

Software Requirement Specification
AI-Driven Optimization of Local Body
and MP Fund Management

Prepared by

Paul Binu (MAC22CD048)

Paulu Wilson (MAC22CD049)

Ronal Shoey George (MAC22CD051)

February 13, 2025

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Document Conventions	3
1.3	Intended Audience and Reading Suggestions	3
1.4	Product Scope	3
1.5	References	4
2	Overall Description	5
2.1	Product Perspective	5
2.2	Product Functions	5
2.3	User Classes and Characteristics	5
2.4	Operating Environment	5
2.5	Design and Implementation Constraints	5
2.6	User Documentation	6
2.7	Assumptions and Dependencies	6
3	External Interface Requirements	7
3.1	User Interfaces	7
3.2	Hardware Interfaces	7
3.3	Software Interfaces	7
3.4	Communication Interfaces	7
4	System Features	8
4.1	Fund Allocation Optimization	8
4.2	Project Prioritization	8
5	Other Nonfunctional Requirements	9
5.1	Performance Requirements	9
5.2	Security Requirements	9
5.3	Software Quality Attributes	9
6	Appendices	10
6.1	Appendix A: Glossary	10
6.2	Appendix B: Analysis Models	10

1 Introduction

1.1 Purpose

This document provides a detailed description of the AI-driven web application for optimizing Local Body and MP fund management. It outlines the purpose, key functionalities, and system requirements to ensure equitable fund allocation, real-time monitoring, and enhanced transparency using AI/ML.

1.2 Document Conventions

The document follows these formatting conventions:

- Headings: **Bold**
- Sub-Headings: *Italicized*
- Normal Text: Regular

1.3 Intended Audience and Reading Suggestions

This SRS is intended for:

- Developers and AI engineers
- Government policy makers
- Project stakeholders and evaluators
- End-users (local government officials, auditors)

1.4 Product Scope

The system provides AI-powered analytics for fund allocation, project prioritization, fraud detection, real-time monitoring, and public engagement. It supports:

- **AI-driven fund allocation** based on socio-economic data.
- **Project prioritization** using NLP-based sentiment analysis of community feedback.
- **Fraud detection** through anomaly detection and blockchain-based tracking.

- **Real-time project tracking** via IoT and satellite imagery.
- **Public engagement** using chatbots and AI-driven recommendation systems.

1.5 References

- Government fund management guidelines
- Research papers on AI in governance
- Best practices in financial technology and data analytics

2 Overall Description

2.1 Product Perspective

The system is a web application that integrates machine learning, NLP, and real-time monitoring to optimize fund utilization and governance transparency.

2.2 Product Functions

- Fund allocation optimization
- Project prioritization using AI/ML
- Real-time fraud detection and compliance monitoring
- Public sentiment analysis and engagement
- Automated impact assessment

2.3 User Classes and Characteristics

- **Government Officials** – Manage fund allocation and project approvals.
- **Auditors & Analysts** – Monitor fund usage and detect anomalies.
- **General Public** – Provide feedback on projects.
- **Developers & Data Scientists** – Maintain AI models and infrastructure.

2.4 Operating Environment

- Web-based system accessible via modern browsers.
- Cloud-based deployment with AI-driven backend.
- Secure data storage and processing.

2.5 Design and Implementation Constraints

- Compliance with government data security policies.
- Integration with existing fund management systems.
- Scalability to support nationwide implementation.

2.6 User Documentation

- Online help center.
- User manuals for officials and stakeholders.
- API documentation for developers.

2.7 Assumptions and Dependencies

- Reliable internet connectivity.
- Availability of structured government fund allocation data.
- Support from government bodies for implementation.

3 External Interface Requirements

3.1 User Interfaces

- Web dashboard for fund allocation and project monitoring.
- Chatbot interface for public engagement.
- Visual analytics for fraud detection and compliance.

3.2 Hardware Interfaces

- Server infrastructure to support AI processing.
- IoT and satellite-based tracking for project monitoring.

3.3 Software Interfaces

- Backend: Python (Django/Flask)
- Frontend: React.js / HTML, CSS
- Database: PostgreSQL / MongoDB
- AI Models: TensorFlow / PyTorch
- NLP: Hugging Face Transformers
- Blockchain: Hyperledger Fabric

3.4 Communication Interfaces

- Secure HTTPS protocol for data transfer.
- API integrations with government databases.

4 System Features

4.1 Fund Allocation Optimization

- AI-driven predictions for fund distribution.
- Clustering algorithms for region-based prioritization.
- Optimization models for impact maximization.

4.2 Project Prioritization

- NLP-based analysis of community feedback.
- Multi-criteria decision-making models for project ranking.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

- Quick response time for AI predictions.
- Scalable cloud-based architecture.

5.2 Security Requirements

- End-to-end encryption for sensitive data.
- Role-based access control for officials and auditors.

5.3 Software Quality Attributes

- **Reliability** – Robust AI models for consistent predictions.
- **Maintainability** – Modular architecture for easy updates.
- **Portability** – Compatible across various devices and browsers.
- **Scalability** – Designed to handle large-scale government data.

6 Appendices

6.1 Appendix A: Glossary

- **Drug Repurposing:** Finding new uses for existing drugs.
- **SRS:** Software Requirement Specification.
- **HDGAT:** Heterogeneous Drug-Disease Graph Attention Network.

6.2 Appendix B: Analysis Models

- Flow Diagram.

