CLARK CONSULTING

Software Design Specification

CredSafe - Credit Card Fraud Detection System

Revision 2.0

Teammates:

1. Haani Behzad Kuniyil (22MIC0179)

2. Raakesh MJ (22MIC0180)

3. Ashin B Shankar (22MIC0178)

4. Sujal Ravindra Dixit (22MIC0115)

Table of Contents

Revision History

Version	Name	Reason For Changes	Date
1.0	*Sujal Dixit* *Haani Behzad* *Raakesh MJ* *Ashin B Shankar*	*Initial Revision*	*09/03/2023*
2.0	*Sujal Dixit* *Haani Behzad* *Raakesh MJ* *Ashin B Shankar*	*Final Revision*	*24/03/2023*

1. Introduction

1.1 Purpose

In the digital age, the convenience of credit card transactions has been paralleled by an alarming rise in fraudulent activities. These activities not only lead to significant financial losses for individuals and institutions but also erode trust in the financial ecosystem. Recognizing the critical need to address this challenge, the development of a robust Credit Card Fraud Detection System has become imperative. The goal of the Credit Card Fraud Detection System is to identify instances of fraud involving credit card transactions. It analyzes transaction data using a variety of algorithms and approaches to spot trends that point to fraudulent activity. The Credit Card Fraud Detection System's software design specifications are described in this paper.

1.2 System Overview

The Credit Card Fraud Detection System is an advanced, intelligent solution designed to monitor, analyze, and identify fraudulent activities within credit card transactions. In essence, the system uses an advanced combination of machine learning models, anomaly detection algorithms, and rule-based logic to seamlessly interact with transaction processing pipelines and examine every transaction in real-time. It compares known fraud trends and learning behaviors from historical data with a variety of transaction variables, including amount, location, merchant type, and transaction frequency. When the system notices a transaction that looks suspect, it flags it for evaluation right away and sets off an alert to alert the appropriate parties. This allows for quick action to stop possible fraud. The solution also has a dynamic reporting module that provides informative insights and trends to help fraud management teams make better decisions. The Credit Card Fraud Detection System provides a strong defense against fraud in the context of digital banking by means of its extensive design and implementation, guaranteeing the security and integrity of credit card transactions.

1.3 Design Map

Data Collection Module: This module feeds the system with transaction data for analysis after collecting it from a variety of sources, such as banks, merchants, and payment gateways.

Preprocessing Module: To get ready for analysis, the raw transaction data is put through preprocessing procedures such feature extraction, normalization, and outlier detection.

Fraud Detection Module: Based on trends and abnormalities in the preprocessed data, this module uses a combination of machine learning algorithms, anomaly detection approaches, and rule-based systems to identify possibly fraudulent transactions.

Alerting Module: In order to enable prompt intervention and fraud prevention, the system creates alerts and notifies pertinent stakeholders when it detects a questionable transaction.

Reporting Module: This module helps fraud management teams make decisions and develop strategies by producing thorough reports and visualizations of fraudulent activity, trends, and patterns.

1.4 Definitions and Acronyms

Transaction data: Details on credit card transactions, such as the amount, time and date of the transaction, location, merchant information, and cardholder details, among others.

Preprocessing: The act of cleansing and getting ready unprocessed data for analysis; it may include data transformation, feature extraction, outlier identification, and normalization.

Anomaly detection: Finding odd patterns that deviate from conventional behavior, and it's usually used to spot questionable transactions.

Rule-based Systems: These systems, which are employed here to identify transactions that suggest fraud, make decisions by applying user-defined rules to data.
The process of turning unprocessed data into a collection of features that may be examined further is known as feature extraction.

Alerting Mechanism: A part of the system that alerts stakeholders to possibly fraudulent transactions so that they can look into them right away and take appropriate action.

API: Application Programming Interface - a set of rules that allows different software entities to communicate with each other.

PCA: Principal Component Analysis - a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables.

2. Design Considerations

All design considerations were handled in the Phase 1 Release of this paper.

2.1 Assumptions

The design and development of the Credit Card Fraud Detection System are based on several critical assumptions. These assumptions play a pivotal role in guiding the system's architecture, technology choices, and operational strategies. Understanding these assumptions is essential for stakeholders to set realistic expectations and plan for potential challenges. These include High Volume Handling, Data Quality, Seamless Integration, Consistent Fraud Patterns, Regulatory Compliance, User Competency, Robust Security.

2.2 Constraints

Several constraints may impact the design, development, and deployment of the Credit Card Fraud Detection System. Understanding these constraints is crucial for stakeholders to manage expectations and plan accordingly. Key constraints are Resource Limitations, Data Privacy, Legacy Systems Integration, Scalability, Performance Requirements, User Acceptance.

2.3 System Environment

Through redundancy mechanisms, the Credit Card Fraud Detection System is designed to function in a complex environment that includes both physical and virtual server infrastructures, enabling scalability and high availability. It operates with a variety of operating systems, manages data using a combination of relational and NoSQL databases, and performs machine learning and data processing tasks using Python and frameworks like TensorFlow and Scikit-learn. The network infrastructure of the system is protected by security methods, such as firewalls and encryption, to guarantee safe data transfer and adherence to legal requirements like GDPR and PCI DSS. Financial institution APIs and other third-party services are examples of external dependencies that make real-time data analysis and sharing possible. The system lifecycle is supported by development and testing environments that are outfitted with contemporary IDEs, version control, and staging servers. These environments provide deployment flexibility across on-premises data centers and cloud platforms such as AWS, Azure, or Google Cloud, thereby effectively meeting organizational needs and operational requirements.

2.4 Design Methodology

The design methodology for the Credit Card Fraud Detection System integrates a comprehensive approach that begins with an in-depth analysis of both functional and non-functional requirements to ensure alignment with stakeholder needs and regulatory standards. Using a modular design approach to improve system scalability and maintainability, prototyping is used for early validation, and an iterative and agile development framework is adopted. The methodology places a strong emphasis on flexibility and continual feedback. The build, testing, and deployment processes are automated and streamlined through the use of continuous integration and deployment (CI/CD) techniques, guaranteeing the timely and dependable delivery of updates. The development lifecycle incorporates quality assurance, and there are strict testing procedures in place to quickly detect and address problems. Additionally, comprehensive documentation is kept up to date to promote easy comprehension and maintenance, guaranteeing that the system not only effectively satisfies present needs but can also be modified to accommodate evolving requirements and technological advancements.

2.5 Risks and Volatile Areas

There are a number of dangers and unstable regions that must be carefully considered when developing and implementing the Credit Card Fraud Detection System. Regulatory changes that affect compliance requirements, the possibility of data breaches or unauthorized access compromising sensitive transaction information, the possibility of errors in fraud detection algorithms resulting in false positives or negatives, and the dependence on third-party APIs and data sources for real-time transaction data are a few of these risks. Furthermore, as transaction volumes rise, scalability problems can appear, and performance problems or system outages could impair real-time fraud detection capabilities. Proactive steps like strong security standards, extensive algorithm testing and validation, keeping up with legislative changes, creating backup plans in case of API failures, and putting in place scalable infrastructure to enable system expansion and resilience are all necessary to mitigate these risks.

3. Architecture

The Credit Card Fraud Detection System's architecture is built to quickly identify suspected fraud, process and analyze transaction data in real-time, and assist with decision-making. To guarantee excellent accuracy and performance, the architecture is multi-tiered, flexible, and makes use of both machine learning techniques and data-driven insights.

3.1 Overview

The Credit Card Fraud Detection System's architecture is a strong, multi-tiered structure intended to identify and stop fraudulent transactions instantly. At its core, the system combines a number of different components, such as sophisticated data storage solutions for organizing large volumes of information, data ingestion mechanisms for gathering transaction data, and a potent processing layer that uses rule-based logic and machine learning algorithms to analyze transactions for possible fraud. Effective monitoring and control of fraud detection operations is made possible by a user-friendly interface, and decision-making and alarm systems enable prompt action on suspicious actions. Together with security safeguards and compliance standards to secure sensitive data, the architecture also places a heavy emphasis on powerful integration capabilities, enabling smooth communication with external financial systems and third-party services.

3.2 Subsystem, Component, or Module 1 …N

The Fraud Detection Engine is one architectural component that has to be explained in more detail. This part is in charge of looking through transaction data to find any fraud cases. The Fraud Detection Engine is made up of the following subcomponents:

Feature Engineering Module: This subcomponent is in charge of drawing pertinent features—like transaction amount, location, time, and merchant category—from transaction data. The fraud detection models use these features as input variables.

Machine Learning Models: To identify fraudulent transactions, the Fraud Detection Engine uses a variety of machine learning models. Neural networks, decision trees, random forests, and logistic regression are some of these models. To identify trends suggestive of fraud, each model is trained using transaction data from the past.

Rule-based Systems: Rule-based systems are used to flag transactions based on established rules or thresholds, in addition to machine learning models. Transaction thresholds, the frequency of transactions from a specific region, or odd transaction times are a few examples of these rules.

Ensemble Techniques: Using ensemble techniques like bagging, boosting, or stacking can increase the accuracy of detection. These methods generate a final forecast by combining the results of several rule-based systems or machine learning models.

Transaction data is transferred from the preprocessing component to the feature engineering module, where pertinent features are extracted, as part of the interactions inside the Fraud Detection Engine. Following that, these attributes are sent for analysis to rule-based systems and machine learning models. To arrive at a final fraud detection conclusion, the output of these models and systems—which may contain predictions or flags for possibly fraudulent transactions—is then aggregated or integrated using ensemble approaches.

3.3 Strategy 1…N

The choice was made for the main machine learning models in the Fraud Detection Engine to be a blend of Random Forest and Neural Networks. This approach makes use of Random Forest because it is resistant to overfitting and can manage imbalanced datasets, which is a typical problem in fraud detection. Because they can discover intricate patterns and connections in huge datasets, neural networks—especially deep learning models—were selected to provide a sophisticated comprehension of transaction data.

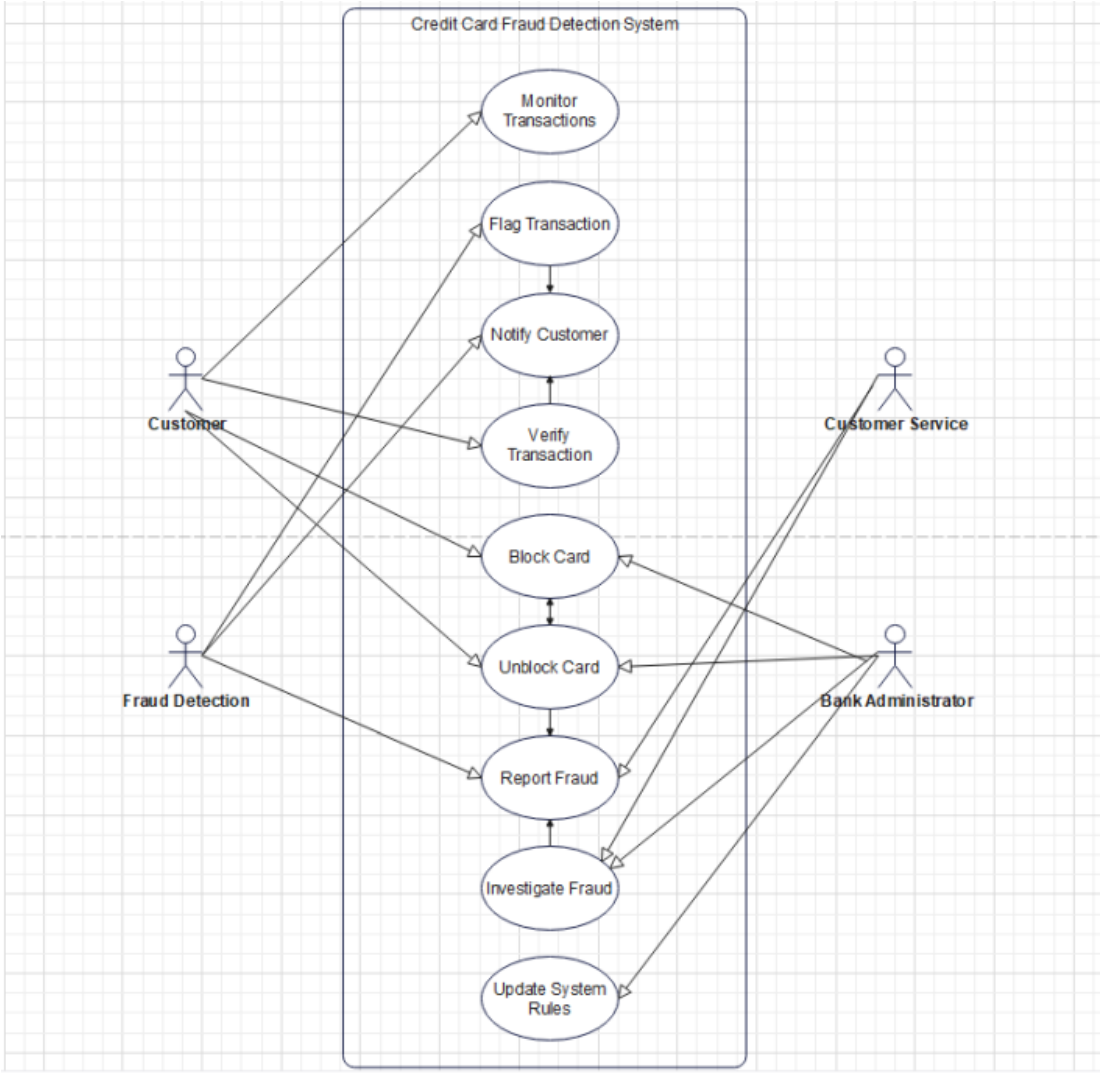
Alternatives:
Logistic Regression: Although logistic regression is simple and effective for binary classification tasks, its inability to handle complicated, non-linear correlations in the data—which are frequently present in fraud detection scenarios—led to its eventual rejection.

Support Vector Machines (SVM): Because of its efficiency in high-dimensional environments, SVMs were taken into consideration. But because of their high computing overhead and scalability issues—particularly when handling the massive volumes of transaction data found in real-time fraud detection systems—they were judged less appropriate.

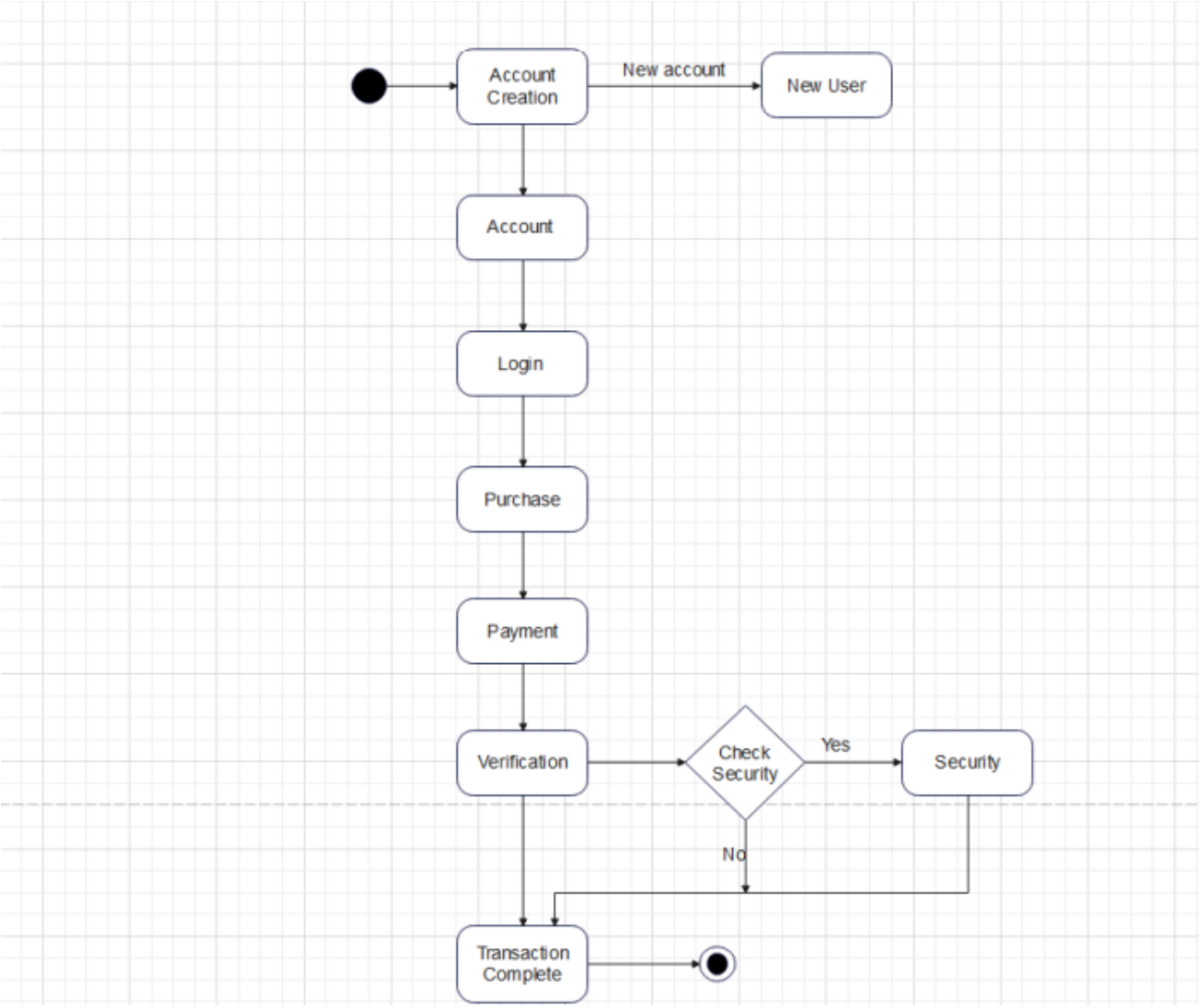
By integrating these models, the system can maintain high accuracy and adaptability in identifying fraudulent transactions, while also ensuring scalability and performance efficiency. This strategic choice also allows for continuous improvement and refinement as new data becomes available and fraud patterns evolve, ensuring the system remains effective over time.

3.4 Workflow Report

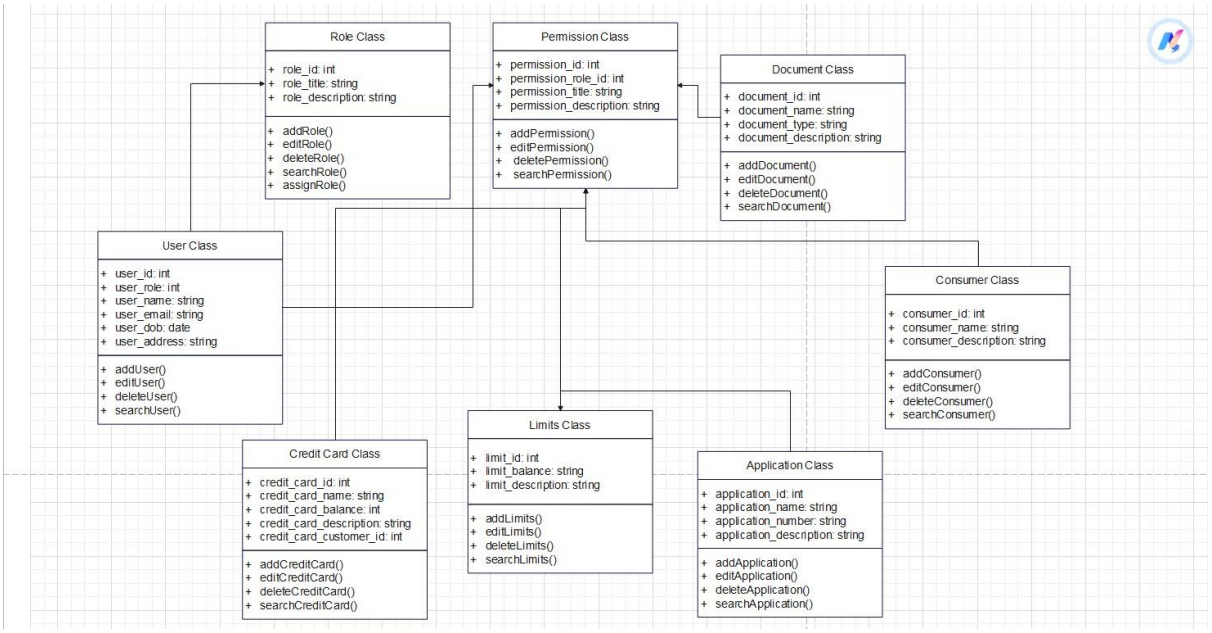
USER CASE DIAGRAM:



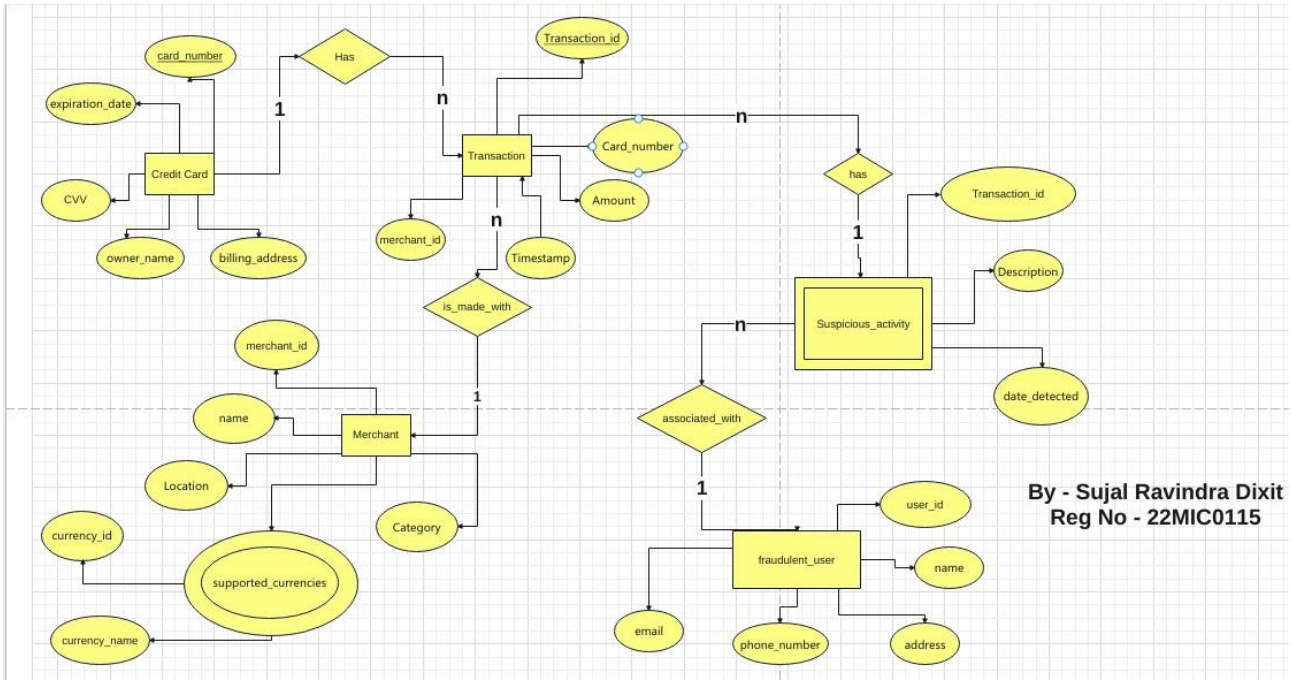
STATE DIAGRAM:



CLASS DIAGRAM:



ER DIAGRAM:



4. Database Schema

4.1 Tables, Fields and Relationships

The structure of the suggested database schema for the Credit Card Fraud Detection System is made to effectively handle and evaluate transaction data as well as associated account and customer data. The Transactions Table, which contains the specifics of each transaction, is the central component of the schema. It is connected to the Accounts Table, which contains information relevant to individual accounts, and to the Customers Table, which contains the personal data of the account holders. The FraudulentTransactions Table, which logs transactions marked as fraudulent and links back to the Transactions Table for a detailed picture of the incident, is created to solve the issue of fraud detection. This table lists the investigators for each case and is linked to the Investigators Table. Other tables, like the Machine Learning Models and Rules Tables, monitor the effectiveness and application of different rule-based approaches and detection algorithms in order to improve the fraud detection process.

4.1.1 Databases

Relational Database: PostgreSQL, NoSQL Database: MongoDB, Time-Series Database: InfluxDB, Graph Database: Neo4j. This multi-database strategy provides a comprehensive data management framework for the Credit Card Fraud Detection System.

4.1.2 New Tables

*In the Credit Card Fraud Detection System, the **Transactions Table** records each credit card transaction; the **Accounts Table** holds account-specific data; the **Customers Table** stores personal information of the cardholders; the **FraudulentTransactions Table** logs transactions flagged as fraudulent; the **Investigators Table** contains information about individuals investigating fraud cases; the **MachineLearningModels Table** tracks details and performance of models used in fraud detection; the **Rules Table** manages rule-based conditions for fraud detection; the **FraudPatterns Table** stores identified patterns of fraudulent behavior; the **ModelPerformance Table** tracks performance metrics of various machine learning models; the **AlertLogs Table** records alerts generated by the fraud detection system; and the **InvestigationRecords Table** maintains records of investigations into flagged transactions, all contributing to a comprehensive framework for efficient fraud detection and management.*

4.1.3 New Fields(s)

List any new tables that will be needed, for each one including table name, table description, and related tables.

Table Name	Field Name	Data Type	Allow Nulls	Field Description
Transactions	FraudScore	int		A numerical score indicating the likelihood of a transaction being fraudulent, calculated by the fraud detection engine

Accounts	LastTransactionDate	Date		Date of the most recent transaction associated with the account.
FraudulentTransactions	InvestigationOutcome	Varchar(50)		Outcome of the investigation (Fraud Confirmed, False Alarm, etc.).
ModelPerformance	DeploymentStatus	Varchar(50)	yes	Indicates whether the model is currently deployed in the production environment (Deployed/Not Deployed).
AlertLogs	ResolutionDate	Date		Date when the alert was resolved or closed.

4.1.4 Fields Change(s)

For each field change (such as data types, required/not required, or renaming), please complete a row of the following table. (Insert additional rows as needed.)

Table Name	Field Name	What to change?
Transactions	TransactionAmount	Required/Not Required
Accounts	AccountStatus	Renaming – Status to AccountStatus
Transactions	FraudScore	Addition – N/A to INT

4.1.5 All Other Changes

Indexes: Implement indexes on frequently queried fields such as TransactionDate in the Transactions Table, AccountNumber in the Accounts Table, and AlertStatus in the AlertLogs Table to optimize query performance and speed up data retrieval processes.
Stored Procedures: Develop stored procedures for common data manipulation tasks such as inserting new transactions, updating account information, and querying alert logs. These stored procedures will streamline database interactions and ensure consistency in data operations across the system.

Relationships: Review and optimize existing relationships between tables to ensure data integrity and enforce referential integrity constraints. Consider adding foreign key constraints and cascade options where appropriate to maintain data consistency.

Security Settings: Enhance database security measures by implementing role-based access control (RBAC) to restrict access to sensitive data and functionalities based on user roles and privileges. Additionally, encryption at rest and in transit to safeguard data against unauthorized access and ensure compliance with data protection regulations.

Monitoring and Maintenance Plans: Establish regular monitoring and maintenance plans to monitor database performance, identify potential bottlenecks, and proactively address issues. Schedule routine backups, index maintenance tasks, and database optimizations to ensure the system operates smoothly and efficiently.

4.2 Data Migration

A systematic procedure should be followed in order to transfer current data from the Credit Card Fraud Detection System to new tables and fields. To begin with, a complete evaluation of the existing data is

necessary in order to comprehend its quality, organization, and mapping to the new schema. This entails locating the source data that needs to be moved, evaluating its quality, and figuring out what needs to be cleaned up or transformed. The data must be meticulously mapped to the new schema after the assessment, taking care to address any inconsistencies or mismatches. The data must then be extracted, the necessary transformations must be applied to guarantee that it is compatible with the new structure, and the data must be cleaned to improve quality. The data is imported into the new tables and fields after it has been converted. To protect relational linkages and data integrity, this critical stage needs to be attended to. Thorough validation and verification after loading are essential to guarantee the transferred data is true to its original source and meets quality requirements. Thorough testing of the data in the new setting confirms both the system's operation and its integrity. The data is deployed to the production environment after testing and validation go well, and continuous monitoring is done to quickly address any issues that may arise after the migration. Clarity and accountability are maintained throughout the migration project by thoroughly documenting the whole process, which also serves as a useful reference for upcoming migrations and audits. Data integrity and system dependability are maintained during the smooth transition to the new system configuration thanks to this systematic method.

5. High Level Design

5.1 Data Flow:

The Data Ingestion Module initiates the system's data flow by gathering transaction data from sources like banks, retailers, and payment gateways. Before entering the Fraud Detection Engine, this data is preprocessed to improve its quality. To identify possible fraud cases, the Fraud Detection Engine examines transaction patterns utilizing rule-based systems and machine learning techniques. The Alerting and Decision Module receives the flagged transactions and sends alerts for additional inquiry. In the meantime, stakeholders can gain insights into fraud detection performance and trends through the extensive reports and visualizations generated by the Reporting and Analytics Module. In the Credit Card Fraud Detection System, data flows seamlessly through a meticulously orchestrated process, starting with data collection from diverse sources and undergoing initial processing and feature engineering to prepare it for analysis. The Fraud Detection Engine, powered by machine learning models and rule-based algorithms, scrutinizes each transaction for suspicious patterns, flagging potential fraud cases for further investigation within the Alerting and Decision Module. Here, human analysts assess flagged transactions, leveraging available tools and data to make informed decisions. The system operates within a feedback loop, continuously refining its algorithms based on analysts' decisions and investigation outcomes. Meanwhile, the Reporting and Analytics Module generates insights into system performance and fraud trends, empowering stakeholders with actionable information for strategic decision-making and system optimization. Through this comprehensive data flow, the system ensures thorough fraud detection while minimizing false positives, thereby safeguarding against fraudulent activities in the digital transaction landscape.

5.2 User Interface Modifications

Dashboard Updates: Redesigning the dashboard to offer quick access to important performance data including fraud detection rates, volume of alerts, and investigation status. Users can immediately spot emerging fraud tendencies and prioritize their actions by using interactive charts and graphs that depict trends and patterns.

Alert Management Interface: Redesigning the alert management interface will make it easier to examine and handle transactions that have been flagged. By putting sophisticated filtering and sorting tools into place, users may effectively prioritize warnings according to risk level, transaction type, or severity. Including collaboration tools like comments and annotations also makes it easier for team members to communicate and share expertise while conducting an inquiry.

Presenting Real-time Monitoring solutions: These solutions offer real-time updates on transaction activity as well as fraud alarms. By identifying regions with the highest concentration of fraudulent activity, interactive heat maps and geographic visualizations enable analysts to proactively monitor suspicious behavior and take preventative measures to reduce risks.

configurable Workflows: Providing users with the ability to establish and automate routine tasks and decision-making processes through the use of configurable workflow templates. Users can standardize and expedite regular processes, freeing up time for more in-depth analysis and research, by customizing workflows to their unique needs and preferences.

5.3 Workflow sub-processes

Workflow sub-processes in the Credit Card Fraud Detection System are essential for coordinating the series of choices and activities involved in fraud detection and mitigation. These supporting procedures optimize the operational workflow, guaranteeing effective management of flagged transactions and promoting stakeholder collaboration. Important workflow subprocesses consist of:

Triage and Prioritization of Alerts: The initial sub-process after receiving alerts from the Fraud Detection Engine include triaging and ranking flagged transactions according to predetermined standards including risk level, transaction value, and client history. Lower-priority warnings may be subjected to additional automatic analysis or postponed for manual assessment, but high-priority alerts that suggest possible cases of large-scale fraud or substantial financial damage are escalated for prompt response.

Investigation and Analysis: The investigation and analysis sub-process involves analyzing flagged transactions in detail to establish their validity and spot any possible fraudulent behavior, after alert priority. To obtain further context, fraud analysts use a variety of techniques and data sources, including as transaction logs, customer profiles, and past transaction patterns.

Reporting and Documentation: Careful reporting and documentation sub-processes make sure that all decisions and activities are appropriately documented for audit and compliance purposes throughout the workflow. This entails recording in a central repository the results of investigations, the reasoning behind decisions, and the resolutions. Frequent reporting procedures enable datadriven decision-making and strategic planning by giving stakeholders insight into fraud detection performance, trends, and new risks.

6. Low Level Design

The low-level design of the Credit Card Fraud Detection System delves into the specifics of individual modules, detailing the implementation of functionalities and interactions between components. Below are descriptions of key modules along with their design considerations:

6.1 Component modules

6.1.1 Data Ingestion Module

The Data Ingestion Module is in charge of gathering transaction data and storing it in a centralized database for further processing from a variety of sources, such as banks, merchants, and payment gateways. Establish reliable data connections and adapters that handle a variety of protocols and formats to enable smooth integration with a variety of data sources.

- Implement data intake pipelines that can handle large volumes of data and elegantly recover from failures to ensure scalability and fault tolerance.
- Include procedures for validating and cleaning data to guarantee consistency and integrity of data before storing it.

6.1.2 Preprocessing Component:

The Preprocessing Component carries out operations including feature extraction, normalization, and outlier detection to get ready for analysis of the raw transaction data.

Create modular pretreatment pipelines that enable the preprocessing processes to be configured and customized in a flexible way according to certain data properties.

Large-scale data preprocessing jobs can be effectively handled by utilizing scalable data processing frameworks like Apache Spark or Apache Beam.

Use caching techniques to prevent repeating calculations for frequently accessed data and to store interim preprocessing results.

6.1.3 Fraud Detection Engine

In order to identify possibly fraudulent activities, the Fraud Detection Engine examines preprocessed transaction data utilizing rule-based systems and machine learning algorithms.

Choose the right machine learning models (such as random forests, logistic regression, and neural networks) depending on the data that is available and the type of fraud detection issue at hand.

Create a modular framework that can quickly integrate newly produced or updated models and support multiple detection techniques.

Use real-time scoring systems to create fraud scores or probability by comparing transactions to detection algorithms.

6.1.4 Alerting and Decision Module

The Fraud Detection Engine generates warnings, which are then received by the Alerting and Decision Module, which helps fraud analysts make decisions.

Provide alert management systems with user-friendly interfaces that make it easy for fraud analysts to examine, rank, and look into transactions that have been detected.

Install automated decision support systems to aid analysts in making defensible conclusions by offering suggestions and insights derived from analysis results.

Include collaborative elements like case management tools and discussion threads to let analysts communicate and share expertise.

6.1.5 Other Changes

Several changes and improvements are taken into consideration in the low-level design area in order to improve the Credit Card Fraud Detection System beyond its fundamental features. These changes include a wide range of enhancements, from user interface improvements to performance optimizations. To improve system efficiency and scalability, performance improvements could include parallelizing data processing operations, putting caching systems in place, or improving algorithms. Refinements to user interfaces, such as workflow redesigns, the addition of user-friendly data visualization tools, and the application of responsive design principles for cross-device compatibility, are aimed at improving usability and user experience. To increase system capabilities, integration with external services or APIs may also be investigated. Examples of such integrations include the addition of geolocation data for improved fraud detection and real-time transaction monitoring through integration with payment gateway APIs.

6.2 Workflow sub-processes

Data Collection and Preprocessing: To ensure robustness and reliability in the system's functioning, every step of the painstaking low-level detailed design workflow for a credit card fraud detection system is closely examined through the prism of test cases and real-world instances. For example, a test case is used to confirm that the system appropriately processes different data formats and anonymizes sensitive data, such as customer IDs, during the first data collecting and preprocessing phase. To protect the privacy of personal data, a transaction timestamped "09-152024 14:00" in MM-DD-YYYY format, for instance, should be changed to a predefined format (such as ISO 8601). Following this, transactions undergo a thorough analysis process known as feature extraction, from which attributes including transaction amount, merchant type, and time of day are extracted. For example, notifications are triggered when a transaction differs significantly from a customer's typical spending pattern.

Post Detection Actions: Test cases verify the system's capacity to react fast and precisely to verified fraud scenarios. Post-detection steps, such restricting transactions and returning unlawful funds, are carefully carried out. Furthermore, the feedback loop for model improvement incorporates fresh information and learnings from previous fraud incidents to continuously improve the system's detection skills. Stakeholders are empowered to make strategic decisions and enhance compliance efforts by gaining actionable insights into fraud patterns and system performance through thorough reporting and analytics. At the end of the day, the Credit Card Fraud Detection System runs with the highest precision and efficacy, protecting against fraudulent activities in the digital transaction environment, thanks to the integration of test cases and real-world instances at every level.

7. User Interface Design

This section outlines the user interface (UI) design for the credit card fraud detection system. It focuses on the key functionalities and screen layouts to guide the development process. It is crucial for providing an intuitive and efficient experience for users. The design focuses on presenting relevant information clearly and facilitating quick decision-making processes for fraud detection specialists. Below are descriptions of key aspects of the user interface:

7.1 Targeted Users

Fraud Analysts: Users with expertise in analyzing financial data and identifying fraudulent patterns.

Customer Service Representatives (CSRs): Users who interact with cardholders regarding flagged transactions.
Professionals in risk management are those who create risk profiles, set fraud detection guidelines, and keep an eye on the overall performance of the system.

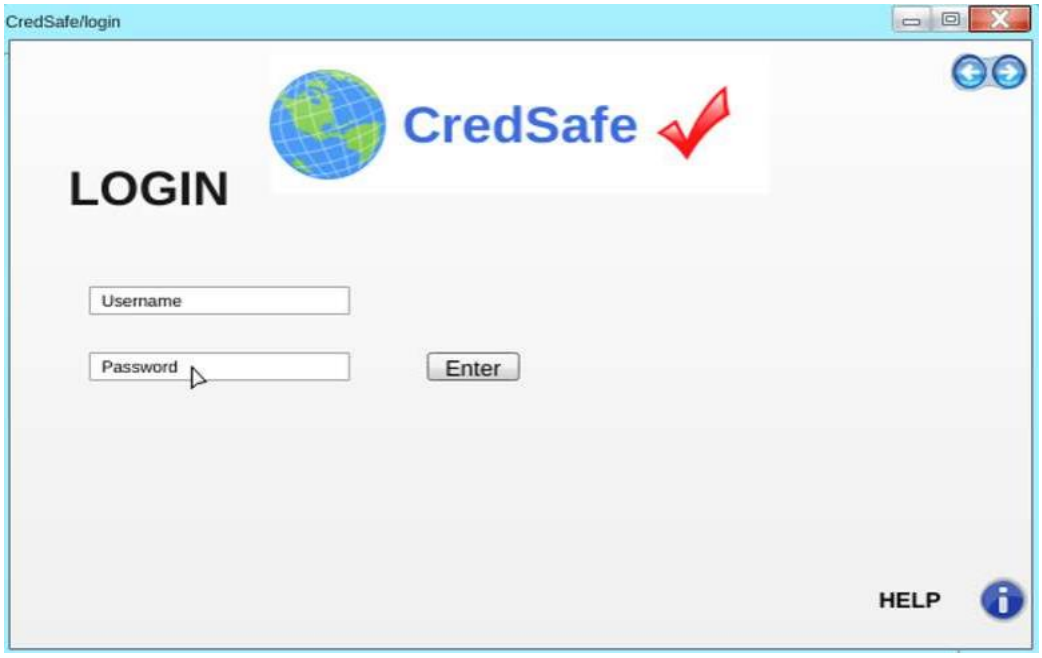
Analysts at the Security Operations Center (SOC): People who look into any security lapses and major fraud attempts.

Business analysts are users who look for patterns and trends in fraud to find places where the system needs to be improved.

Cardholders: The system eventually guards against fraudulent activity even though they don't interact with the UI directly. Future features for cardholder self-service choices (such as transaction review and reporting suspicious activity) should be taken into account in the design.

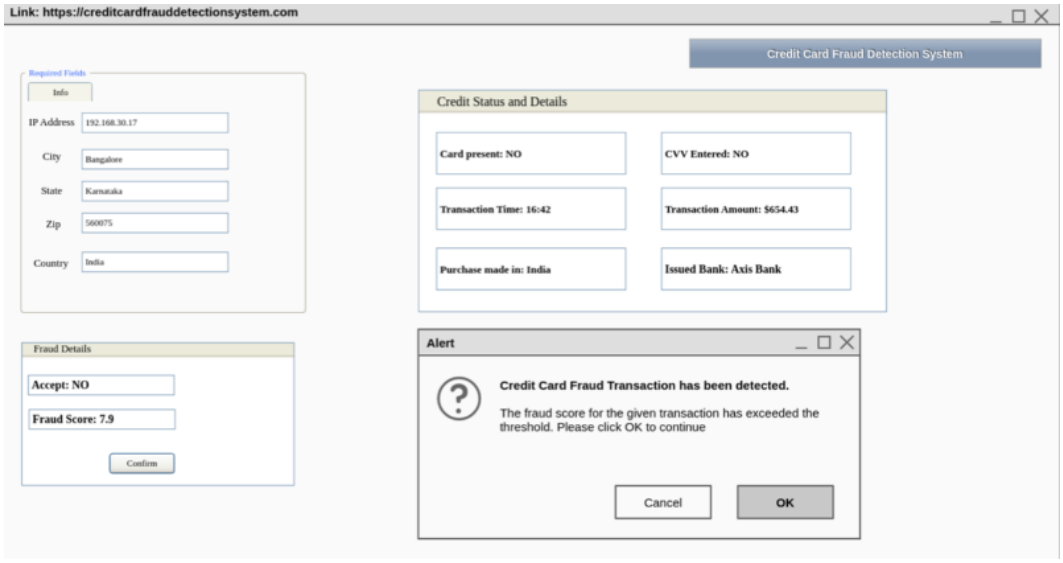
7.2 UI Screens Overview

Login Screen: State changes from input mode to validation mode. Upon validation success, moves to the Dashboard. Shows an error message if the event fails.



Home Screen: Changes content dynamically in response to real-time data; user selection controls screen transitions.

Fraud Detection Module: Based on user actions (approve/reject), transaction status changes that may have an impact on the fraud detection model.

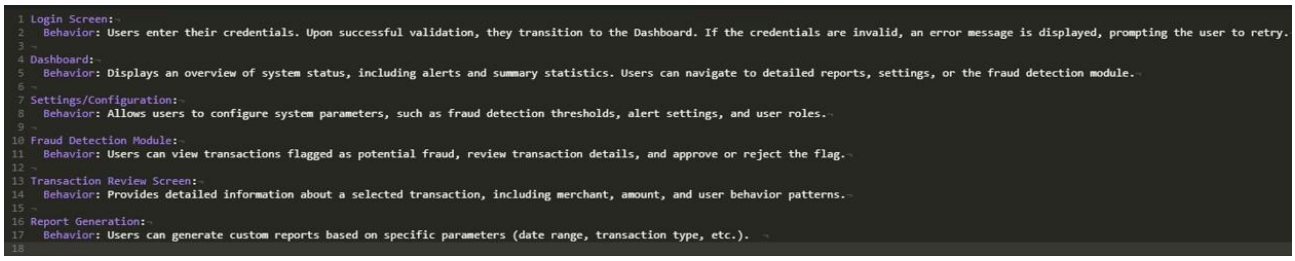


About Section: General details regarding the making and the developers behind it.

Transaction Review Screen: Based on user actions (approve/reject), the transaction status is updated to reflect the current review status.

Report Generation Screen: The parameter selection screen gives way to the report generation/viewing screen.

Settings/Configuration screen: saves changes made to the system configuration parameters when a user interacts with it.



7.2.1 Workflow Reports

Users can create customized reports offering insights on flagged transactions, investigation results, fraud patterns, and the efficacy of fraud detection algorithms using the Credit Card Fraud Detection System's Workflow Reports feature. To customize reports to their own requirements, users can set characteristics like the date range, transaction type (online, in-store), and status (approved, denied, pending). In order to provide thorough analysis and well-informed decision-making in fraud management, the system provides historical data analysis, export functionality in many formats (such as PDF and Excel), and automated report scheduling for regular delivery to stakeholders.

The table is the expected workflow reports provided with its functionality:

Label Name	Note	Source
Fraud Case ID	Unique identifier for each suspected fraud case.	System-generated
Transaction ID	Reference ID for the transaction under investigation.	Transaction Data
Fraud Status	Current status of the fraud investigation (e.g., Pending, Confirmed Fraud, False Positive).	Investigation Outcome
Investigation Notes	Notes added by the investigator regarding the fraud case.	Investigation Data
Fraud Category	Classification of the fraud type based on the investigation (e.g., Identity Theft, Skimming).	Investigation Outcome
Detection Date	The date and time when the fraudulent activity was detected by the system.	System-generated

Preventive Action Taken	Description of actions taken to prevent similar fraud occurrences in the future.	Post-Investigation Actions

Appendix A: Project Timeline

Phase 1: Planning and Requirements Gathering
Project Kickoff Meeting
Completion of Requirements Gathering

Phase 2: System Design
Completion of High-Level Design
Completion of Detailed Design Specifications

Phase 3: Development
Development Start
Initial Prototype Completed
Internal Testing and Iterations

Phase 4: Testing
Start of System Testing
Start of User Acceptance Testing (UAT)
Completion of All Testing

Phase 5: Deployment
System Deployment Preparation
Go-Live and System Rollout

Phase 6: Maintenance and Review
Post-Deployment Review
Implementation of Initial Maintenance Updates

Phase 7: Project Closure
Final Project Review and Closure