Problem Statement: Data Analysis software for financial frauds

Design and implement advanced Financial Data Analysis Software to analyze and process financial data from various sources, including transaction records, trading activities, and customer behavior. The goal is to identify potentially fraudulent activities.

Solution:

We have created a software system for real-time transaction monitoring, using advanced machine learning and pattern recognition. This system continuously watches financial transactions as they happen, helping law enforcement spot potential fraud.
Following are some key features of software :

1) Pattern Recognition for Fraud Detection:

- Implemented pattern recognition to find specific signs of fake accounts and fraudulent transactions.
- Used a mix of transaction frequency, amounts, location info, and user behavior to build a strong fraud detection system

2) Machine Learning Integration:

- Trained the algorithms with historical data, so the system quickly spots suspicious activities.

3) Alert System for Quick Response:

- set up an alert system that triggers in real time when potential fraud is detected.
- Alerts are instant, helping law enforcement react quickly and investigate suspicious activities

.

4) Centralized SQL Database for Fraud Accounts:

- Set up a centralized SQL database to store and manage info on confirmed fraud accounts.
- Implemented efficient search features for quick retrieval of relevant data during real-time monitoring.

5) Web Extension for Customer Notifications:

- Made a customer-friendly web extension for real-time notifications about potential fraudulent activities.
- Included important details like account holder name, address, and transaction specifics for customer awareness.

6) Transparent Visualization for Analysts:

- Designed a clear and user-friendly interface for law enforcement analysts to monitor transactions in real time.
- Showed flagged transactions and potential fraud patterns for easy analysis

7) Dynamic Updating for New Fraud Trends:

- Created a dynamic updating system that keeps the machine learning models up-to-date with new fraud trends.
- Regularly fed the system with fresh data to stay effective in spotting emerging patterns linked to fake accounts and fraud.

Tech stack

Programming languages used are java , c#, python, javascript.
Various libraries of python used are numpy,pandas,scikitlearn,tensorflow,keras.

Backend Framework:

Django (Python): Django is a high-level Python web framework that promotes rapid development and clean, pragmatic design. It comes with built-in security features and is well-suited for scalable applications.
Database:

PostgreSQL: A powerful, open-source relational database management system. It offers advanced features like full-text search and supports spatial data, which can be beneficial for a software application dealing with law enforcement data.

Frontend Framework:

React.js (JavaScript): A widely-used JavaScript library for building user interfaces. It's maintained by Facebook and a community of individual developers, providing a smooth and responsive user experience.

Mobile App Development:

React Native (JavaScript): If there is a need for a mobile application, React Native allows for cross-platform development, enabling the use of a single codebase for both iOS and Android.

Authentication:

OAuth 2.0: Implement OAuth 2.0 for secure and standardized authentication. This is crucial for maintaining user security, especially in a system that involves law enforcement data.

Containerization and Orchestration:

Docker: For containerization, enabling the packaging of the application and its dependencies into a standardized unit for seamless deployment.

Kubernetes: For orchestrating and managing containerized applications, ensuring scalability, and enhancing resilience.

Message Brokers:

Security:

SSL/TLS for Encryption: To ensure secure communication over the network.

Security Headers: Implement HTTP security headers to enhance the security of web applications.

Machine Learning:

PyTorch: An alternative deep learning library to TensorFlow, depending on the specific requirements of the machine learning components.
Numpy
Pandas
Tensorflow
Scikit Learn

Jenkins or GitLab CI/CD: For continuous integration and continuous deployment to streamline the development and release process.

As we have made this software for both the police use as well as the personal use hence for the extension version the tech stack which will be used is

HTML/CSS/JavaScript: for the front of the web extension

Browser Extension APIs:

- chrome.runtime: For handling runtime events.
- chrome.storage: For storing and retrieving data.
- chrome.tabs: For managing browser tabs

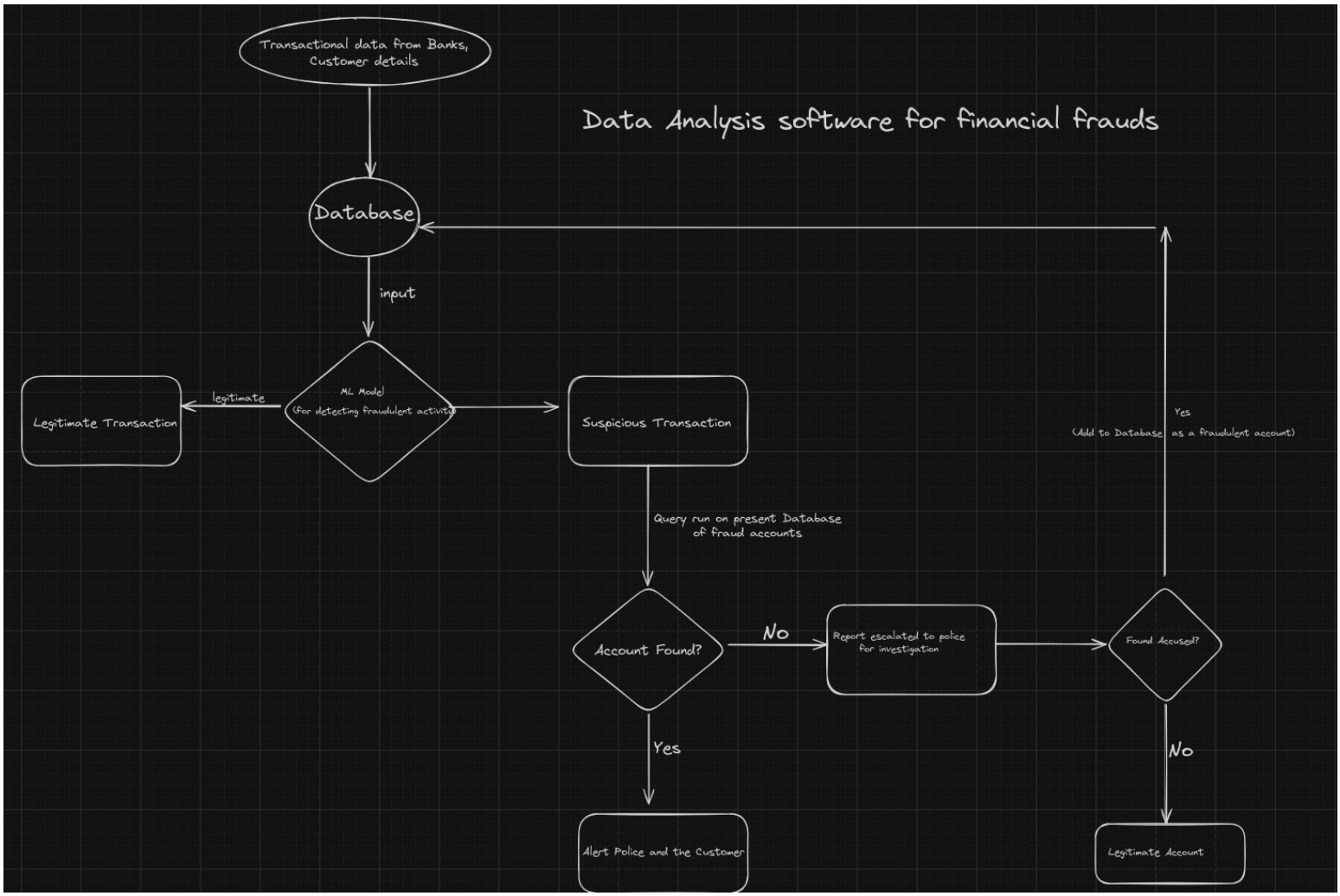
Back-End (Server-Side):

- Node.js
- Express.js
- Database (SQL Server)
- ORM (Object-Relational Mapping)

Members and Responsibility:

- Md Ashhar (Frontend Developer and UI/UX Design)
- Deepank Singh (DevOps and ML)
- Abhirajkar Bajpai (Full Stack Developer)
- Puneet Prashar(Data Analytics and ML)

Flowchart:



Schedule:

14 Dec, 2023: Task : Establish Specific Project Requirements Compile a thorough list of the features, performance standards, and security requirements for the fraud detection system. as well as the division of responsibility among participants

24 Dec, 2023: Task : The prototype has been finished with appropriate UI and UX design.

30th December,2023 : data gathering, data rearrangement, and training and testing of the ML model

4th january,2024:last iteration of the working prototype. Conduct appropriate testing during front-end and back-end integration