NodeJS Backend

.gitlab-ci.yml

```
Cody
stages:
  - deploy

Cody
deploy_to_server:
  stage: deploy
  script:
    - 'which ssh-agent || ( apt-get update -y && apt-get install openssh-client -y )'
    - eval $(ssh-agent -s)
    - echo "$SSH_PRIVATE_KEY" | tr -d '\r' | ssh-add -
    - mkdir -p ~/.ssh
    - chmod 700 ~/.ssh
    - ssh-keyscan -H $SERVER_IP >> ~/.ssh/known_hosts
    - ssh khuathieu@$SERVER_IP "
        . /home/khuathieu/.nvm/nvm.sh &&
        nvm use 20.18.0 &&
        cd /home/khuathieu/sap/send-mail/ &&
        git pull &&
        npm i &&
        pm2 restart sap-send-email --update-env
      "
  only:
    - main
```

The NodeJS Backend application will be deployed on an external server. Can be accessed using URL sap-be.naucomnha.com

Inside server, clone the source code from repository, then execute npm i && pm2 start

```
routes/emailRoutes.js                 |   2 +
service/validateRequest.js            |   1 +
templates/caseLevelChange.ejs         | 192 ++++++++++++++++++++++++++++++
templates/newCase.ejs                 |   4 +-
utils/urlAPI.js                       |   1 +
6 files changed, 242 insertions(+), 2 deletions(-)
create mode 100644 controllers/caseLevelChangeController.js
create mode 100644 templates/caseLevelChange.ejs
up to date, audited 241 packages in 980ms
37 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
[PM2] Applying action restartProcessId on app [sap-send-email](ids: [ 13, 14, 15, 16, 17, 18 ])
[PM2] [sap-send-email](13) ✓
[PM2] [sap-send-email](14) ✓
[PM2] [sap-send-email](15) ✓
[PM2] [sap-send-email](16) ✓
[PM2] [sap-send-email](17) ✓
[PM2] [sap-send-email](18) ✓
```

id	name	namespace	version	mode	pid	uptime	↺	status	cpu	mem	user	watching
0	comnha-api	default	0.0.1	cluster	1448904	3h	717	online	0%	121.1mb	khuathi…	disabled
1	comnha-api	default	0.0.1	cluster	1502364	10m	629	online	0%	110.6mb	khuathi…	disabled
2	comnha-api	default	0.0.1	cluster	1502209	11m	721	online	0%	112.5mb	khuathi…	disabled
3	comnha-api	default	0.0.1	cluster	1502457	8m	633	online	0%	112.6mb	khuathi…	disabled
4	comnha-api	default	0.0.1	cluster	1502755	3m	716	online	0%	114.2mb	khuathi…	disabled
5	comnha-api	default	0.0.1	cluster	1440964	3h	713	online	0%	123.1mb	khuathi…	disabled
13	sap-send-email	default	1.0.0	cluster	1503268	0s	127	online	0%	72.5mb	khuathi…	disabled
14	sap-send-email	default	1.0.0	cluster	1503269	0s	127	online	0%	76.8mb	khuathi…	disabled
15	sap-send-email	default	1.0.0	cluster	1503296	0s	127	online	0%	62.5mb	khuathi…	disabled
16	sap-send-email	default	1.0.0	cluster	1503308	0s	127	online	0%	57.3mb	khuathi…	disabled
17	sap-send-email	default	1.0.0	cluster	1503324	0s	127	online	0%	47.9mb	khuathi…	disabled
18	sap-send-email	default	1.0.0	cluster	1503337	0s	127	online	0%	41.7mb	khuathi…	disabled

```
Cleaning up project directory and file based variables
Job succeeded
```

00:01

V. Notification System

The Node.js server uses **Nodemailer** to send notifications. The server is configured to handle various types of email and message, such as **incident reports**, **case status changes**, and more. Each API has a specific template for sending emails using **SMTP**.

1. API Specifications for Sending Notifications

The server exposes several APIs that trigger email notifications based on specific events. These events include reporting incidents, creating new cases, changing case statuses, assigning new users, and adding new comments.

Each API corresponds to a particular action, such as sending an **incident report** or notifying about a **new case**. The **level** parameter (with values **High**, **Medium**, **Low**) is used to indicate the severity of the alert in certain APIs, like incident reporting and case updates.

Here are the APIs available for sending emails:

1.1 New Case Notification

- **Email**

  Endpoint: /notification/new-case
  Description: Sends an email notifying the creation of a new case.
  Json Example:

  ```
  {
   "to": "recipient@example.com",
   "cc": ["cc1@example.com"],
   "userName": "Hieu",
   "caseId": "CASE002",
   "caseName": "New Case Test",
   "terminal": "Device123",
   "assignee": "John Doe",
   "level": "Medium",
   "timeExecuted": "2025-03-08T12:00:00"
  }
  ```

- **Message**
  - Send to user who created case

  API:

  https://prod-39.southeastasia.logic.azure.com:443/workflows/a2795dbcc9584f00bbd4cd77827f4282/triggers/manual/paths/invoke?api-version=2016-06-01&sp=%2Ftriggers%2Fmanual%2Frun&sv=1.0&sig=zthh9BbQwNtRzCZNNp6vsG1ONPjtmgCQVtLZzXn1SNE

  Json Schema:

  Request Body JSON Schema

```
{
    "type": "object",
    "properties": {
        "emailUser": {
            "type": "string"
        },
        "userName": {
            "type": "string"
        },
        "title": {
            "type": "string"
        },
        "descriptions": {
            "type": "string"
        },
        "caseId": {
            "type": "string"
        }
    }
}
```

  - Send to user (role **Analysis**)

API:
https://prod-26.southeastasia.logic.azure.com:443/workflows/d8938eed51b54de6a785cb3b23f45171/triggers/manual/paths/invoke?api-version=2016-06-01&sp=%2Ftriggers%2Fmanual%2Frun&sv=1.0&sig=lZKwikIjippHpWLx_TsnJ2sN8eCeKwuOSnDl4FC5VOM

Json Schema:

```
{
    "type": "object",
    "properties": {
        "emailAnalyst": {
            "type": "string"
        },
        "analystName": {
            "type": "string"
        },
        "title": {
            "type": "string"
        },
        "descriptions": {
            "type": "string"
        },
        "caseId": {
            "type": "string"
        },
        "incidentImpact": {
            "type": "string"
        }
```

- Send to user (role **Manager**)

API:
https://prod-23.southeastasia.logic.azure.com:443/workflows/8e0757e124fd457f8f6ce14f56c12587/triggers/manual/paths/invoke?api-version=2016-06-01&sp=%2Ftriggers%2Fmanual%2Frun&sv=1.0&sig=rj2W25B-Dn37AWTk7q-URzHgFik8RXUqDFeLnJxWtPc

Json Schema:

1.2. Case Status Change

- **Email**

Endpoint: /notification/case-status-change

Description: Sends an email when the status of a case changes.

JSON Example:

{

"to": "recipient@example.com",

"cc": ["cc1@example.com"],

"userName": "Hieu",

"caseId": "CASE003",

"caseName": "Status Change Test",

"terminal": "192.168.1.1",

"oldStatus": "Open",

```
  "newStatus": "In Progress",
  "level": "Low",
  "timeExecuted": "2025-03-08T12:00:00"
}
```


- **Message**
  - **Description:** With each update case (like status, Incident-impact, new comment). The system will send message to users (role: **Analysis**, **Security**)

    API:
    https://prod-65.southeastasia.logic.azure.com:443/workflows/fb8512821f1442a599b86333049f21db/triggers/manual/paths/invoke?api-version=2016-06-01&sp=%2Ftriggers%2Fmanual%2Frun&sv=1.0&sig=WqEUu1i2GRJut703BQ3Bji9BSDv7EeEGEjOsSojVifc

    Json Schema:


```
{
  "type": "object",
  "properties": {
    "emailTo": {
      "type": "string"
    },
    "name": {
      "type": "string"
    },
    "title": {
      "type": "string"
    },
    "caseId": {
      "type": "string"
    },
    "typeUpdated": {
      "type": "string"
```

```
    },
    "oldValue": {
      "type": "string"
    },
    "newValue": {
      "type": "string"
    }
  }
}
```

1.3. Change Assignee

- **Email**

Endpoint: /notification/change-assignee

Description: Sends an email when a case assignee is changed.

JSON Example:

```
{
  "to": "recipient@example.com",
  "cc": ["cc1@example.com"],
  "userName": "Hieu",
  "caseId": "CASE004",
  "caseName": "Assignee Change Test",
  "terminal": "Device123",
  "oldAssignee": "John Doe",
  "newAssignee": "Jane Smith",
```

```
  "level": "Medium",

  "timeExecuted": "2025-03-08T12:00:00"

}
```


- **Message**
  - **Description:** With each update case (like status, Incident-impact, new comment). The system will send message to users (role: **Analysis**, **Security**)

    API:
    https://prod-65.southeastasia.logic.azure.com:443/workflows/fb8512821f1442a599b86333049f21db/triggers/manual/paths/invoke?api-version=2016-06-01&sp=%2Ftriggers%2Fmanual%2Frun&sv=1.0&sig=WqEUu1i2GRJut703BQ3Bji9BSDv7EeEGEjOsSojVifc

    Json Schema:


```
{

  "type": "object",

  "properties": {

    "emailTo": {

      "type": "string"

    },

    "name": {

      "type": "string"

    },

    "title": {

      "type": "string"

    },

    "caseId": {

      "type": "string"

    },

    "typeUpdated": {

      "type": "string"

    },
```

```
        "oldValue": {
          "type": "string"
        },
        "newValue": {
          "type": "string"
        }
      }
    }
```

1.4. New Comment

- **Email:**

Endpoint: /notification/new-comment

Description: Sends an email when a new comment is added to a case.

JSON:**Example**

```
{
  "to": "recipient@example.com",
  "cc": ["cc1@example.com"],
  "userName": "Hieu",
  "caseId": "CASE005",
  "caseName": "Comment Test",
  "terminal": "192.168.1.1",
  "comment": "This is a test comment",
  "commenter": "Hieu",
  "level": "Low",
  "timeExecuted": "2025-03-08T12:00:00"
}
```

- **Message:**

- **Description:** With each update case (like status, Incident-impact, new comment). The system will send message to users (role: **Analysis**, **Security**)

  API:
  https://prod-65.southeastasia.logic.azure.com:443/workflows/fb8512821f1442a599b86333049f21db/triggers/manual/paths/invoke?api-version=2016-06-01&sp=%2Ftriggers%2Fmanual%2Frun&sv=1.0&sig=WqEUu1i2GRJut703BQ3Bji9BSDv7EeEGEjOsSojVifc

  Json Schema:


```
{
  "type": "object",
  "properties": {
    "emailTo": {
      "type": "string"
    },
    "name": {
      "type": "string"
    },
    "title": {
      "type": "string"
    },
    "caseId": {
      "type": "string"
    },
    "typeUpdated": {
      "type": "string"
    },
    "oldValue": {
      "type": "string"
    },
    "newValue": {
```

```
            "type": "string"
        }
    }
}
```

1.5. Weekly report

Endpoint: /notification/weekly-report

Description: send email every week .

JSON:**Example**

```
{
  "to": "minhhieu2772003@gmail.com",
  "cc": ["cc1@example.com"],
  "userName": "Hieu",
  "CSVfile": "W3siaW5mbyI6IldlZWtseSByZXBvcnQifV0=",
  "newCases": 5,
  "inProgress": 3,
  "awaitingReview": 2,
  "completed": 10,
  "timeExecuted": "2025-03-08T12:00:00"
}
```


1.6. Month report

Endpoint: /notification/monthly-report

Description: send email every month.

JSON:**Example**

```
{
  "to": "minhhieu2772003@gmail.com",
  "cc": ["cc1@example.com"],
```

```
  "userName": "Hieu",
  "CSVfile": "W3siaW5mbyI6IldlZWtseSByZXBvcnQifV0=",
  "newCases": 5,
  "inProgress": 3,
  "awaitingReview": 2,
  "completed": 10,
  "timeExecuted": "2025-03-08T12:00:00"
}
```


monthly_report.png

---

2. Sending Notification via API

Each API sends an email and after that is a message based on the provided JSON input. Here is an example of how to set up and send an email using **Nodemailer** in a Node.js server.

Node.js Email Sending Example:

```
const nodemailer = require('nodemailer');

// Create a transporter for sending emails
const transporter = nodemailer.createTransport({
  service: 'gmail',
  auth: {
    user: 'your-email@gmail.com',
    pass: 'your-email-password'
  }
});
```

```
// Email options
const mailOptions = {
  from: 'your-email@gmail.com',
  to: 'recipient@example.com',
  subject: 'Alert: Suspicious Activity Detected',
  text: 'This is a notification about suspicious activity in the SAP system.'
};

// Send email
Logging writelog = ….;
transporter.sendMail(mailOptions, (error, info) => {
  if (error) {
    writelog.write(“Server send mail fails at: ” time.now());
    return response(500, “Server send mail fails”);
  } else {
    writelog.write(“Server send mail success at: ” time.now());
    this.messageSend(...options)
  }
});
```

The email is sent with the details provided in the JSON, and you can track the result through the console.

---

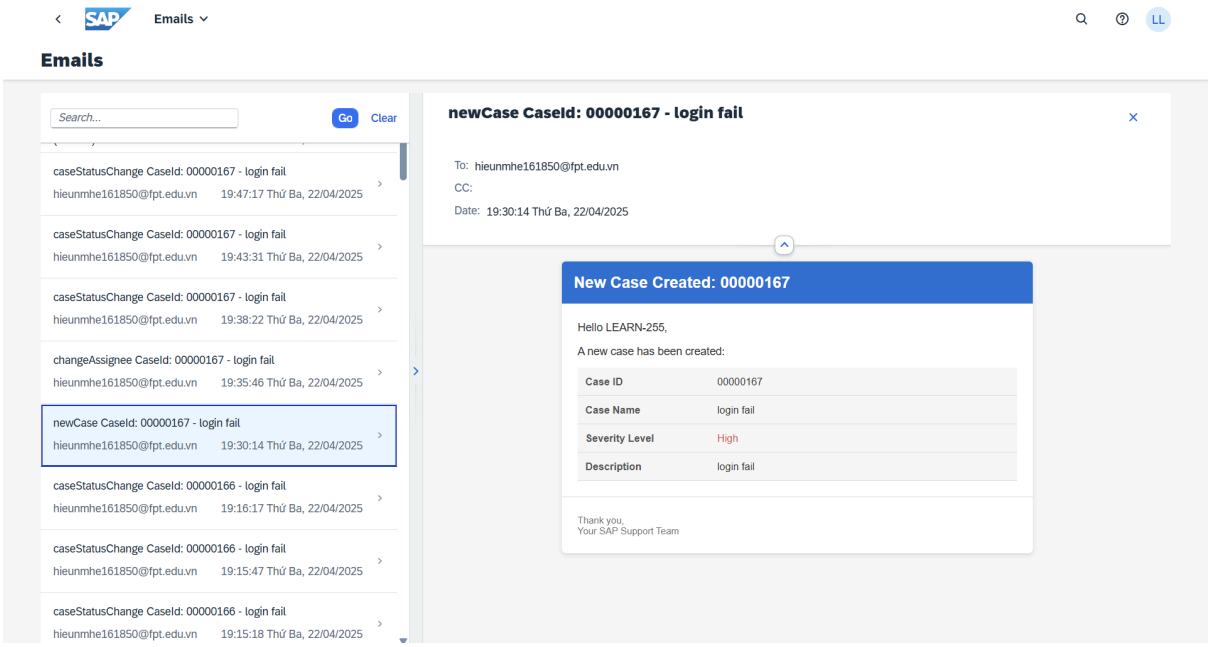
3. Expected Notifications Result

Upon successfully sending the email, the recipient will receive an email in HTML format. The email will contain details such as the case ID, status change, or incident information, depending on the API used.

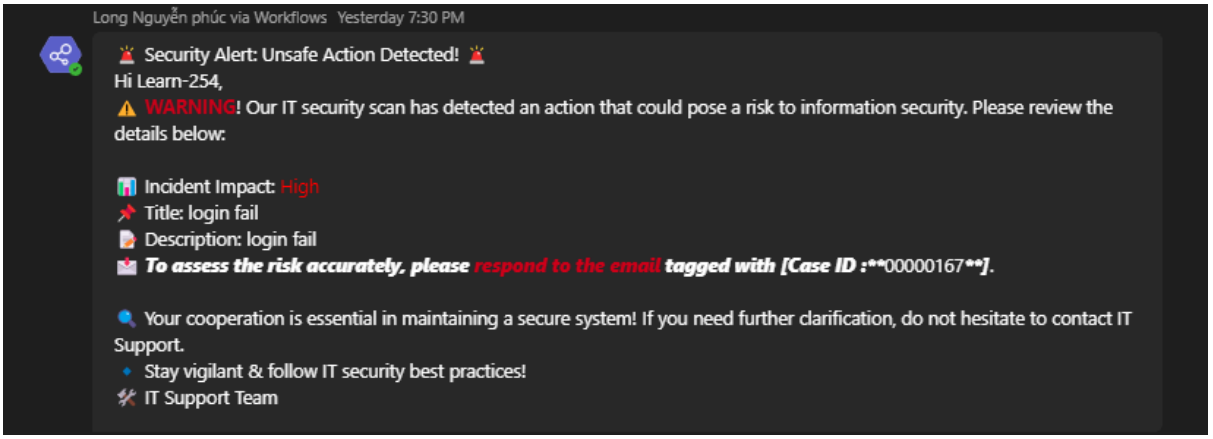
Here are some examples of the expected email content:

- **New Case Notification**:
    - **Subject**: "New Case Created: CASE002"
    - **Body**: Information about the new case, assignee, and description.

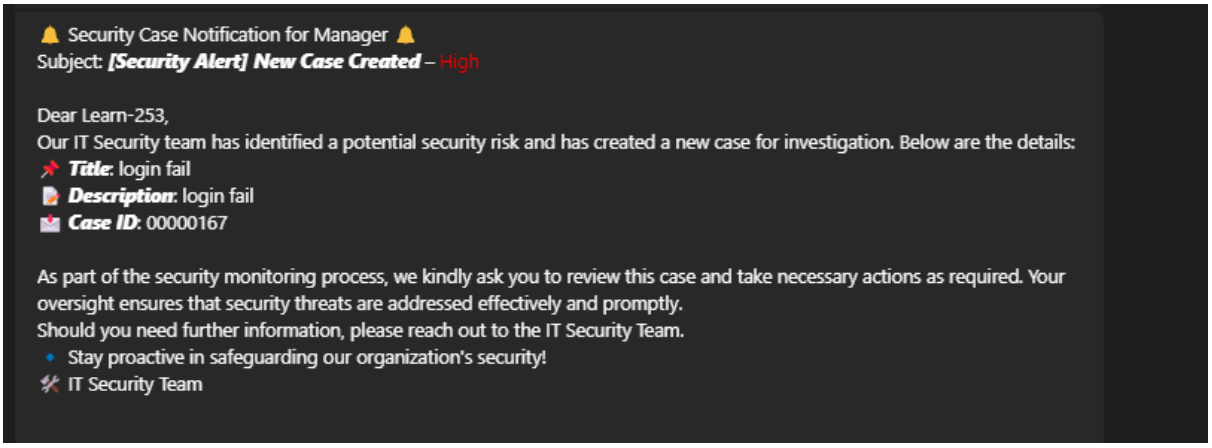
    - Email



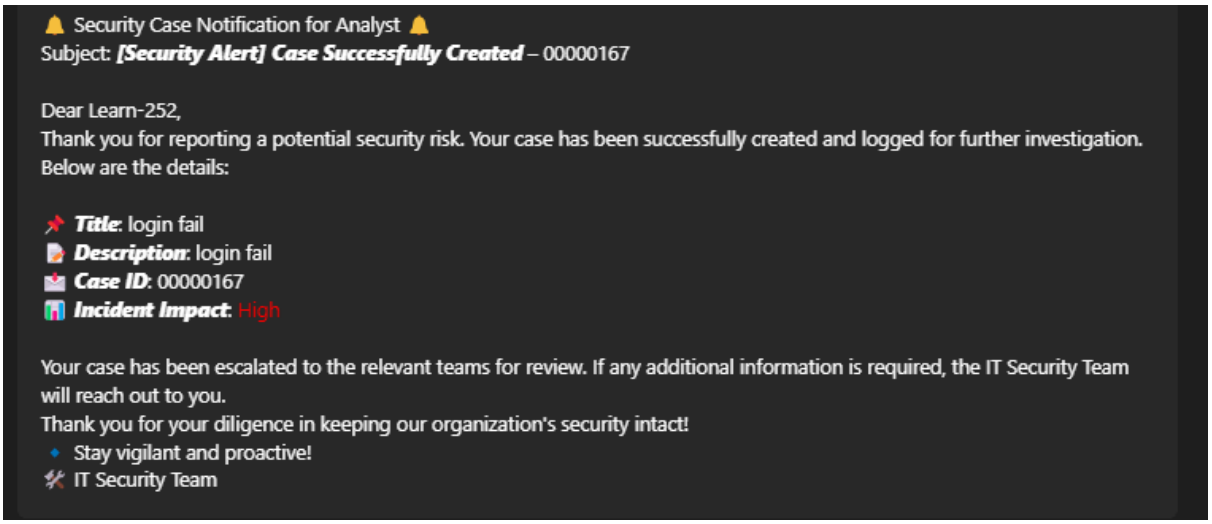
    - Message
        - User (Staff)



- User (Manager)



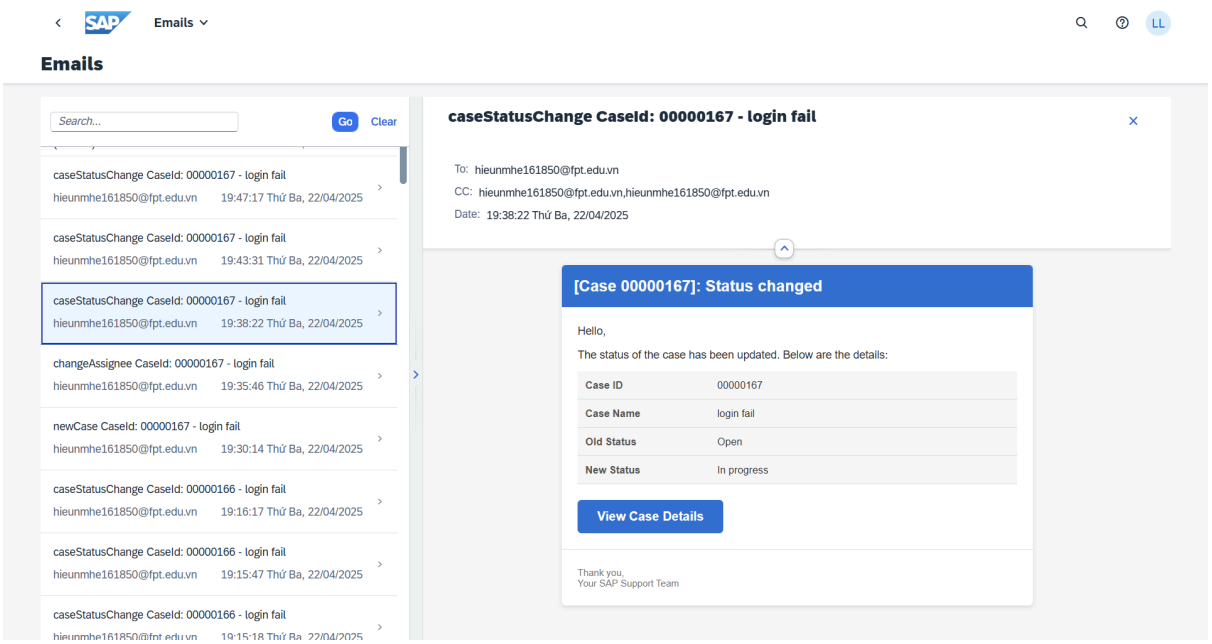
- User (Analyst)



- **Case Status Change**:

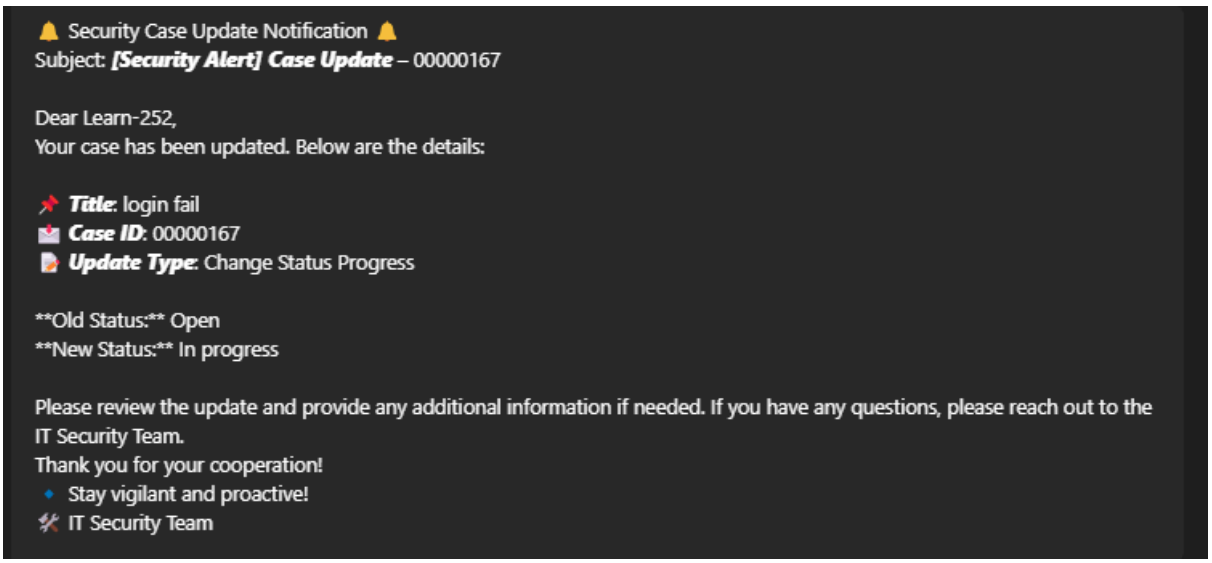
- **Subject**: "Case Status Change for CASE003"
- **Body**: Information about the case status change, including the old and new status.

❖ Email

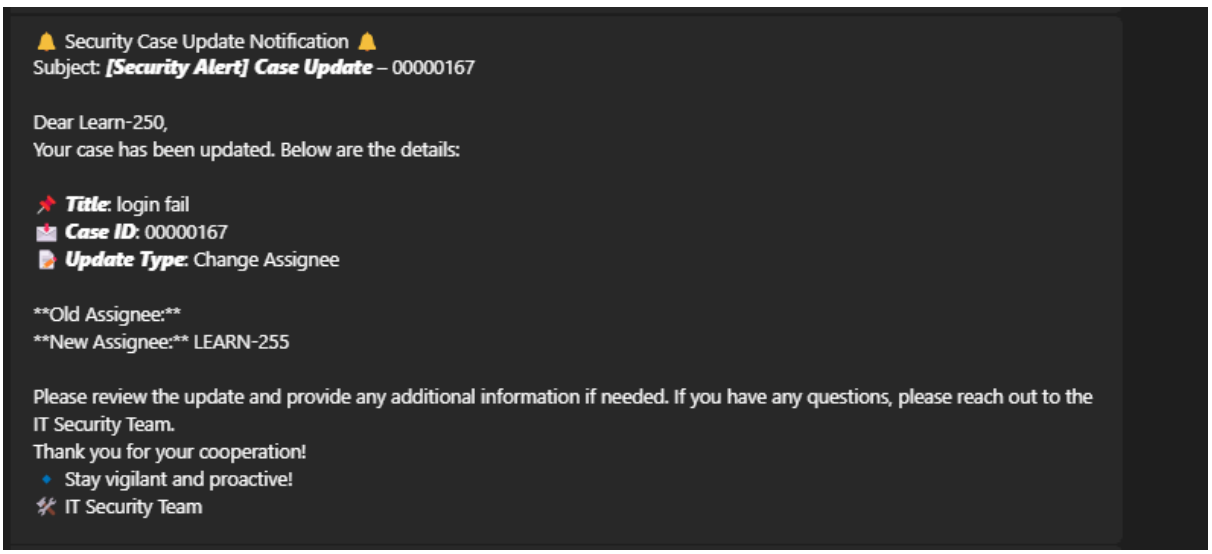


❖ Message

➢ User (**Analysis**)



➢ User (**Security A**)



➢ User (**Security B**)

🔔 Security Case Update Notification 🔔
Subject: ***[Security Alert] Case Update*** – 00000167

Dear Learn-251,
Your case has been updated. Below are the details:

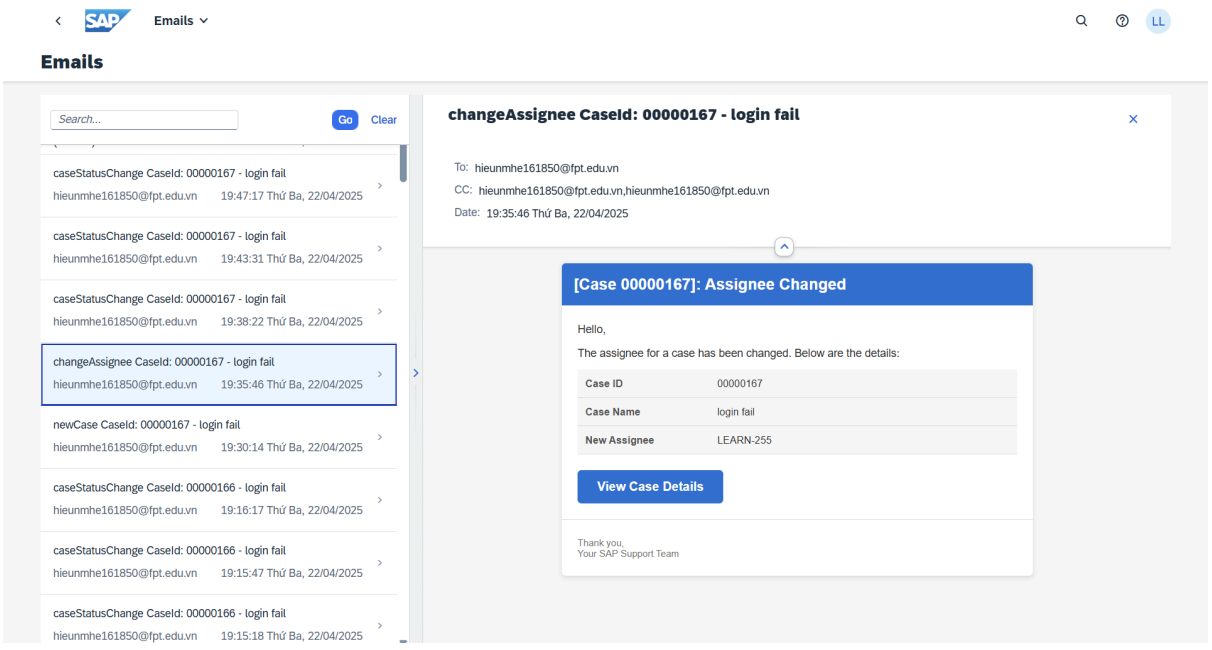
📌 ***Title***: login fail
📩 ***Case ID***: 00000167
📝 ***Update Type***: Change Status Progress

Old Status: Open
New Status: In progress

Please review the update and provide any additional information if needed. If you have any questions, please reach out to the IT Security Team.
Thank you for your cooperation!
🔹 Stay vigilant and proactive!
🛠️ IT Security Team

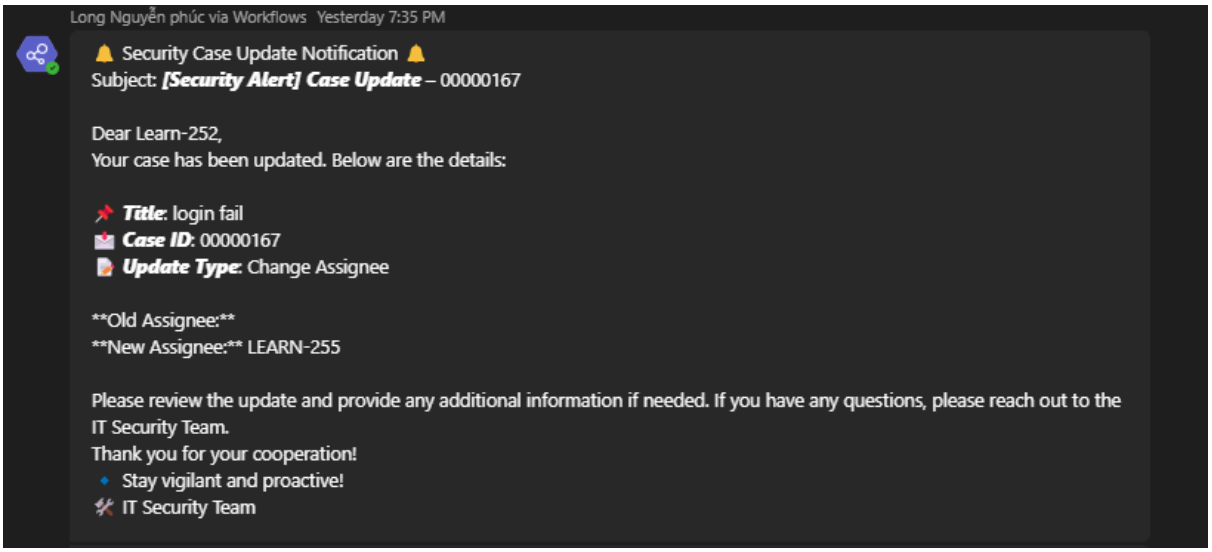
- **Change Assignee**:
    - **Subject**: "Case Assignee Changed: CASE004"
    - **Body**: Information about the case assignee change.

❖ Email

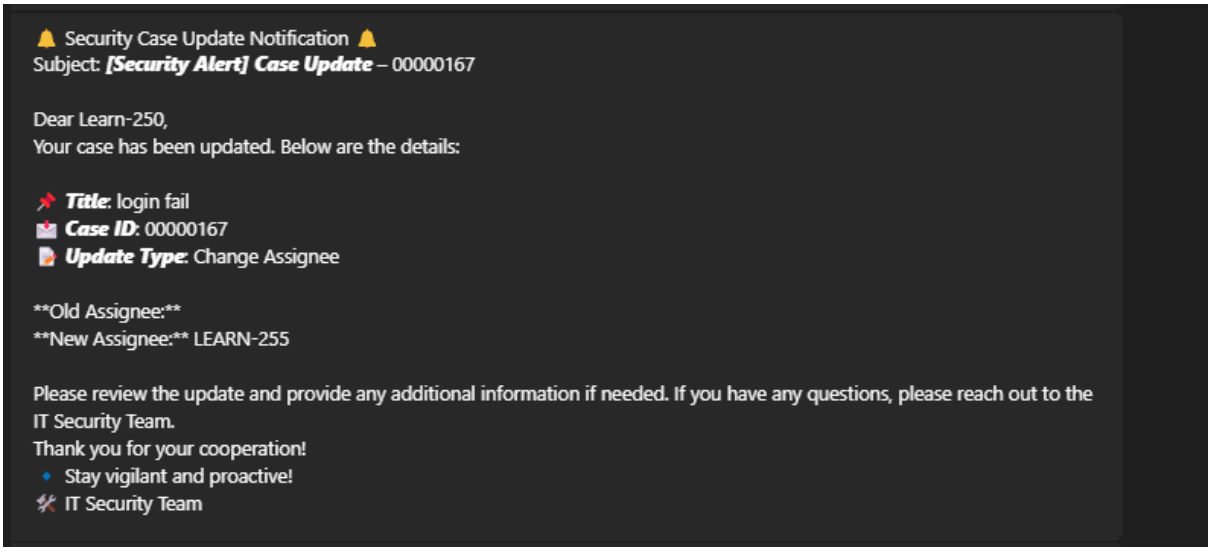


❖ **Message**

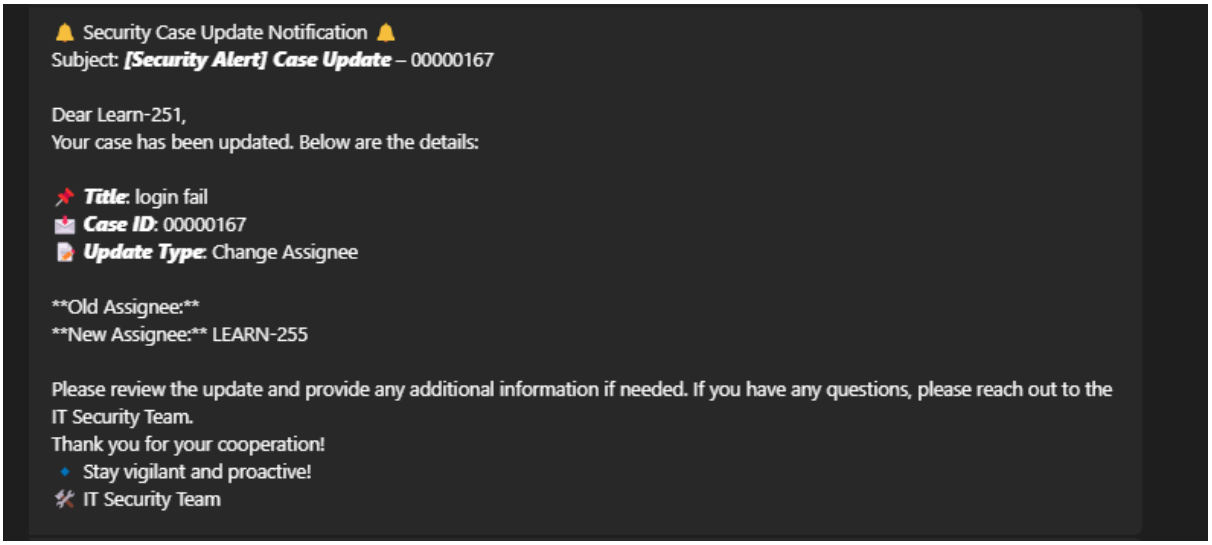
➢ User (**Analysis**)



➢ User (**Security A**)



➢ User (**Security B**)



- **New Comment**:
    - **Subject**: "New Comment on Case CASE005"
    - **Body**: The comment text and commenter information.

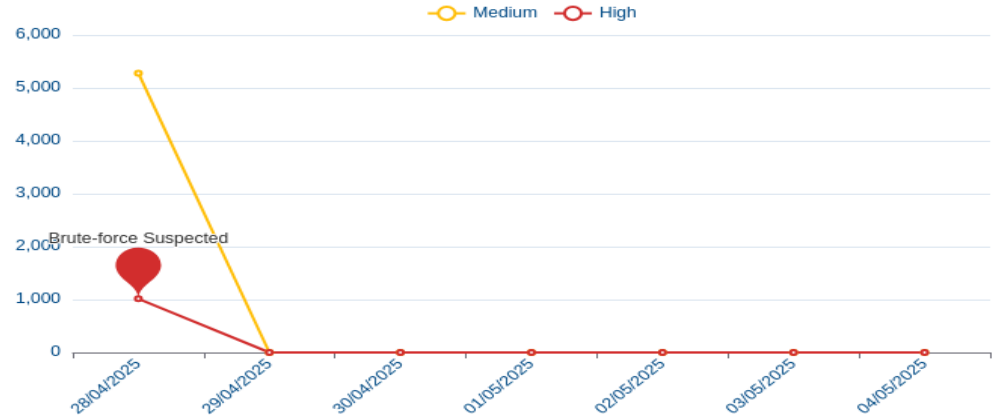
Note: You can test each of these APIs using tools like **Postman** or **curl**. When sending the request, ensure that the to field is filled, as it is required for the email to be sent successfully.

- **Weekly report**:

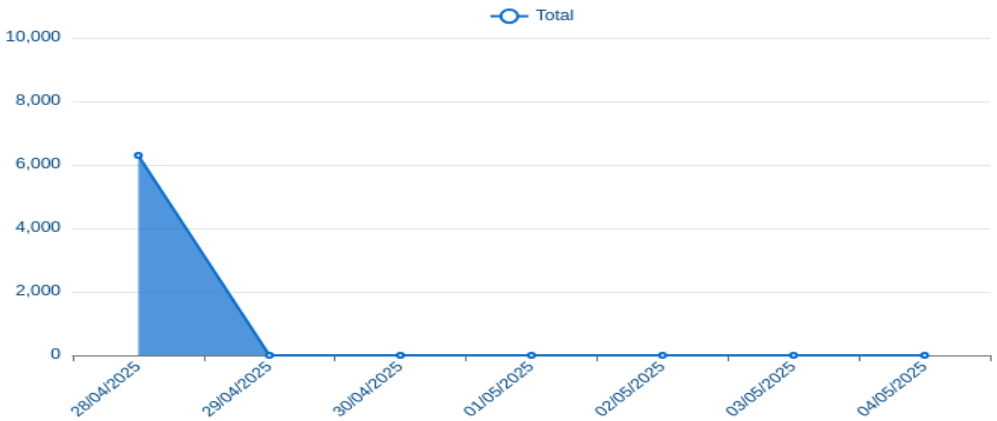
SAL Reports - Week 5, April 2025

Report generated on 28/04/2025 at 23:36:11

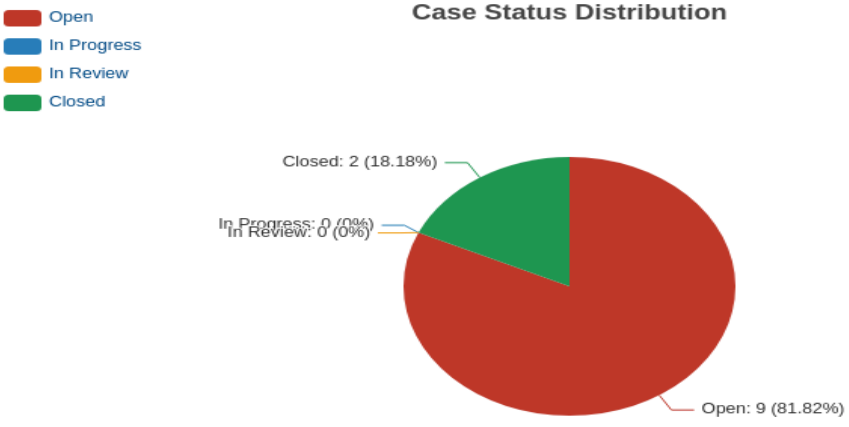
Daily Event Severity Counts



Daily Total Event Counts



Case Status Distribution



Incident Counts by Account

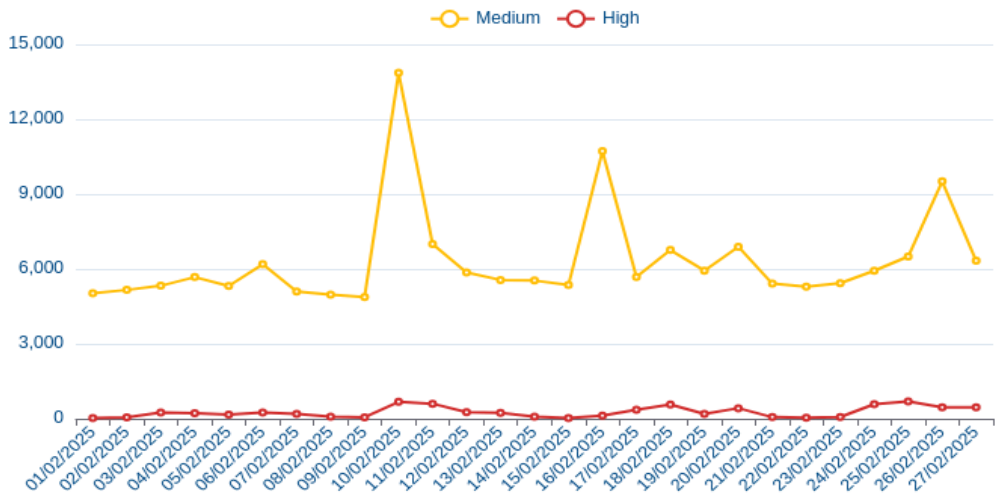
Account	Incident Count
LEARN-255	3
LEARN-253	3
LEARN-254	2
LEARN-256	1
LEARN-518	1
LEARN-257	1

- **Monthly report**

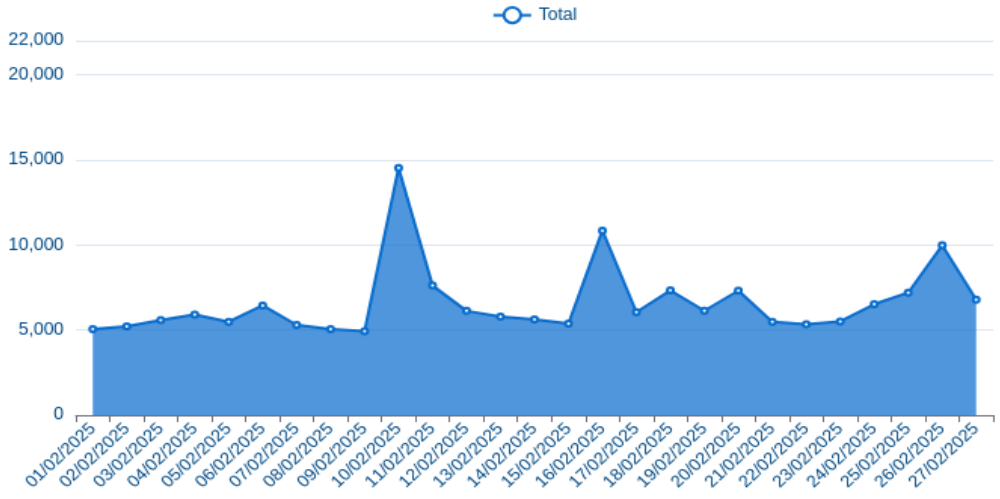
SAL Reports - February 2025

Report generated on 28/04/2025 at 23:29:42

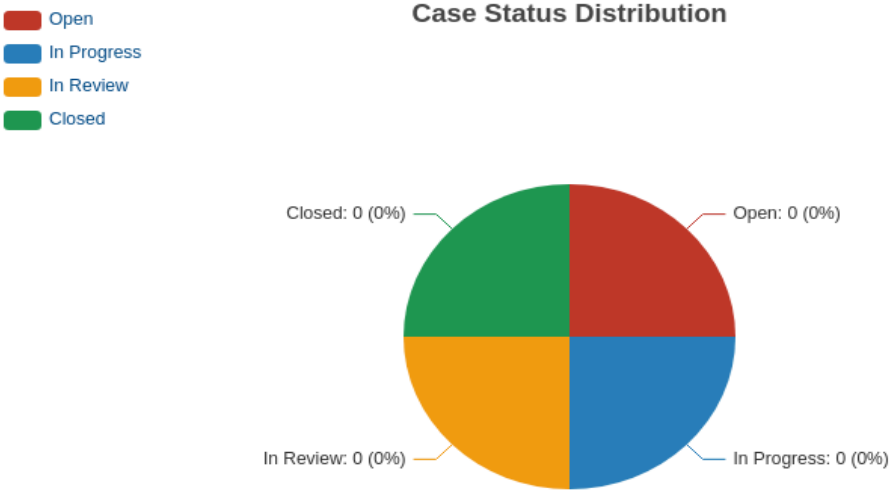
Daily Event Severity Counts



Daily Total Event Counts



Case Status Distribution



Incident Counts by Account

Account	Incident Count

Testing with Postman

To test the /notification/new-case API, for example, use the following **Postman** request:

1. **Method**: POST
2. **URL**: http://localhost:3000/notification/new-case
3. **Headers**:
    - Content-Type: application/json
4. **Body (raw)**:

```
{
 "to": "recipient@example.com",
 "userName": "Hieu",
 "caseId": "CASE002",
 "caseName": "New Case Test",
 "terminal": "Device123",
 "assignee": "John Doe",
 "level": "Medium",
 "timeExecuted": "2025-03-08T12:00:00"
}
```

Expected Response:

```
{
 "to": "recipient@example.com",
 "caseId": "CASE002",
 "templateType": "newCase",
 "status": "Email sent successfully"
}
```


VI. Implementation & Testing

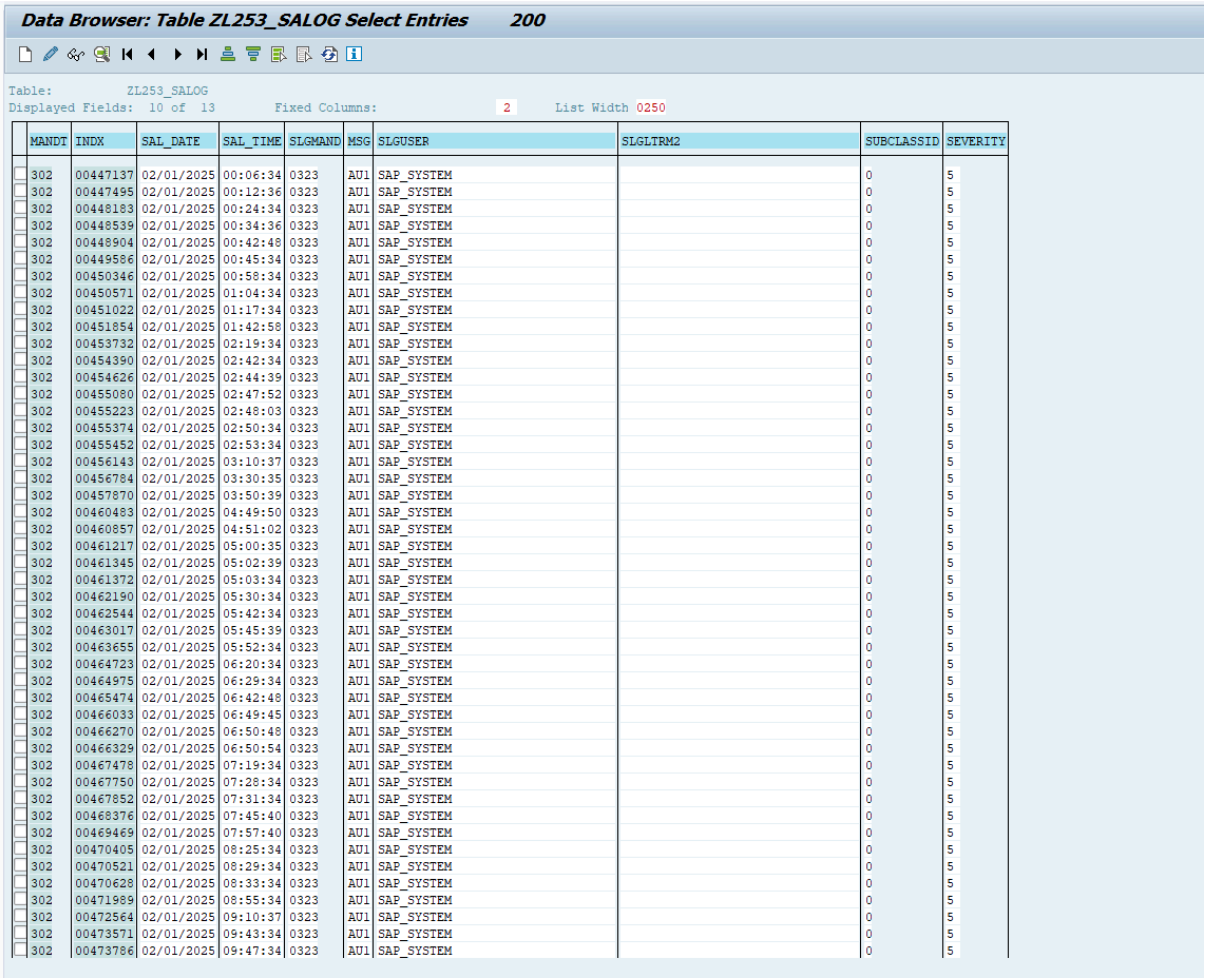
1. Implementation

The implementation phase of the project focused on configuring and developing components within the SAP system to meet internal security monitoring requirements. The key elements included log collection from SM20, custom ABAP logic for detection and notification, and a SAP Fiori-based user interface.

1.1. SAP Process Execution

Although this is not a business process implementation, technical configuration and log processing still follow a defined system flow:

- **Log Collection (SM20 Integration)**:
    - SM20 logs are queried and extracted using predefined filters (e.g., only Client 302).
    - A custom table ZL253_SALOG was created to store selected fields: SAL_DATE, SAL_TIME, MSG, SLGUSER, SEVERITY, etc..

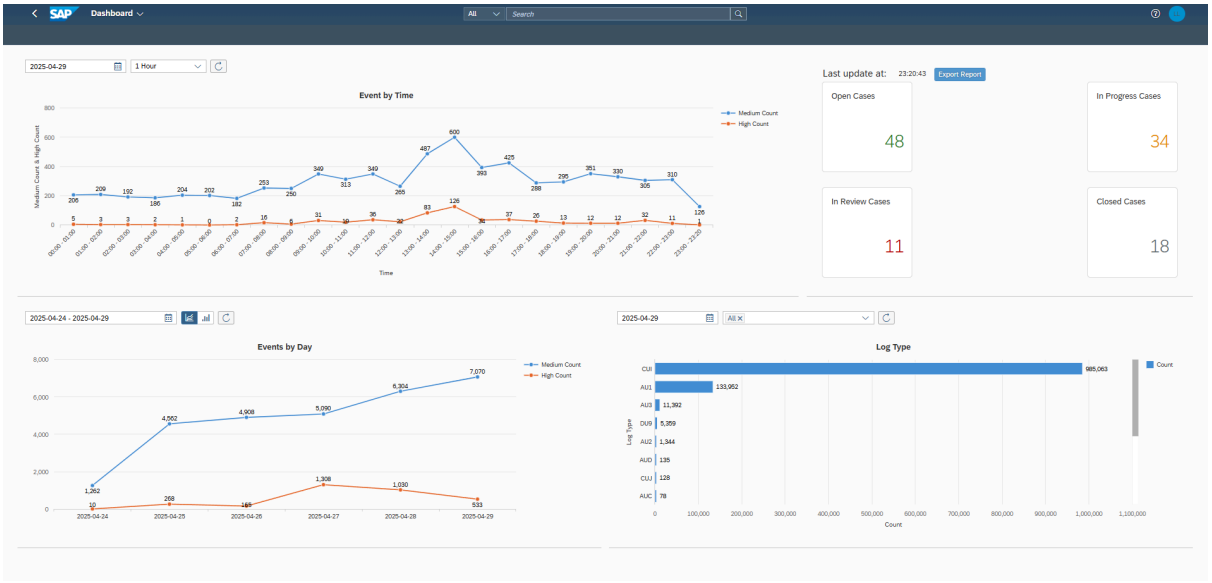


Data Browser: Table ZL253_SALOG Select Entries 200

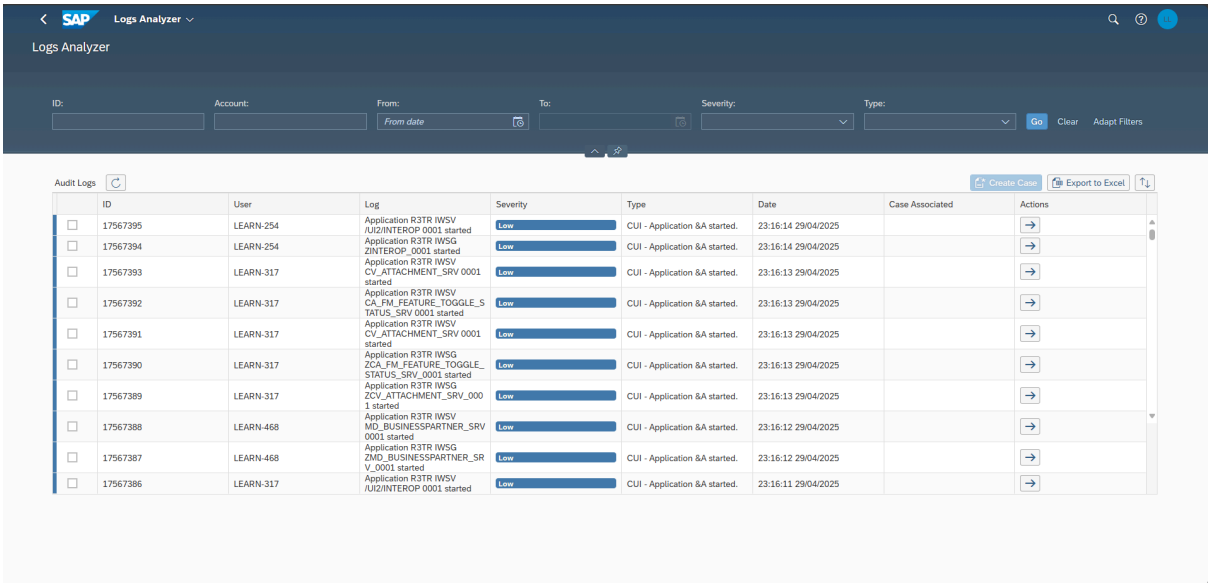
Table: ZL253_SALOG
Displayed Fields: 10 of 13 Fixed Columns: 2 List Width 0250

MANDT	INDX	SAL_DATE	SAL_TIME	SLGMAND	MSG	SLGUSER	SLGLTRM2	SUBCLASSID	SEVERITY
302	00447137	02/01/2025	00:06:34	0323	AU1	SAP_SYSTEM		0	5
302	00447495	02/01/2025	00:12:36	0323	AU1	SAP_SYSTEM		0	5
302	00448183	02/01/2025	00:24:34	0323	AU1	SAP_SYSTEM		0	5
302	00448539	02/01/2025	00:34:36	0323	AU1	SAP_SYSTEM		0	5
302	00448904	02/01/2025	00:42:48	0323	AU1	SAP_SYSTEM		0	5
302	00449586	02/01/2025	00:45:34	0323	AU1	SAP_SYSTEM		0	5
302	00450346	02/01/2025	00:58:34	0323	AU1	SAP_SYSTEM		0	5
302	00450571	02/01/2025	01:04:34	0323	AU1	SAP_SYSTEM		0	5
302	00451022	02/01/2025	01:17:34	0323	AU1	SAP_SYSTEM		0	5
302	00451854	02/01/2025	01:42:58	0323	AU1	SAP_SYSTEM		0	5
302	00453732	02/01/2025	02:19:34	0323	AU1	SAP_SYSTEM		0	5
302	00454390	02/01/2025	02:42:34	0323	AU1	SAP_SYSTEM		0	5
302	00454626	02/01/2025	02:44:39	0323	AU1	SAP_SYSTEM		0	5
302	00455080	02/01/2025	02:47:52	0323	AU1	SAP_SYSTEM		0	5
302	00455223	02/01/2025	02:48:03	0323	AU1	SAP_SYSTEM		0	5
302	00455374	02/01/2025	02:50:34	0323	AU1	SAP_SYSTEM		0	5
302	00455452	02/01/2025	02:53:34	0323	AU1	SAP_SYSTEM		0	5
302	00456143	02/01/2025	03:10:37	0323	AU1	SAP_SYSTEM		0	5
302	00456784	02/01/2025	03:30:35	0323	AU1	SAP_SYSTEM		0	5
302	00457870	02/01/2025	03:50:39	0323	AU1	SAP_SYSTEM		0	5
302	00460483	02/01/2025	04:49:50	0323	AU1	SAP_SYSTEM		0	5
302	00460857	02/01/2025	04:51:02	0323	AU1	SAP_SYSTEM		0	5
302	00461217	02/01/2025	05:00:35	0323	AU1	SAP_SYSTEM		0	5
302	00461345	02/01/2025	05:02:39	0323	AU1	SAP_SYSTEM		0	5
302	00461372	02/01/2025	05:03:34	0323	AU1	SAP_SYSTEM		0	5
302	00462190	02/01/2025	05:30:34	0323	AU1	SAP_SYSTEM		0	5
302	00462544	02/01/2025	05:42:34	0323	AU1	SAP_SYSTEM		0	5
302	00463017	02/01/2025	05:45:39	0323	AU1	SAP_SYSTEM		0	5
302	00463655	02/01/2025	05:52:34	0323	AU1	SAP_SYSTEM		0	5
302	00464723	02/01/2025	06:20:34	0323	AU1	SAP_SYSTEM		0	5
302	00464975	02/01/2025	06:29:34	0323	AU1	SAP_SYSTEM		0	5
302	00465474	02/01/2025	06:42:48	0323	AU1	SAP_SYSTEM		0	5
302	00466033	02/01/2025	06:49:45	0323	AU1	SAP_SYSTEM		0	5
302	00466270	02/01/2025	06:50:48	0323	AU1	SAP_SYSTEM		0	5
302	00466329	02/01/2025	06:50:54	0323	AU1	SAP_SYSTEM		0	5
302	00467478	02/01/2025	07:19:34	0323	AU1	SAP_SYSTEM		0	5
302	00467750	02/01/2025	07:28:34	0323	AU1	SAP_SYSTEM		0	5
302	00467852	02/01/2025	07:31:34	0323	AU1	SAP_SYSTEM		0	5
302	00468376	02/01/2025	07:45:40	0323	AU1	SAP_SYSTEM		0	5
302	00469469	02/01/2025	07:57:40	0323	AU1	SAP_SYSTEM		0	5
302	00470405	02/01/2025	08:25:34	0323	AU1	SAP_SYSTEM		0	5
302	00470521	02/01/2025	08:29:34	0323	AU1	SAP_SYSTEM		0	5
302	00470628	02/01/2025	08:33:34	0323	AU1	SAP_SYSTEM		0	5
302	00471989	02/01/2025	08:55:34	0323	AU1	SAP_SYSTEM		0	5
302	00472564	02/01/2025	09:10:37	0323	AU1	SAP_SYSTEM		0	5
302	00473571	02/01/2025	09:43:34	0323	AU1	SAP_SYSTEM		0	5
302	00473786	02/01/2025	09:47:34	0323	AU1	SAP_SYSTEM		0	5

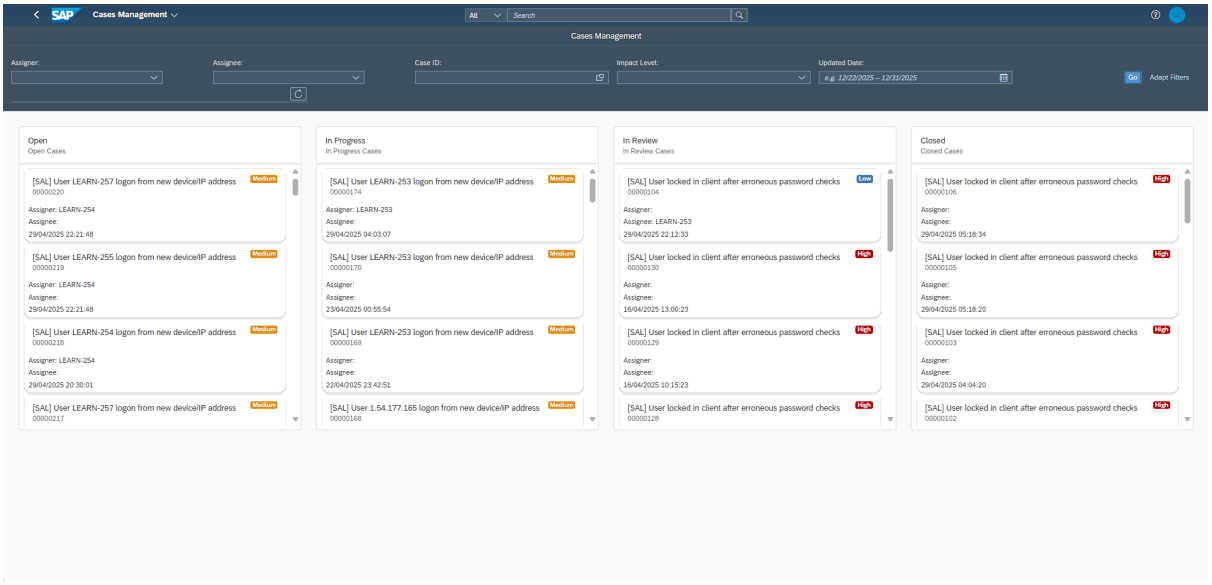
- **Custom ABAP Functions**:
    - Dashboard: Provide executive-level summaries and high-level visual reports of security-related activities based on logs, visualizations and case data



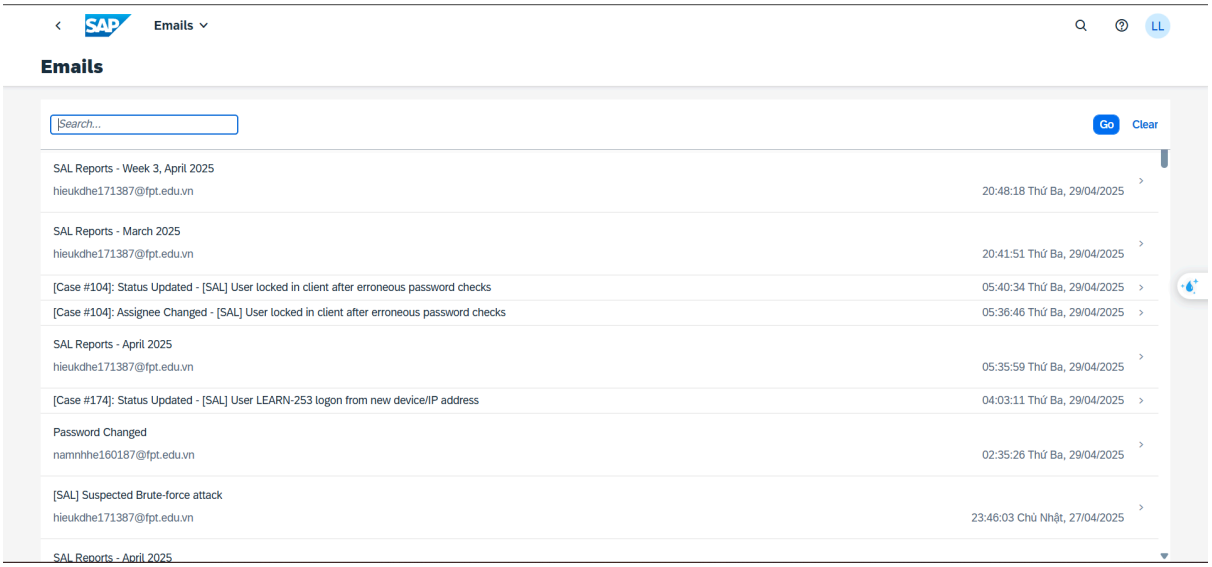
- Logs Analyze: monitor abnormal logs and identify cases that require further investigation and case creation.



- Cases Management: Retrieve cases for **Security Officers** to review, manage, and **take appropriate actions**



- Emails: Views sent emails from notifications in this screen

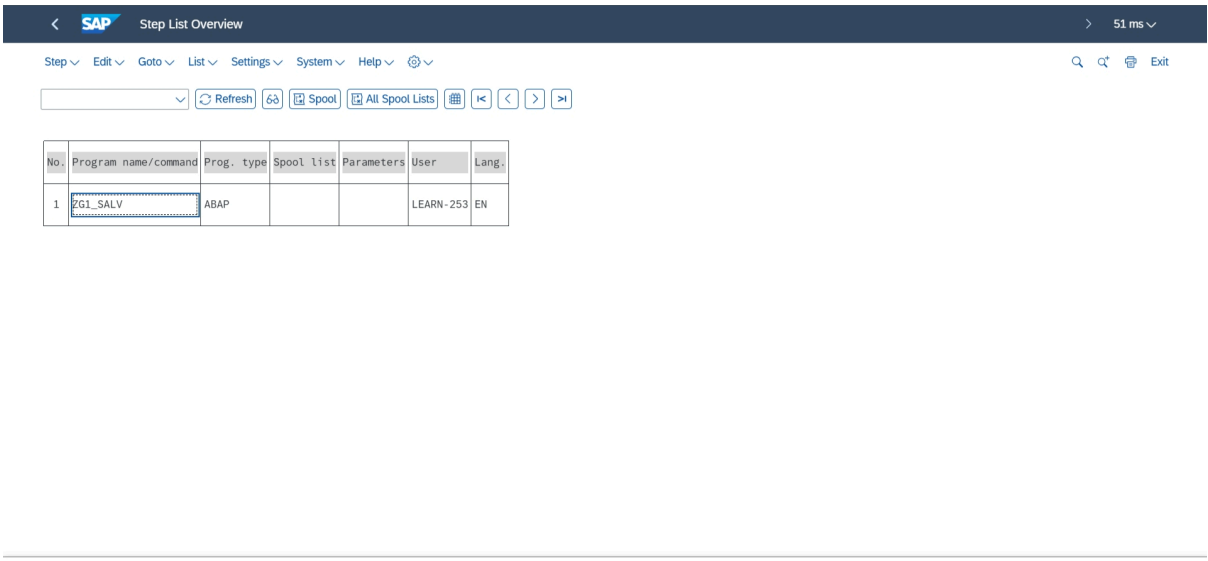


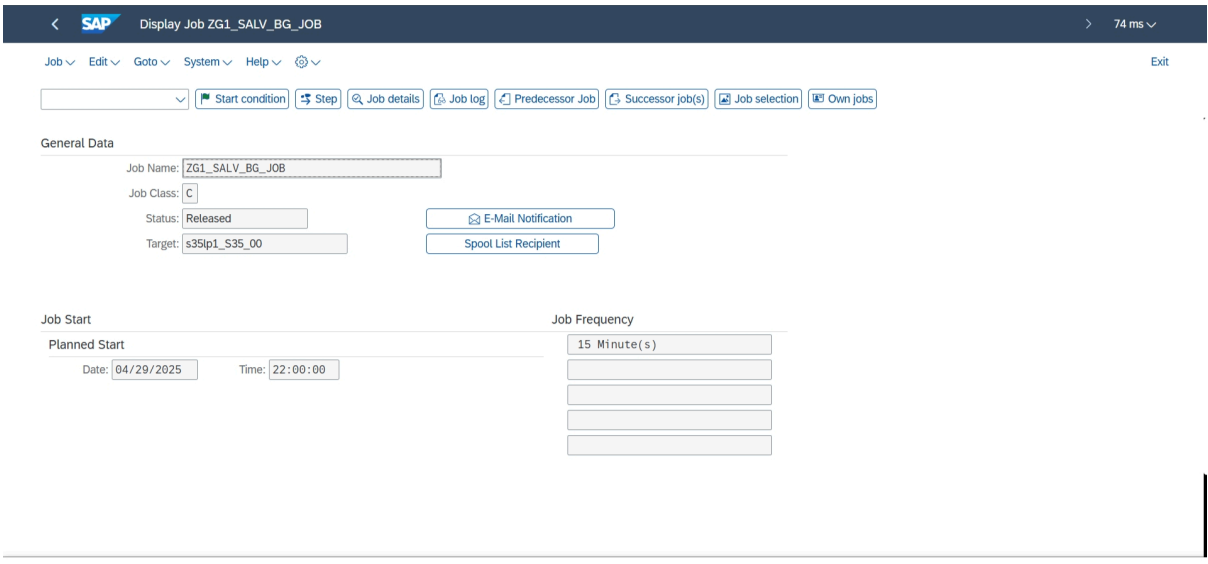
- **Notification System**:
    - Notifications are automatically triggered when a suspicious log is identified.
    - Alerts can be sent via **email** or **Microsoft Teams API**, depending on the configuration.

1.2. Installation Guides

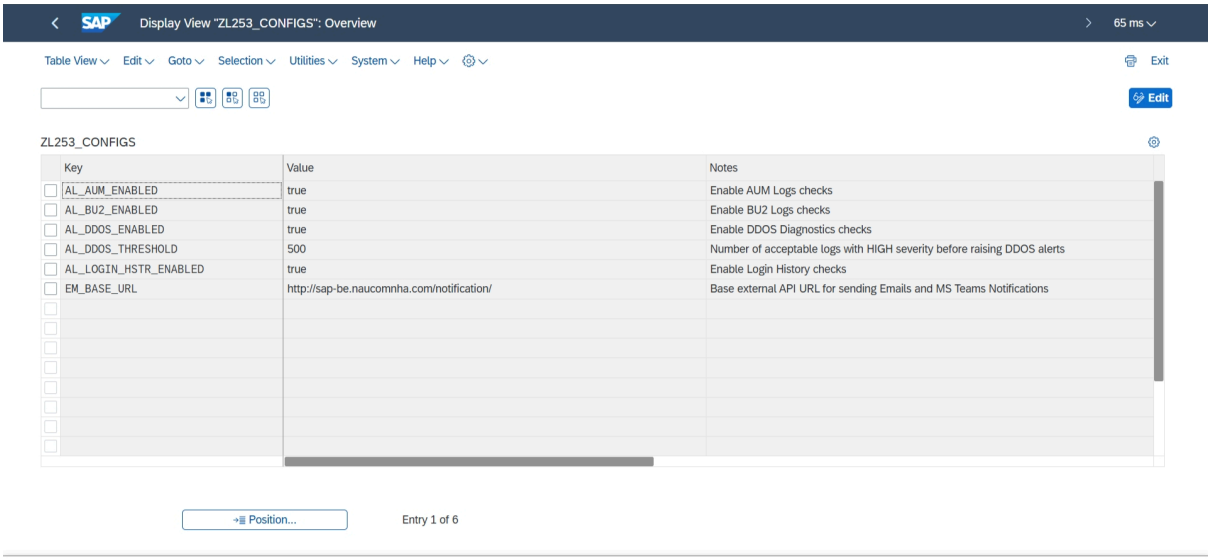
A set of comprehensive **installation guides** was created to assist the deployment team in setting up the solution. These guides covered the following areas:

- **App Deployment**:
    - Transport Source Codes, Database Tables, Classes and Fiori Applications to Target Server
    - Create background job: Run program **ZG1_SALV** repeatedly**,** separated by an amount of time (15 minutes for example)

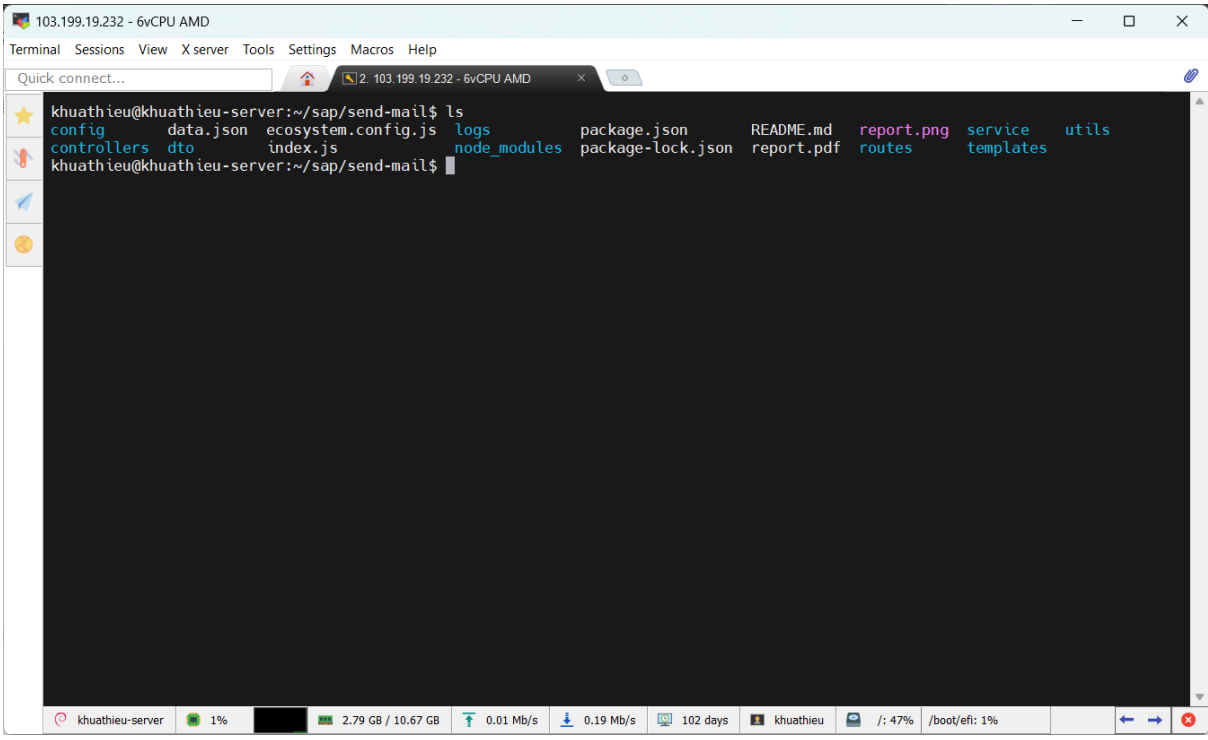


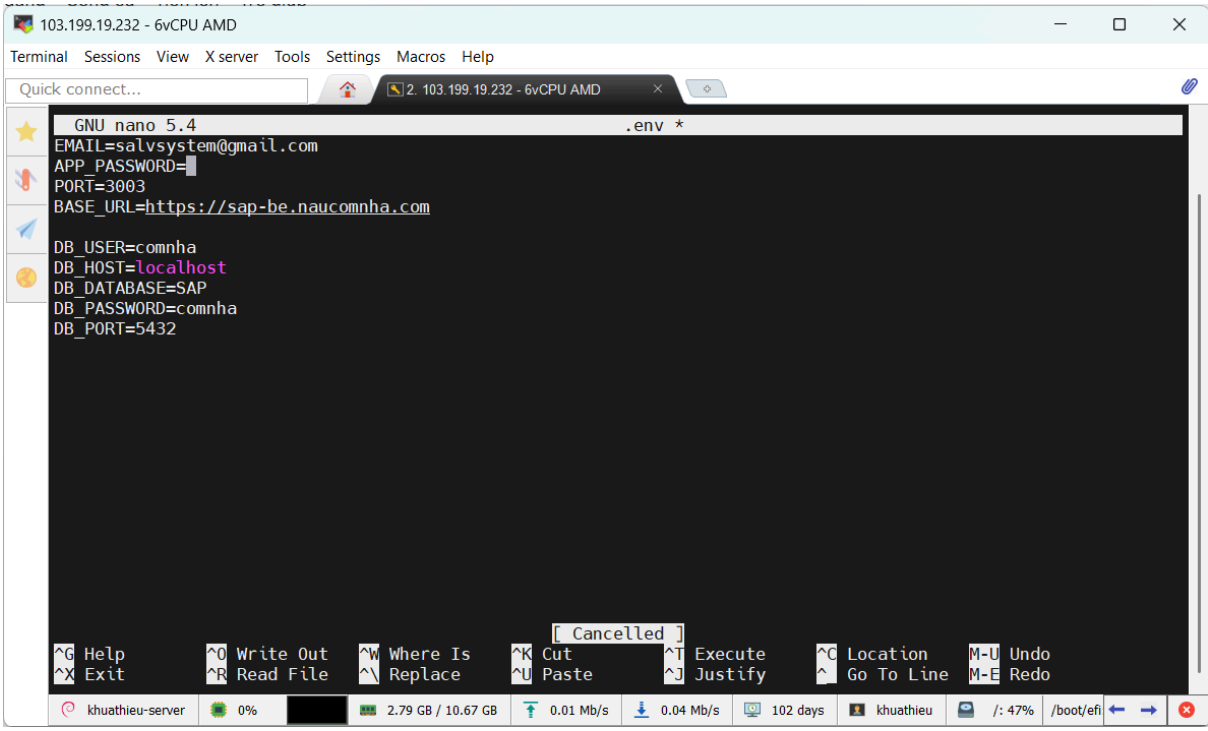


    - Run TCode **ZG1CFG** and setup configuration variables

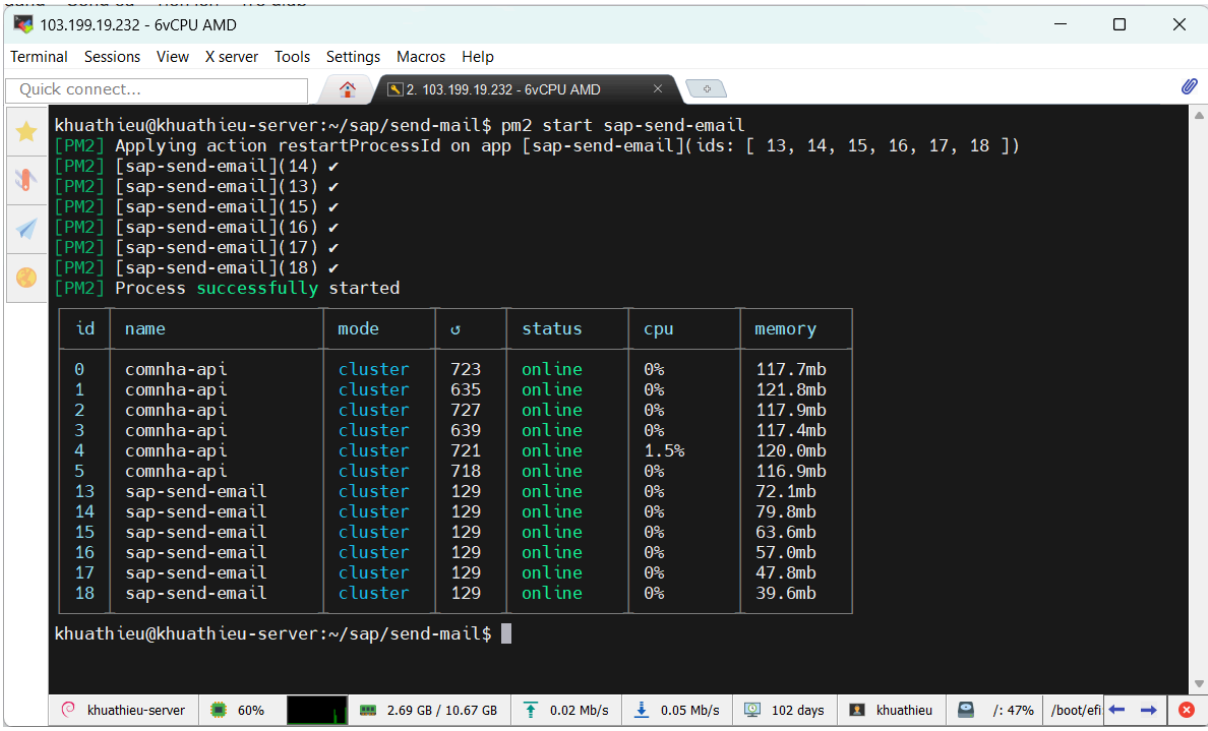


- Clone NodeJS Source Code to Server
- Run npm install, setup .env

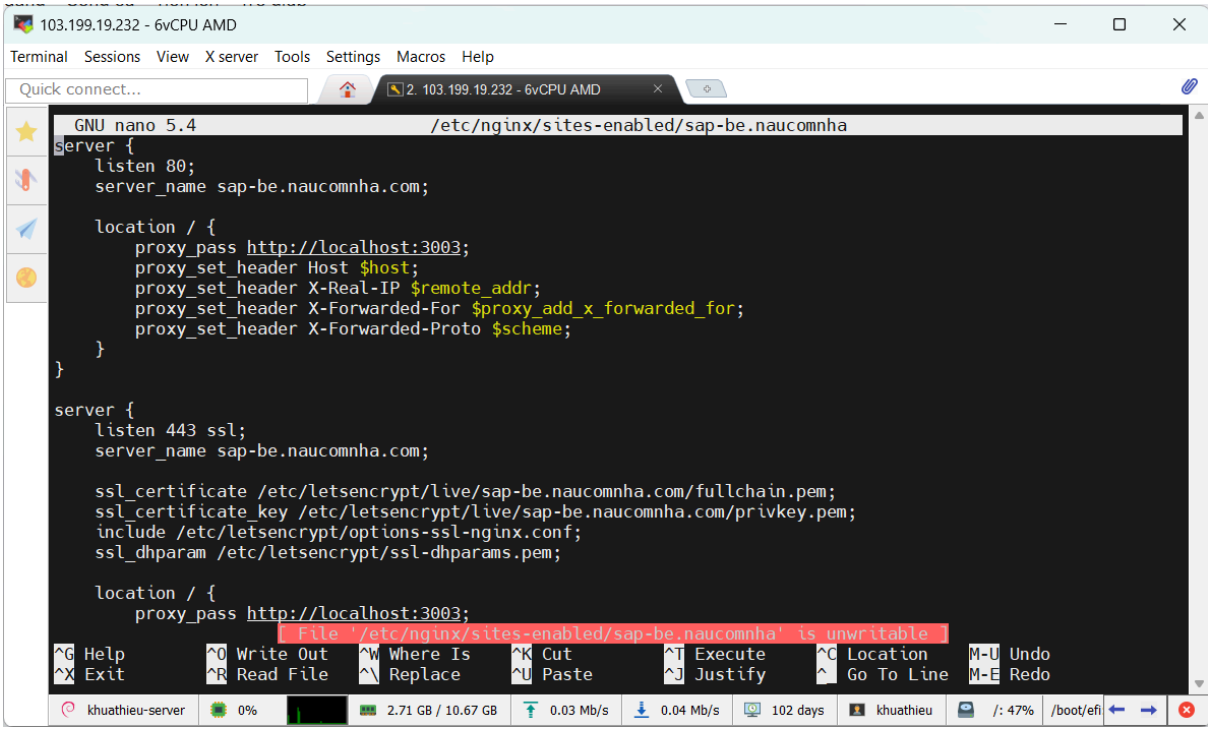




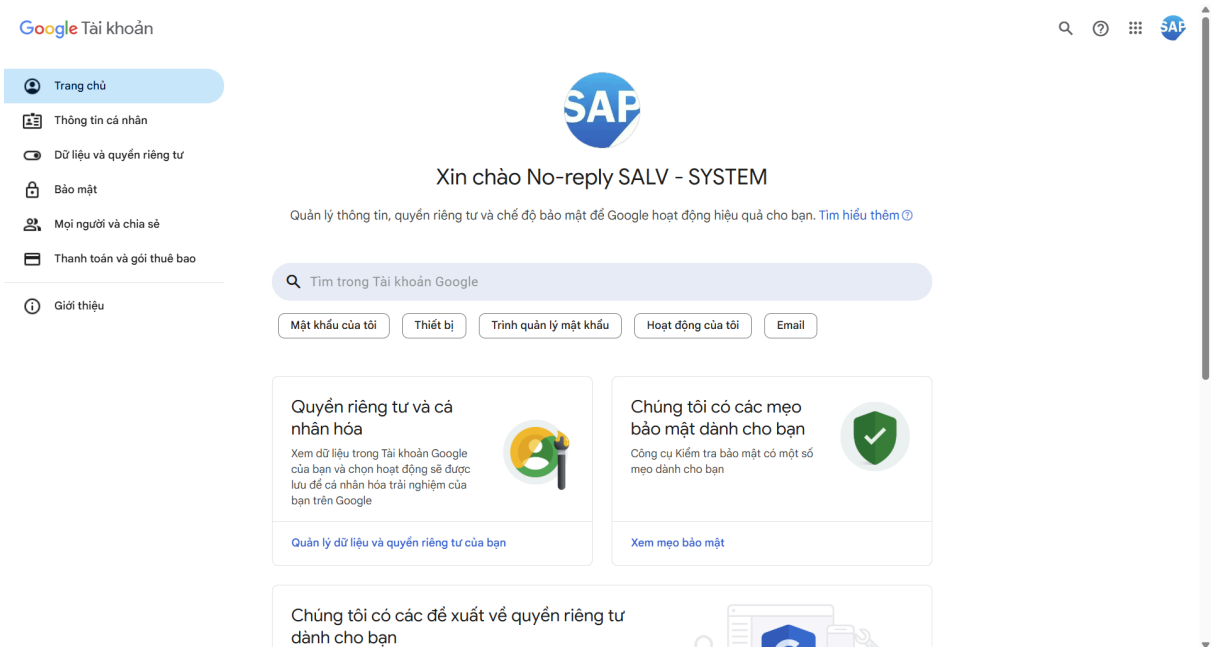
- Run using pm2

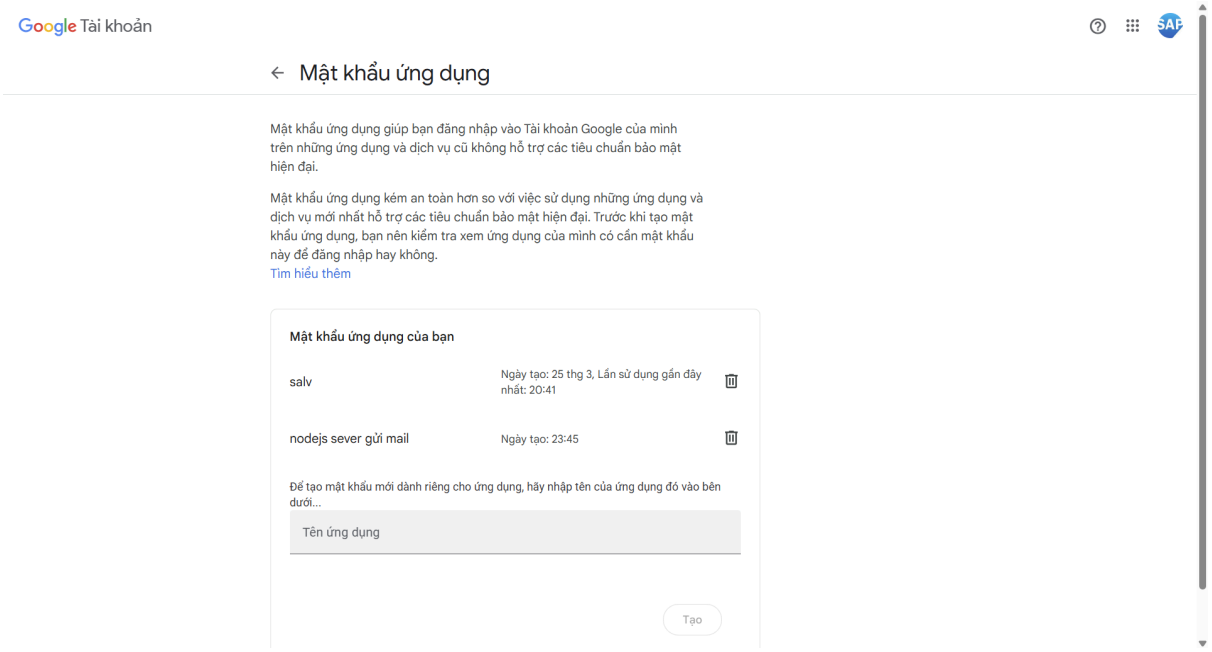


- Setup Nginx as a reverse proxy

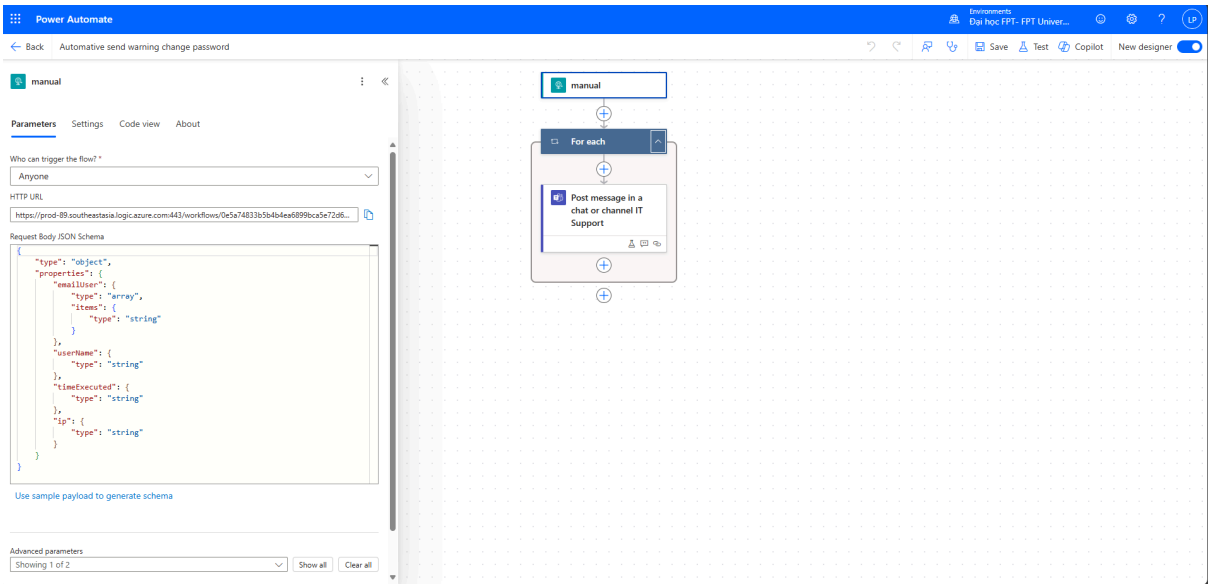


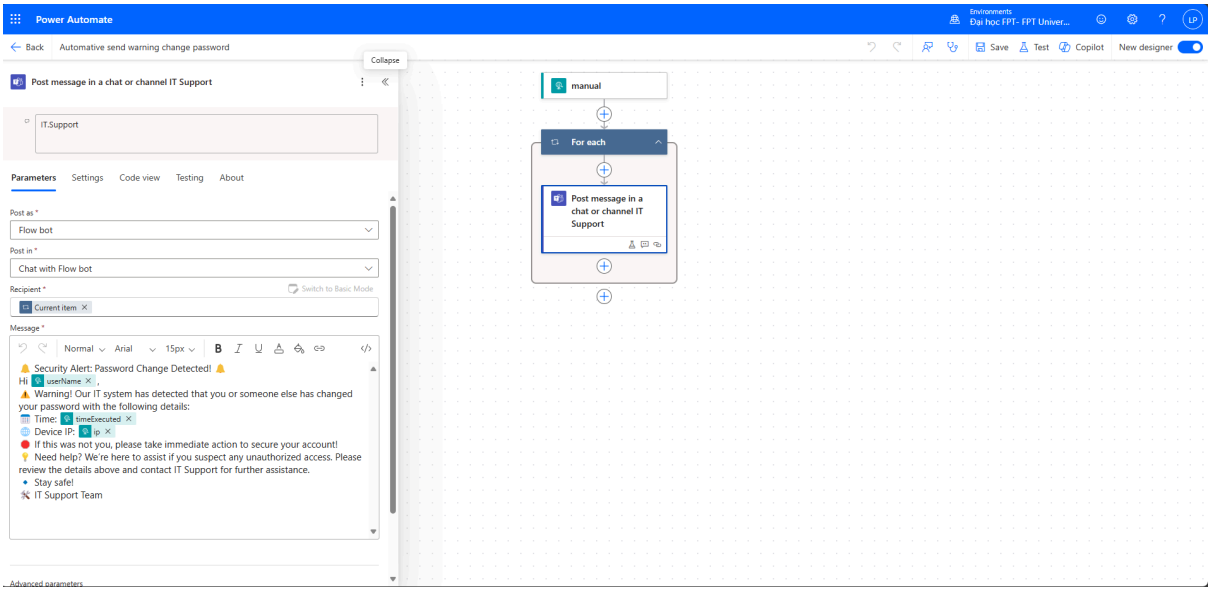
- **Notification Config**:
    - Step-by-step guidance was given for configuring the email notification system, including setting up the SMTP server and defining notification templates.
        - Emails (GMail):





- Message (MS Teams):





1.3. User Manual

A **detailed user manual** was created to provide end-users with instructions on how to interact with the solution. The manual covered the following aspects:

- **Log Monitoring**:
    - The user manual described how users could monitor security logs in real time through the **ZSALV_LOGS_TABLE** function. This section covered filtering logs, viewing details, and using the **Fetch** button to retrieve the latest log entries.
- **Responding to Alerts**:
    - Users were guided on how to respond to automated alerts. The manual explained how email notifications were triggered and how users could take necessary actions based on the severity of the alerts received.
- **Case Creation**:
    - A detailed step-by-step guide was provided on how users could create cases from suspicious logs. This included instructions on filling in the case title, description, impact level, and attaching relevant logs as evidence.
- **Case Management**:
    - The manual explained the process for managing created cases, including how to update case statuses, add comments, and assign cases to different users for resolution. It also covered how to view case details and track progress through the SAP Fiori

interface.

- **Dashboard Usage**:
    - Users were instructed on how to navigate and use the **System Dashboard** to monitor overall system health and security activity trends. The dashboard provided an overview of log events, case statuses, and key performance indicators (KPIs).
- **Exporting Logs**:
    - The manual also described how users could export log data and case information to Excel for offline analysis and reporting. This feature was especially useful for audit purposes.

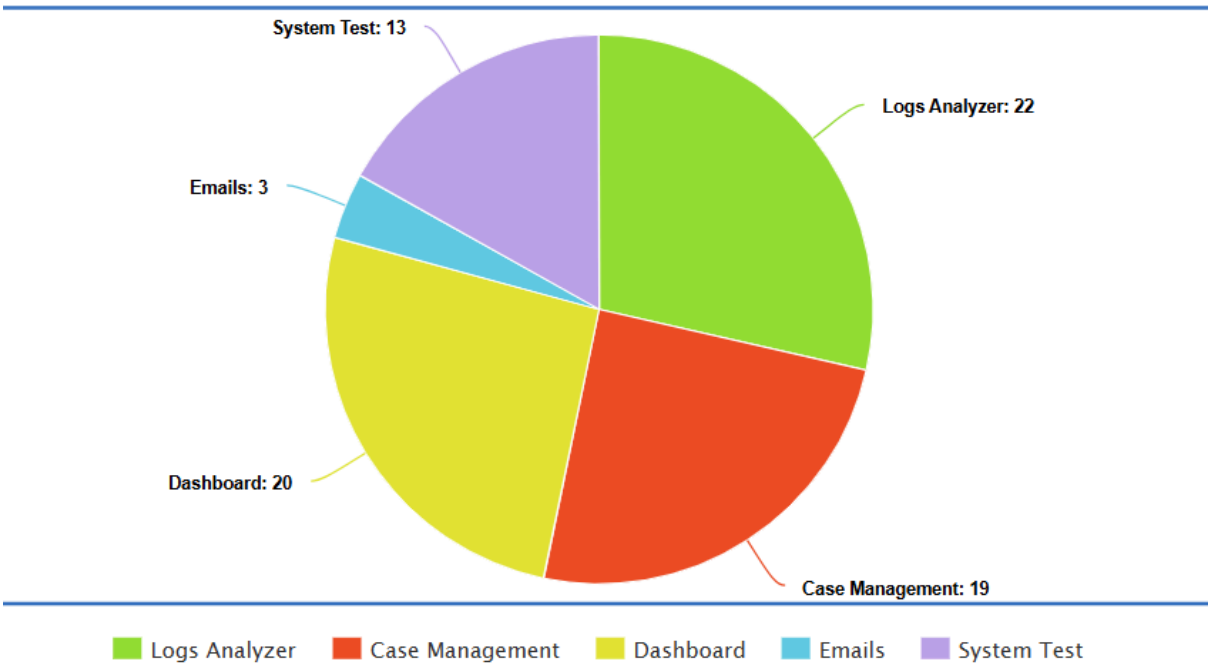
---

2. Testing

2.1. Unit Test

Test Results Report

Metric	Count
Total Test Cases	77
Total Test Cases Passed	77
Total Bugs Recorded	8
Total Bugs Fixed	8
Total Test Cases Failed	0



2.2. System Test

System Test Summary

Test Case ID	Test Case Description	Status
TC01	Login to sap view 4 apps	Pass
TC02	Go to App Dashboard, View Dashboard Static Logs and Cases	Pass
TC03	App Logs Analyzer, View Table Log	Pass
TC04	Find and Select Logs	Pass
TC05	Create Case	Pass
TC06	Change Information of Newly Created Case	Pass
TC07	Update the Case Being Processed and Comment on the Case	Pass
TC08	Update the Case Being Reviewed and Comment on the Case	Pass
TC09	Update Closed Cases and Send Notification Emails	Pass
TC10	Test Creating a Case with Invalid Input Data	Pass

TC11	Test Creating a Case without Selecting the Mandatory Fields	Pass
TC12	Verify System Behavior Under Heavy Load	Pass
TC13	Verify Data Integrity When a Case is Created from Multiple Devices	Pass

VII. Appendix

1. Tobe Process
SAP490_G1_Tobe_Process
2. Configuration
SAP490_G1_Configuration
3. User Manual
SAP490_G1_User_Manual
4. Functional/ Technical Specification
SAP490_G1_Functional Specification
5. Test Document
SAP490_G1_Test_Case