Software Requirements Specification

for

Real-Time Fraud Detection System

Version 1.0 approved

Prepared by Akshat Abhishek

Nitte Meenakshi Institute of Technology

25/04/2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to specify the functional and non-functional requirements of the FraudShield system, a real-time fraud detection solution for financial transactions. The document aims to provide a clear and complete understanding of what the system must accomplish without prescribing how it should be implemented.

1.2 Document Conventions

- **SRS** – Software Requirements Specification
- **ML** – Machine Learning
- **API** – Application Programming Interface
- **UI** – User Interface
- **UAT** – User Acceptance Testing
- **FR** – Functional Requirement
- **NFR** – Non-Functional Requirement

1.3 Intended Audience and Reading Suggestions

This document is intended for stakeholders, developers, system architects, testers, and end users involved in the FraudShield project. Stakeholders can review the Introduction and Overall Description for system understanding, while developers and testers should focus on the Specific Requirements. System architects may refer to the product functions and constraints to guide design. End users can refer to use cases in the Appendices.

Readers are advised to begin with the Introduction and Overall Description for context, then move to the Specific Requirements for implementation and validation details. Appendices provide supplementary information like sample data and use cases.

1.4 Product Scope

FraudShield is designed to detect fraudulent transactions in real time within a financial institution. It will receive transaction data, classify it using pre-defined rules and machine

learning techniques, and generate alerts for suspicious activity. The system supports multiple user roles and provides an interface for reviewing and acting upon flagged transactions.

1.5 References

- *IEEE 830-1998: Recommended Practice for Software Requirements Specifications*
- *IEEE 1012-1998: Standard for Software Verification and Validation*
- *ISO/IEC 9126: Software Engineering - Product Quality*

2. Overall Description

2.1 Product Perspective

FraudShield will function as an independent module within a financial institution's broader transaction infrastructure. It will ingest transaction streams via API and evaluate each transaction using configurable rules and machine learning models. Alerts will be generated when suspicious patterns are detected.

2.2 Product Functions

The FraudShield system is designed to monitor financial transactions in real time, allowing it to identify potential fraudulent activity as it occurs. It analyzes transactions using both predefined detection rules and trained machine learning models to ensure accuracy and adaptability. When suspicious transactions are identified, the system generates alerts and presents them through a user-friendly review interface. It maintains a secure log of all transaction data and detection results, ensuring traceability and auditability. Additionally, administrators are provided with controls to manage detection thresholds, update rules, and configure logic as needed to adapt to new fraud patterns.

2.3 User Classes and Characteristics

The system will be used by three primary user groups. Fraud Analysts are responsible for reviewing and validating transactions that the system flags as potentially fraudulent. These users require access to alert details and historical context. System Administrators manage

system rules, update detection parameters, and monitor performance and activity logs. They ensure that the system remains aligned with institutional policies and regulatory requirements. Lastly, Developers and Data Scientists are responsible for maintaining and updating the machine learning models. They analyze model performance and retrain detection algorithms using new data to improve system accuracy over time.

2.4 Operating Environment

FraudShield will operate in a Linux-based server environment or within a containerized infrastructure. It will expose a web-based API for transaction input and a web dashboard for user interaction. The system will be built using technologies such as Python for the core logic, PostgreSQL for database management, and Docker for deployment and containerization. It is designed to function effectively in both cloud-based and on-premise infrastructures, providing flexibility for different deployment needs.

2.5 Design and Implementation Constraints

The system must enforce strong data security by ensuring all transaction data is encrypted both during transmission and while at rest. It must comply with PCI-DSS standards, which are essential for handling sensitive financial information in a secure and legally compliant manner. Furthermore, the system is expected to operate in a high-throughput, low-latency environment, which places performance constraints on all components of the architecture.

2.6 User Documentation

The FraudShield system will be delivered with comprehensive user documentation to support different user roles, including analysts, administrators, and technical staff. The user documentation components will include a ***User Manual****, which provides step-by-step instructions for accessing the system, reviewing alerts, and managing configurations. An* ***Administrator Guide*** *will be provided to explain role management, rule updates, and system maintenance tasks. Additionally,* ***on-line help*** *will be integrated into the user interface to assist users in real time, offering tooltips, FAQs, and contextual guidance.*

Interactive ***tutorials*** *and* ***walkthroughs*** *will also be developed for first-time users to understand system workflows and functionality. All documentation will be available in* ***PDF format*** *for offline access and in* ***HTML format*** *for online access via the system dashboard. The content will follow standard technical writing practices and adhere to user documentation guidelines such as* ***Microsoft Manual of Style*** *or* ***ISO/IEC 26514*** *for consistency and clarity.*

2.7 Assumptions and Dependencies

The system assumes that historical transaction data is available and accessible for machine learning model training and evaluation. Real-time transaction input is expected to be delivered in a consistent and predefined format that aligns with the system's API structure. Additionally, it is assumed that all users of the system will have secure login credentials and appropriate access roles defined in accordance with institutional policies.

3. External Interface Requirements

3.1 User Interfaces

The fraud detection system will feature an intuitive and user-friendly web-based interface. This interface will be accessible across multiple devices, including desktops, laptops, and mobile devices, ensuring that users can interact with the system conveniently from anywhere. The primary users of the system will include fraud analysts, administrators, and security officers, and each will have personalized access based on their roles. Users will be able to log in securely, manage fraud detection alerts, configure system settings, and view real-time dashboards showcasing transaction analyses. Additionally, the user interface will allow administrators to configure fraud detection thresholds, and analysts can examine flagged transactions in detail to make informed decisions.

3.2 Hardware Interfaces

The system is designed to operate on standard server hardware, including high-performance CPUs and sufficient memory to handle large volumes of real-time transaction data. Additionally, the system may interface with third-party hardware, such as point-of-sale (POS) devices or biometric authentication hardware (e.g., fingerprint scanners, facial recognition cameras), depending on the integration needs. Communication between hardware systems will be achieved using secure network protocols, ensuring that data transferred between devices and the fraud detection system remains safe and intact.

3.3 Software Interfaces

The fraud detection system must interface with various existing financial software platforms, including transaction processing systems, databases, and other enterprise applications used by the financial institution. This integration allows seamless data flow from transaction logs to the fraud detection system for real-time analysis. The system will

also interact with machine learning models (either internal or external) that help identify patterns of fraudulent activity. Furthermore, it will integrate with external APIs to gather real-time data on potentially fraudulent accounts or blacklisted entities, enhancing detection capabilities. The system will allow exporting of reports in formats such as CSV and PDF for use by internal teams.

3.4 Communications Interfaces

For secure communication, the fraud detection system will rely on HTTPS for all web-based transactions. It will have built-in integration with email and SMS gateways to send real-time alerts and notifications to administrators and relevant personnel when fraud detection events occur. The system will also feature an API that allows it to interact with other internal systems for data sharing and coordination of fraud response actions. Secure communication channels will be a priority to ensure that all notifications and alerts are transmitted safely and reliably.

4. System Features

4.1 Fraud Detection

The core functionality of the system will be real-time fraud detection. This feature will use predefined rules, patterns, and machine learning models to analyze every incoming financial transaction for signs of fraud. Each transaction will be examined for factors such as large amounts, unusual behavior for the account holder, and known patterns of fraud. If any transaction matches these criteria, it will be flagged for further analysis. The system will also continually learn from previous transactions, improving its ability to detect fraud over time.

4.2 Alerts and Notifications

Upon detecting suspicious activity, the system will send alerts in real-time to authorized users. These alerts will include detailed information about the flagged transaction, such as the account involved, the nature of the suspicious activity, and suggested actions. The system will provide flexible configuration options, allowing users to set thresholds for when alerts should be triggered. Alerts will be sent via email and SMS to ensure immediate attention is given to high-priority events, and users can adjust their preferences for different types of alerts based on severity and frequency.

4.3 Reporting

The system will offer detailed reporting capabilities, enabling users to generate various types of reports on detected fraud incidents, transaction trends, and system performance. These reports will help administrators and analysts review past activities, track fraudulent trends, and make data-driven decisions. Reports can be customized to filter data based on specific time periods, fraud types, or user-defined criteria. These reports will be exportable in formats like CSV and PDF, making them suitable for both internal audits and external regulatory compliance.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The fraud detection system must be capable of processing a high volume of financial transactions in real-time. Given that financial institutions often deal with thousands of transactions per second, the system should be able to process and analyze at least 10,000 transactions per second without significant performance degradation. The detection algorithms should respond within 5 seconds of a transaction being submitted, ensuring no noticeable delay in the fraud detection process. The system should also be scalable, allowing for easy expansion as transaction volumes grow or during peak transaction periods.

5.2 Safety Requirements

To ensure the safety of both the system and its users, the fraud detection system must be built with redundancies in place. For example, in case of a system failure, there should be automatic failover procedures, ensuring continuous transaction processing. However, fraud detection services may experience brief downtimes in such situations, but these should not affect critical transaction processes. Additionally, the system should maintain a robust data backup strategy to recover data in case of catastrophic failures.

5.3 Security Requirements

Security is of utmost importance for a fraud detection system, particularly when dealing with sensitive financial data. All user data, including personal and transaction data, should be encrypted using industry-standard encryption methods, such as AES-256. The system

should support multi-factor authentication (MFA) for all users, especially administrators, to enhance security. Role-based access control (RBAC) will ensure that users only have access to data and features they are authorized to interact with. All user actions within the system will be logged for auditing purposes, and the system must be compliant with regulatory standards, such as PCI DSS, to ensure the protection of sensitive information.

5.4 Software Quality Attributes

Reliability is key for a fraud detection system. The system should operate with a target uptime of 99.99%, ensuring minimal downtime. In addition, the system must be scalable, capable of handling increased load as transaction volumes grow. Maintenance should be streamlined, with the system allowing for easy updates, patches, and enhancements without significant system downtime. Furthermore, the user interface must be intuitive, minimizing the need for extensive user training.

5.5 Business Rules

The system will have certain business rules that dictate the conditions under which fraud detection occurs. For example, any transaction above a defined threshold, or one that occurs from an IP address known for fraudulent activity, should automatically be flagged for review. Additionally, transactions that deviate from a user's typical spending behavior will trigger an alert. These business rules will be adjustable, allowing for the fine-tuning of fraud detection based on real-time data and changing threat patterns.

6. Other Requirements

The fraud detection system must be capable of future-proofing to accommodate new financial systems or regulatory changes. It should allow for seamless integration with new banking systems, financial data sources, or third-party fraud detection services that may be introduced in the future. Furthermore, the system must comply with local and international data protection and privacy laws to ensure that all personal and financial data is handled appropriately.

Appendix A: Glossary

- ***Fraud Detection:*** *The process of identifying fraudulent activities by analyzing transaction patterns and behaviors.*

- ***Real-Time Processing:*** *The ability to process transactions and detect fraud as soon as they occur.*
- ***Machine Learning:*** *A technique used by the system to improve fraud detection accuracy by learning from historical data and adapting to new fraud patterns.*

Appendix B: Analysis Models

- *Flow diagrams illustrating the fraud detection process.*
- *Use case models showing how different roles interact with the system.*
- *Data flow diagrams demonstrating how data moves within the system for fraud detection.*

Appendix C: To Be Determined List

- *Final decision on the selection of machine learning algorithms.*
- *Details on external fraud detection APIs or systems that may be integrated in the future.*
- *Determination of acceptable fraud thresholds and risk parameters for the institution.*