

**Commercial Bank of Ethiopia**

**Suspicious Transaction Reporting System Design Document**

**Prepared By.**

**System Development and Customization**

**May, 2024 G.C**

Table of Contents

[Table Of Figure II](#_Toc166089906)

[List of Table III](#_Toc166089907)

[1. Introduction 1](#_Toc166089908)

[Simply STR Overview 1](#_Toc166089909)

[1.1 Purpose of the System Design Document (SDD) 2](#_Toc166089910)

[1.2 Audience 3](#_Toc166089911)

[1.3 Executive Summary 3](#_Toc166089912)

[1.3.1 System Overview Summary 3](#_Toc166089913)

[1.3.2 User Types 4](#_Toc166089914)

[2. General Overview and Design Approach 5](#_Toc166089915)

[2.1 General Overview 5](#_Toc166089916)

[2.1.1 Proposed Solution 6](#_Toc166089917)

[2.2 Stakeholder Roles Concerns 6](#_Toc166089918)

[2.2.1 Concerns 7](#_Toc166089919)

[2.3 System Assumptions/Constraints/Dependencies/Risks 7](#_Toc166089920)

[2.3.1 Assumptions 7](#_Toc166089921)

[2.3.2 Constraints 7](#_Toc166089922)

[2.3.3 Dependencies 7](#_Toc166089923)

[2.3.4 Risks 7](#_Toc166089924)

[2.4 Operational Environment 8](#_Toc166089925)

[2.5 Development Methods & Contingencies 8](#_Toc166089926)

[3. System Architecture and Architecture Design 9](#_Toc166089927)

[3.1 System Architecture Diagrams 9](#_Toc166089928)

[3.1.1 STR Systems diagram 9](#_Toc166089929)

[3.2 Hardware Architecture 10](#_Toc166089930)

[3.3 Software Architecture 13](#_Toc166089931)

[3.4 Security Software Architecture 14](#_Toc166089932)

[4. System Design 15](#_Toc166089933)

[4.1 Business Requirements 15](#_Toc166089934)

# Table Of Figure

[Figure 1 STR System Diagram 12](#_Toc166089765)

[Figure 2 Hardware Architecture 13](#_Toc166089766)

[Figure 3 Software architecture tiers 16](#_Toc166089767)

# List of Table

[Table 1 Project members contact information 10](#_Toc166089824)

[Table 2 Systems descriptions 15](#_Toc166089825)

# 1. Introduction

Anti-Money Laundering (AML) and Countering the Financing of Terrorism (CFT) compliance are crucial components of the global effort to combat financial crimes. Banks play a pivotal role in ensuring the integrity of the financial system by adhering to AML/CFT regulations. These regulations are designed to detect and prevent illicit activities such as money laundering and terrorist financing, which could undermine the stability and security of the financial sector. Among the identified particulars that the corporate compliance requested System Development and Customization Department of the bank was preparing an automated system for reporting of Suspicious Transactions. When suspicious transactions are identified, banks are required to file Suspicious Activity or Transactions Reporting with the relevant regulatory authorities. These reports provide detailed information about the transaction and the reasons for suspicion, enabling authorities to investigate further.

Goals are to use service coordination and technology integration to:

* Increase a reporting facilitations and secure data transmission for customer privacy.
* Achieve more efficient use of information for finance intelligence.

## Simply STR Overview

Simply STR is web-based system that used to report any suspected transaction by Branch Officer from any CBE Branch to Head Office Risk and Compliance team for further identification. And if the information and suspicion is genuine and useful for the Finance intelligence it will send for other stockholders by exporting the report as Pdf or excel from the system. Else the information will return for the sender Branch officer weather to add some addition justification or other suspicion supportive documents. Occasionally some of the reports will be rejected after the head office team discuses on it and if it is found genuine.

STR Summary:

* STR is one login and Send Report
  + Following up the report status based
* Responsive design for use on computers, tablets, and smartphones
* Includes multiple file uploads to clarify the suspicion
* Incudes pdf and excel export based on FIC reporting format
* Software application developed with System Development and Customization Department

## Purpose of the System Design Document (SDD)

The SDD documents and tracks the necessary information required to effectively define architecture and system design in order to give the development team guidance on the architecture of the system to be developed. Design documents are incrementally and iteratively produced during the system development life cycle, based on the particular circumstances of the Software projects and the system development methodology used for developing the system. SDC Team is the developer of the current solution.

The intent of the solution was to provide a “Paper clean environment” solution. This was not achieved in the initial implementation. The vision of the suspicious transaction reporting system was to automate the paper-based reporting systems in to one centralized database with a multi-functional web interface. This concept would maximize staff resources and make information accessible to a wider range of the Bank.

The proposed solution will simply add additional features and functionality to the existing working environments to meet the original scope and vision of the suspicious transaction reporting Methods. These features will extend STR to be a real and integrated

The following are major functionality that corporate compliance team requested to be added or improved include:

* Entity making the Report (Bank Information)
* Personal Information
* Transactions Information
* Account Information
* Association Information
* Particulars of Suspicious or Unusual Activity
* List of Available Document
* Create Report Field
* Profile of staff who report the transactions
* Approved/Reject Menu
* Reject Comment Box
* Reject Notification Popup messages via outlook
* Assign/Reassign Menu
* STR Dashboard menu
* Admin User

## 1.2 Audience

The intended audience for the Softer Design Document is the Software Development and Customization manager, project Development team, the Risk and Compliance team and Security Clearance Team

## 1.3 Executive Summary

### 1.3.1 System Overview Summary

The purpose of this discourse is to develop some kind of program and automate for the end user to fill the information and derive out some meaningful document either in pdf or word. One of the requirements of the Financial Intelligence Services is for the end operators to identify Suspicious Transactions. Presently, Commercial Bank of Ethiopia is using a paper-based information content to be filled manually and be sent by email after scanning. CBE has systems that are able to identify suspicious transactions that are required to be reported to the Financial Intelligence Centre. Branches have obligation to fill suspicious transactions. But the filled suspicious transactions are presently done manually and the current manual process can be prone to secrecy losses. Instead, if there is some sort of application created so that the staff filling the suspicious transaction just fills the information and at the end a PDF or word document props up and can be sent to compliance or FIC according to the need.

Automating suspicious transactions reporting is crucial in enhancing the efficiency, accuracy, and effectiveness of Anti Money Laundering (AML) and Countering the Financing of Terrorism (CFT) compliance processes within Commercial Bank of Ethiopia. It enables the bank to respond swiftly to emerging threats, enhance accuracy in detecting illicit activities, and streamline their compliance processes in the face of increasing data volumes and evolving regulatory landscapes. Thus, Corporate Compliance need a system that eases the reporting process and reporting of the suspicious transactions to FIC in an automated way.

### 1.3.2 User Types

STR is designed to serve the needs of many different types of users, with features and functions appropriate for each one:

* For any CBE Employee officer to send the suspected transaction report
* Risk and Compliance officer to receive the report and check the completeness of the report for weather to send the report for finance intelligence team or to return the report for the sender to corrections.
* Risk and Compliance Manager to assign and reassign the officer for that specific report validation tasks
* Accesses Management team to manage the user and their access

# 2. General Overview and Design Approach

## 2.1 General Overview

The current solution was built by CBE System Development and Customization Team. STR is a web-based system that connects CBE employee who need report a suspicious transaction in secure manner for further investigations to keep our country finance health and fight against money laundries.

STR is built on an Asp.net framework. Source code is available using standard open-source management tools such as Git. All source code is stored in a GitLab repository. Any CBE SDC Team can contribute to the application and CBE security Department can make vulnerability assessment on it.

***STR utilizes a CBE windows web server.*** Windows Server is a robust operating system developed by Microsoft specifically designed for server-side applications and services, it provides web server functionality with a focus on high performance, concurrency, and efficient resource usage. It supports features like load balancing, caching, access control, and integration with various applications, making it well-suited for modern website architectures. Widely used in enterprise environments and data centers, Windows Server offers reliability, extensive management tools, and security features, making it a popular choice for hosting websites and applications that require scalability and comprehensive administrative controls.

***STR is hosted on CBE data center on server address 10.3.11.75(https://str.cbe.com.et/)***

***STR development language is C#.NET MVC,*** which is commonly used for building web applications. C# is a widely adopted programming language, and in the context of the system, the .NET MVC framework is utilized to extend its capabilities. This framework, known as ASP.NET MVC, is essential for software developers to maintain and enhance the functionality of the application.

***STR database is SQL Server.*** SQL Server is a highly regarded and preferred database management system for developing applications within the CBE. It offers a range of robust features and provides excellent support for meeting the specific needs of enterprise applications.

***GitLab is used as the software development platform.*** GitLab is utilized as the software development platform, offering version control and source code management capabilities. GitLab serves as a repository for storing and managing source code. GitLab is known as one of the leading hosts of source code globally, providing a reliable and efficient platform for developers.

In summary, the system design includes the following sub-systems:

* STR Web Application
* SQL Server Database
* Windows Web Server
* GitLab Version and Source Code Control

There is no expectation that any of these systems will be changed or modified with the proposed system.

### 2.1.1 Proposed Solution

The proposed solution is suspicious transaction reporting system. This system enables any CBE employee to report suspicious transactions by filling out the necessary form within the system. The compliance team at the head office receives, reviews, and verifies these reports. Once verified, the reports are exported in PDF or Excel FIC format and sent to the Financial Intelligence Center (FIC) for further analysis and investigation. This streamlined process helps ensure timely reporting of suspicious activities and facilitates effective collaboration between the bank and regulatory authorities.

## 2.2 Stakeholder Roles Concerns

#### Project Stakeholders

The following table provides the role and contact information for the key technical and project stakeholders associated with the system design.

Table 1 Project members contact information

|  |  |  |
| --- | --- | --- |
| Name | Role | Email |
| Genet Chanyalew | Project Manager | *GenetChanyalew@cbe.com.et* |
| Zenebe Nigusue | Application Owner | *ZenebeNigusie@cbe.com.et* |
| Getnet Adane & yohannes Sintayhu | Developer | *Getnetadane1@cbe.com.et* & *yohannessintayhu@cbe.com.et* |

### 2.2.1 Concerns

Due to the fact that the proposed system is simply new standalone. features and will not require any design or architectural changes, there are no technical concerns.

## 2.3 System Assumptions/Constraints/Dependencies/Risks

### 2.3.1 Assumptions

The largest assumption is that all CBE Employs have outlook username and password can access the system based on their outlook credential.

### 2.3.2 Constraints

There are no hardware, software, or software technical constraints identified with this project.

### 2.3.3 Dependencies

The current application is dependent on Ad (Active Directory) for authentication purpose.

### 2.3.4 Risks

There is no identified potential critical risk associated with this reporting system. That is why the system will not integrate with other CBE System. It is standalone system only used to automate the manual paper-based reporting system. Additionally, the system has schedule backup system for any data lose.

## 2.4 Operational Environment

* Asp.net MVC Development
* Git Version Control
* GitLab Repository
* SQL Server Database
* Windows Web Server

### 2.5 Development Methods & Contingencies

The basics of a good architecture is to layer the application into multiple autocratic and autonomous applications that can be replaced individually and allow us to keep the application running while we are working on a specific layer.

Scalability: Ensure that the architecture can be scaled horizontally, across multiple servers and across multiple regions. That means that once your traffic goes up, you should be able to add and remove new servers as the solution requires.

Availability: The architecture should support a high availability environment. Infrastructure redundancy is required. This ensures the solution is available if multiple servers or an entire data center fail.

Security: Solution architecture should expose only the minimal amount of code possible. Most of the back-end pieces should be hidden away. In addition to that, security of each system should be multi-layered.

Extensibility: Architecture must be able to swap out modules, change layers, and add pieces to the application without having to worry about the underlying data contracts in place.

Separation of responsibility: System should be modular enough that each piece of code has a set of responsibilities and not more. The back-end should not create front end code nor should the front-end code include business logic.

# 3. System Architecture and Architecture Design

This section outlines the system and hardware architecture design of the system.

### 3.1 System Architecture Diagrams

This section provides the conceptual view of the system and its functionality.

Simply Get There currently provides the following major components.

* Create, preview and submit ST Report
* View, assign and reassign ST report to compliance officer
* view, reject, accept and Generate Report
* easily follow up ST Reports

### 3.1.1 STR Systems diagram

Figure 1 STR System Diagram

## 3.2 Hardware Architecture

Different deployments can use different server configurations, but STR is typically deployed on four servers:

* A web server running Windows Server to host the web application;
* Storage for various application configuration files (e.g., CSS, images, etc.);
* *https://str.cbe.com.et/*server to respond for requests; and
* A SQL server database server to host the STR database.

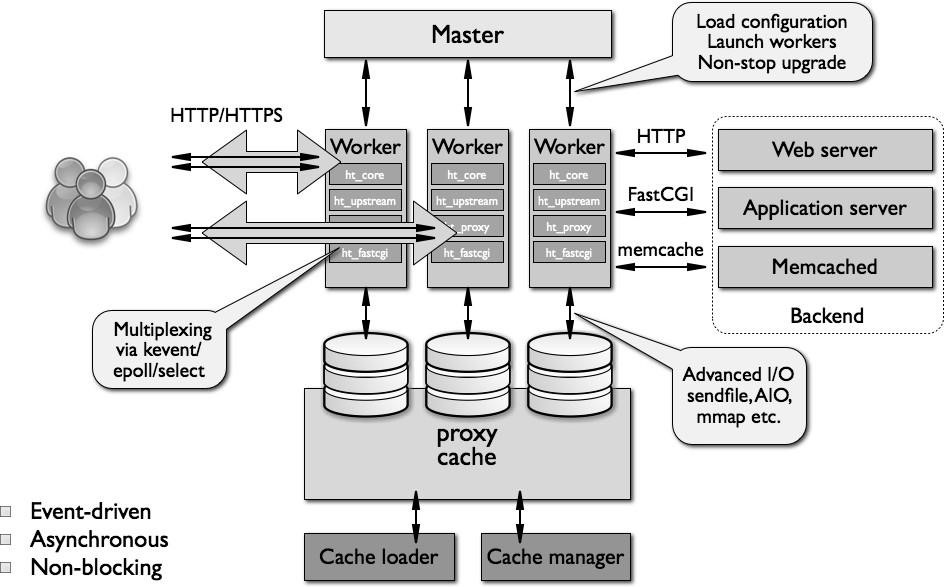


Figure 2 Hardware Architecture

Table 2 Systems descriptions

|  |  |  |
| --- | --- | --- |
|  | SYSTEM | NOTE |
| PROGRAMMING LANGUAGE | C# | C# is a programming language that was created by Microsoft and released in the early 2000s. It is widely used and ranks among the top programming languages in terms of popularity. C# is a general-purpose language, similar to Java or the C language, but it is particularly renowned for its effectiveness in web programming. It offers a robust set of features and a large ecosystem of frameworks and libraries, making it a popular choice for developing web applications and other software solutions. |
| APPLICATION FRAMEWORK | Asp.net MVC | ASP.NET MVC is a framework that extends the ASP.NET platform, allowing developers to build web applications using the C# programming language. ASP.NET MVC combines C# with HTML, CSS, and JavaScript to create dynamic web  applications that run on a web server. When ASP.NET MVC is utilized with C#, it is commonly referred to as "ASP.NET MVC" or ".NET MVC." |
| DEVELOPMENT AND VERSION CONTROL ENVIRONMENT | Git | Git is a version control system that is used for software development and other version control tasks. As a distributed revision control system. Git is free software distributed under the terms of the GNU |
| HOSTING SERVICE | GitLab | GitLab is a web-based Git repository hosting service similar to GitHub. It offers all the distributed revision control and source code management (SCM) capabilities of Git while adding its own unique features. Unlike  Git, which is primarily a command-line tool, GitLab provides a web-based graphical interface and supports desktop and mobile integration. GitLab also offers access control and various collaboration features, including bug tracking, feature requests, task management, and wikis for each project. |
| DATABASE | SQL server | SQL Server is a relational database management system (RDBMS) developed by Microsoft. It emphasizes extensibility and adherence to standards. SQL Server acts as a database server, securely storing data and facilitating retrieval upon request from other software applications. It is capable of handling various workloads, ranging from small single-machine applications to largescale, high-traffic internet-facing applications with numerous concurrent users. SQL Server offers a comprehensive set of features and tools for efficient data management and robust database administration. |
| WEB SERVER SOFTWARE | IIS | IIS (Internet Information Services) is a widely used web server software developed by Microsoft. It is a popular choice for hosting websites and web applications. IIS is an integral part of the Windows Server operating system and is supported on all major server operating systems. It offers robust features and capabilities for serving  HTTP and HTTPS requests, making it a reliable and widely adopted web server solution in the industry. |

## 3.3 Software Architecture

The MVC (Model-View-Controller) architecture is a popular pattern for user-facing applications. It divides the application into three main components: the Model, the View, and the Controller. The Model represents the data and business logic of the application. It includes data structures, database operations, and any other functionality related to data management. The View is responsible for presenting the data to the user, typically through a user interface such as a web page or a mobile app UI. The Controller handles user input and acts as an intermediary between the Model and the View. It processes user requests, updates the Model, and determines which View to display. The MVC architecture promotes separation of concerns and facilitates modular development and maintainability of the application code.

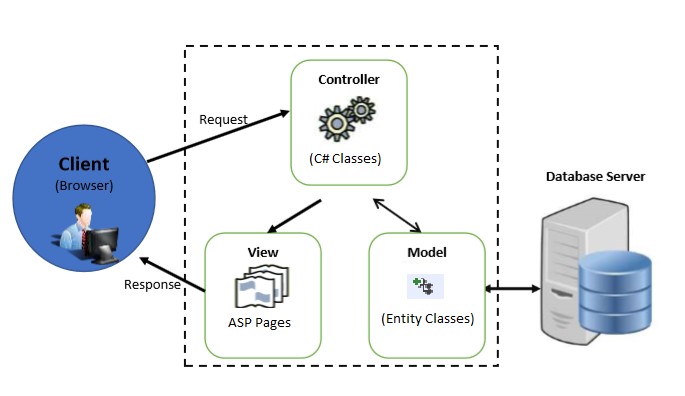


Figure 3 Software architecture tiers

## 3.4 Security Software Architecture

There are a number of principles applied to current and proposed system security.

* Apply security at all layers:
  + Rather than running security appliances (e.g., firewalls) only at the edge of your infrastructure, use firewalls and other security controls on all of your resources (e.g., every virtual server, load balancer, and network subnet).
* Enable traceability:
  + Log and audit all actions and changes to your environment.
* Implement a principle of least privilege:
  + Ensure that authorization is appropriate for each interaction with your IIS resources and implement strong logical access controls directly on resources.

# 4. System Design

The proposed solution is suspicious transaction reporting system. This system enables any CBE employee to report suspicious transactions by filling out the necessary form within the system. The compliance team at the head office receives, reviews, and verifies these reports. Once verified, the reports are exported in PDF or Excel FIC format and sent to the Financial Intelligence Center (FIC) for further analysis and investigation. This streamlined process helps ensure timely reporting of suspicious activities and facilitates effective collaboration between the bank and regulatory authorities.

## 4.1 Business Requirements

System could provide the following functionality of STR reporting System.

Suspicious Transaction Reporting (STR) Requirements

|  |  |
| --- | --- |
| S.No | STR requirements |
| 1 | **Entity making the report (Bank Information)** |
| 1.1 | Bank Making the Report |
| 1.2 | Type of Business |
| 1.3 | Branch Name |
| 1.4 | Date of Report |
| 1.5 | Contact Person Information |
| 1.6 | Contact Person Address Information |
| 2 | **Personal Information** |
| 2.1 | Identification Information (ID type, ID Number…) |
| 2.2 | Address Information |
| 3 | **Transactions Information** |
| 3.1 | Number of Transaction |
| 3.2 | Manner Transaction Conducted |
| 3.3 | Date and Time |
| 3.4 | Amount |
| 4 | **Account Information** |
| 4.1 | Branch Name |
| 4.2 | Account number |
| 4.3 | Account Type |
| 4.4 | Account Opened Date |
| 4.5 | Balance Held |
| 4.6 | Date of Balance Held |
| 4.7 | Beneficiary Account Information |
| 5 | **Association Information** |
| 5.1 | Individual Association Information |
| 5.2 | Company Association Information |
| 6 | **Particulars of Suspicious or Unusual Activity** |
| 7 | **List of Available Documents** |
| 7.1 | Document Name |
| 7.2 | Document Type |
| 7.3 | Document Extension |
| 7.4 | Document Action |
| 8 | **Create Report Field** |
| 8.1 | Export format (CSV and PDF) |
| 8.2 | Report purpose (for Internal and FIC report) |
| 8.3 | Exportability |
| 8.4 | Report Date Range (Monthly, Quarterly, Yearly |
| 8.5 | Date Category (Between, Less than, Greater than, equals or…) |
| 8.6 | Filtering Criteria |
| 8.7 | Reporting values at column (Report by, approved by, report branch…) |
| 9 | **Profile of staff who report the transactions** |
| 9.1 | Branch Name |
| 9.2 | Phone number |
| 10 | **Approve/Reject menu** |
| 11 | **Reject Comment box** |
| 12 | **Reject Notification popup messages via outlook** |
| 13 | **Assign/Re-Assign menu** |
| 14 | **STR Dashboard menu** |
| 15 | **Admin user** |