System and Project Requirements Specification (SRS)

Project Name: Aetheris.network

Version: 1.0

Date: October 2024

Prepared by: Abhishek Shirsath

---

1. Introduction

1.1 Purpose

The purpose of this SRS document is to define the requirements and specifications for the development of **Aetheris**, a centralized, AI-driven data management platform aimed at providing actionable insights for industries such as healthcare and finance. The MVP (Minimum Viable Product) will focus on delivering a robust centralized AI analytics system while maintaining data security and user accessibility.

1.2 Scope

The scope of the Aetheris MVP includes:

- A centralized AI engine that analyzes data and provides industry-specific insights.
- A user-friendly dashboard for clients to upload data, interact with AI analytics, and view performance metrics.
- Secure data handling with encryption and compliance with industry standards.

1.3 Definitions, Acronyms, and Abbreviations

- **MVP**: Minimum Viable Product
- **AI**: Artificial Intelligence
- **API**: Application Programming Interface
- **SaaS**: Software as a Service
- **CI/CD**: Continuous Integration/Continuous Deployment
- **HIPAA**: Health Insurance Portability and Accountability Act
- **PCI DSS**: Payment Card Industry Data Security Standard

1.4 Overview

This document will outline the functional and non-functional requirements, system architecture, technology stack, and features of Aetheris. It aims to provide a clear understanding of the project to facilitate successful development and implementation.

2. Overall Description

2.1 Product Perspective

Aetheris is envisioned as a standalone platform designed for data management and analytics. It will integrate with various data sources to deliver actionable insights using AI. The platform aims to enhance operational efficiency and decision-making for its users in different industries.

2.2 Product Functions

Key functionalities of the Aetheris MVP include:

- **User Authentication**: Secure login system with OAuth integration.
- **Data Upload**: Users can upload datasets for analysis.
- **AI Analytics**: Centralized AI engine that processes uploaded data and provides insights.
- **User Dashboard**: A responsive interface that allows users to visualize analytics and performance metrics.
- **Data Security**: End-to-end encryption and compliance with industry standards.

2.3 User Classes and Characteristics

- **End Users**:
    - **Healthcare Professionals**: Require insights to improve patient outcomes.
    - **Financial Analysts**: Need data analysis for transaction optimization.
- **Administrators**: Manage the platform, oversee user accounts, and ensure data security.

2.4 Operating Environment

The Aetheris platform will operate in a cloud-based environment utilizing services such as AWS, Azure, or GCP. It will be accessible via modern web browsers on desktop and mobile devices.

2.5 Design and Implementation Constraints

- Compliance with **HIPAA** and **PCI DSS** for healthcare and financial data handling.
- User data must be encrypted both at rest and in transit.
- Scalability to accommodate increasing user loads and data volume.

---

3. System Requirements

3.1 Functional Requirements

3.1.1 User Authentication

- The system must provide secure user authentication using OAuth with options for Google and Facebook integration.

3.1.2 Data Upload and Management

- Users must be able to upload datasets in common formats (e.g., CSV, JSON).
- The platform should support batch uploads and provide validation feedback for data integrity.

3.1.3 AI Analytics Engine

- The AI engine must process uploaded data and deliver insights based on predefined algorithms tailored for healthcare and finance.
- Users should be able to request specific analyses (e.g., trend analysis, anomaly detection).

3.1.4 User Dashboard

- A responsive dashboard should display analytics in a user-friendly manner, including graphs, charts, and tables.
- Users should have options to filter and customize their data views.

3.1.5 Data Security and Compliance

- All user data must be encrypted using **AES-256 encryption**.
- The system should comply with relevant industry regulations (HIPAA for healthcare, PCI DSS for finance).

3.2 Non-Functional Requirements

3.2.1 Performance

- The system must support a minimum of **10,000 concurrent users** with a response time of under **200 milliseconds** for analytics queries.

3.2.2 Reliability and Availability

- The platform should ensure **99.9% uptime**, with redundancy and failover mechanisms in place.

3.2.3 Scalability

- The architecture must allow for scaling resources based on user demand and data volume, utilizing cloud services for elasticity.

3.2.4 Maintainability

- Code should be modular and adhere to best practices for maintainability, enabling easier updates and feature additions in the future.

3.2.5 Usability

- The user interface must be intuitive, requiring minimal training for end-users to effectively utilize the platform.

---

4. System Architecture

4.1 High-Level Architecture

The architecture will consist of:

- **Frontend**: Developed using React or Angular, providing a user-friendly interface for interaction.
- **Backend**: Node.js or Python-based services to handle API requests, process data, and interact with the AI engine.
- **AI Engine**: Built with TensorFlow or PyTorch to deliver analytics based on uploaded data.
- **Database**: PostgreSQL or MongoDB for data storage, providing efficient querying and retrieval.

4.2 API Layer

- RESTful API to connect the frontend and backend, allowing seamless data exchange and functionality access.

---

5. Technology Stack

5.1 Frontend

- **Languages**: JavaScript, TypeScript
- **Frameworks**: React, Angular, Mern
- **Libraries**: MUI for UI components

Purpose: Provide a user interface for clients to interact with their AI nodes and view real-time analytics.

5.2 Backend

- **Languages**: Node.js, Python
- **Frameworks**: Express.js, Flask, Django
- **Database**: PostgreSQL or MongoDB

Purpose: Process AI tasks, manage data flows, and handle the communication between frontend and AI nodes.

5.3 AI & Machine Learning

- **Libraries**: TensorFlow, PyTorch, Scikit-learn

Purpose: Perform local AI computations at the node level, feeding into the decentralized Aetheris AI network for global insights.

5.4 Cloud Infrastructure

- **Services**: AWS, Azure, or GCP for cloud hosting and data storage.

Purpose: Ensure secure and scalable storage and computing for AI nodes and backend processing.

6. System Features

6.1 User Authentication

- Secure user authentication methods integrated with OAuth.

6.2 Data Analytics Dashboard

- User-friendly dashboard displaying key insights, analytics, and data visualizations.

6.3 AI Insights Generation

- Centralized AI engine providing actionable insights based on uploaded datasets.

---

7. Conclusion

The Aetheris MVP aims to deliver a robust, user-friendly platform that leverages AI for industry-specific analytics. By focusing on core functionalities and data security, Aetheris will position itself as a competitive player in the data management and analytics market. This SRS document serves as a foundation for the development team to understand the requirements and objectives for the successful delivery of the MVP.