Functional Requirement Document (FRD) for SWIFT Upgrade

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to outline the functional requirements for upgrading the SWIFT system used for international financial transactions. The upgrade aims to enhance the system's performance, security, and compliance with the latest SWIFT standards.

## 1.2 Scope

The scope of the SWIFT upgrade project includes updating the existing SWIFT infrastructure, ensuring compatibility with the latest SWIFT standards, enhancing security measures, improving system performance, and providing training for users.

## 1.3 Definitions, Acronyms, and Abbreviations

- \*\*SWIFT\*\*: Society for Worldwide Interbank Financial Telecommunication  
- \*\*MT\*\*: Message Type  
- \*\*ISO 20022\*\*: International standard for electronic data interchange between financial institutions  
- \*\*BIC\*\*: Bank Identifier Code

## 1.4 References

- SWIFT Standards Documentation  
- ISO 20022 Documentation  
- Current System Architecture Documentation

# 2. Overall Description

## 2.1 Product Perspective

The SWIFT upgrade is an enhancement to the existing system used by financial institutions to send and receive secure, standardized financial messages. The upgrade will integrate seamlessly with the current system architecture while providing new features and improvements.

## 2.2 Product Functions

- Send and receive SWIFT messages  
- Validate message formats  
- Log and audit transactions  
- Enhance security protocols  
- Support for ISO 20022 standards  
- User management and access control

## 2.3 User Classes and Characteristics

- \*\*System Administrators\*\*: Manage system configuration, user access, and overall system health.  
- \*\*Financial Analysts\*\*: Utilize the system for sending and receiving financial messages.  
- \*\*IT Support Staff\*\*: Provide technical support and troubleshooting.  
- \*\*Compliance Officers\*\*: Ensure transactions comply with regulatory standards.

## 2.4 Operating Environment

- Server OS: Linux/Windows  
- Database: SQL Server/MySQL/Oracle  
- Network: Secure VPN or private network

## 2.5 Design and Implementation Constraints

- Compatibility with existing hardware  
- Compliance with SWIFT and ISO 20022 standards  
- Adherence to organizational security policies

# 3. Specific Requirements

## 3.1 Functional Requirements

### 3.1.1 Message Processing

- \*\*FR1\*\*: The system must support sending and receiving all SWIFT MT and MX messages.  
- \*\*FR2\*\*: The system must validate message formats against SWIFT standards.  
- \*\*FR3\*\*: The system must provide real-time processing of SWIFT messages.

### 3.1.2 Security

- \*\*FR4\*\*: The system must use encryption for all message transmissions.  
- \*\*FR5\*\*: The system must support multi-factor authentication for user access.  
- \*\*FR6\*\*: The system must log all access and transaction activities for audit purposes.

### 3.1.3 Compliance

- \*\*FR7\*\*: The system must support ISO 20022 message formats.  
- \*\*FR8\*\*: The system must generate compliance reports for regulatory bodies.

### 3.1.4 Performance

- \*\*FR9\*\*: The system must process messages with a latency of less than 1 second.  
- \*\*FR10\*\*: The system must handle a minimum of 10,000 transactions per minute.

### 3.1.5 User Management

- \*\*FR11\*\*: The system must allow administrators to create, update, and delete user accounts.  
- \*\*FR12\*\*: The system must provide role-based access control.

## 3.2 Non-Functional Requirements

### 3.2.1 Reliability

- \*\*NFR1\*\*: The system must have an uptime of 99.99%.

### 3.2.2 Scalability

- \*\*NFR2\*\*: The system must scale to support increased transaction volumes.

### 3.2.3 Usability

- \*\*NFR3\*\*: The system must have an intuitive user interface for ease of use.

### 3.2.4 Maintainability

- \*\*NFR4\*\*: The system must support easy maintenance and updates without significant downtime.

### 3.2.5 Security

- \*\*NFR5\*\*: The system must comply with industry-standard security practices.

# 4. System Architecture

## 4.1 Overview

The upgraded SWIFT system will include a client-server architecture with secure communication channels, a robust database for transaction storage, and an intuitive user interface for interaction.

## 4.2 Component Description

- \*\*Message Processor\*\*: Handles sending, receiving, and validating SWIFT messages.  
- \*\*Security Module\*\*: Manages encryption, authentication, and logging.  
- \*\*Compliance Module\*\*: Ensures adherence to ISO 20022 and regulatory requirements.  
- \*\*User Interface\*\*: Provides access for users to interact with the system.

# 5. Implementation Plan

## 5.1 Phases

- \*\*Phase 1\*\*: Requirement Analysis and Design  
- \*\*Phase 2\*\*: Development and Integration  
- \*\*Phase 3\*\*: Testing and Validation  
- \*\*Phase 4\*\*: Deployment and Training  
- \*\*Phase 5\*\*: Maintenance and Support

## 5.2 Milestones

- Completion of requirement analysis  
- Development of core modules  
- Successful testing of system components  
- User training and system deployment  
- Ongoing maintenance and support

# 6. Appendices

## 6.1 Glossary

- \*\*SWIFT\*\*: A global provider of secure financial messaging services.  
- \*\*ISO 20022\*\*: An international standard for electronic data interchange between financial institutions.

## 6.2 Acronyms

- \*\*MT\*\*: Message Type  
- \*\*MX\*\*: Message Exchange  
- \*\*BIC\*\*: Bank Identifier Code