Software Requirements Specifications

For

Predictive Maintenance for Industrial Machinery

Prepared by

Group 2

Members:

202101040041 Tanavi Prakash Gaikwad

202101040039 Pritam Rajendra Gajare

202101040135 Nandini Chandrakant Gangurde

202101040137 Gauri Vinod Kardekar

202101040262 Manish Jitendra Meshram

202101070082 Mrinal Gupta

Table of Contents

Table of Contents

Revision History ii

1. Introduction

1.1 Purpose 3
1.2 Document Conventions 3
1.3 Intended Audience and Reading Suggestions 3
1.4 Product Scope 4
1.5 References 4

2. Overall Description

2.1 Product Perspective 4
2.2 Product Functions 5
2.3 User Classes and Characteristics 6
2.4 Operating Environment 7
2.6 User Documentation 7
2.7 Assumptions and Dependencies 8

3. External Interface Requirements

3.1 User Interfaces 8
3.2 Hardware Interfaces 8
3.3 Software Interfaces 8
3.4 Communications Interfaces 9

4. System Features

4.1 System Feature 1 9
4.2 System Feature 2 10
4.3 System Feature 3 10

5. Other Nonfunctional Requirements

5.1 Performance Requirements 10
5.2 Safety Requirements 10
5.3 Software Quality Attributes 10
5.4 Business Rules 10

Appendix A: UML Diagrams 12

Appendix B: Acronyms 16

01. Introduction

SRS or Software Requirements Specification is an important document in the software development process. It outlines a detailed description of what a software system must do. The primary purpose of an SRS is to provide a clear and comprehensive understanding of the software requirements to all stakeholders involved in the development process, including developers, testers, project managers, and clients. There are two types of requirements in SRS Functional and Nonfunctional.

Functional requirements: Functional requirements describe the specific features and capabilities the software must have to meet its users' needs. These requirements outline what the system should do and how it should behave under various circumstances.

Nonfunctional requirements: Nonfunctional requirements define the qualities or attributes of the software system, focusing on aspects other than specific behaviors. These requirements often address performance, usability, reliability, and other characteristics.

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to provide a comprehensive understanding of the requirements for the development of an intelligent system using machine learning algorithms to predict equipment failures and schedule model aims to minimize downtime, optimize maintenance costs and prolong the lifespan of critical machinery.

1.2 Document Conventions

The following conventions are used throughout this document:

- Headings and subheadings: Bold
- Section numbering: Decimal format (e.g., 1, 1.1, 1.2)
- Lists: Bulleted or numbered
- Emphasis: Italics
- Document references: In square brackets (e.g., [1])

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- Project stakeholders
- Development and implementation teams
- Quality assurance personnel
- Decisionmakers involved in the project

Reading Suggestions:

- Stakeholders are advised to review the entire document for a comprehensive understanding of the project.
- Development teams should focus on sections related to technical specifications and requirements.
- Quality assurance personnel should pay attention to testing and validation sections.

1.4 Product Scope

- 1. Objective: Develop a Predictive Maintenance System for Industrial Machinery.

2. Machine Learning Integration: Utilize machine learning algorithms to analyze:

- Historical data
- Sensor readings
- Operational parameters

3. Predictive Capability: Predict equipment failures and schedule maintenance proactively.

4. Anomaly Detection: Identify anomalies indicative of impending malfunctions.

5. Integration with CNC Machine: Integrate with a CNC machine to enhance predictive capabilities and coordinate maintenance actions with production processes.

6. Benefits:

- Minimize downtime
- Optimize maintenance costs
- Extend machinery lifespan

7. User Interface: Develop a userfriendly website interface to:

- Monitor equipment health
- View predictive insights
- Schedule maintenance tasks.

1.5 References

- https://www.slideshare.net/khushikalaria/srsforrailwayreservationsystem
- https://www.geeksforgeeks.org/softwarerequirementspecificationsrsformat/
- https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8660386

- Asif, M., Ali, I., Malik, M. S. A., Chaudary, M. H., Tayyaba, S., & Mahmood, M. T. (2019). Annotation of software requirements specification (srs), extractions of nonfunctional requirements, and measurement of their tradeoff. *IEEE Access*, *7*, 3616436176.
- https://www.rst.software/blog/howtowritesrsdocument

2.Overall Description

2.1 Product Perspective

The Predictive Maintenance System for Industrial Machinery is a designed to analyze historical data, sensor readings, and operational parameters from industrial machinery. The system interacts with existing data storage systems to access historical data. The system will be scalable to handle large volumes of historical data and real-time sensor readings efficiently. Machine learning models will be designed and optimized for performance to ensure timely predictions without significant computational overhead. Model performance will be monitored to maintain accuracy. Visualization tools and feature engineering would be used to understand data better and make it user-friendly.

2.2 Product Functions

1. Data Collection & Integration:
   - Collect and integrate historical machinery data, including sensor readings.
   - Preprocess data for analysis.

2. Feature Engineering:
   - Extract relevant features from the dataset for analysis.

3. Machine Learning Model Development:
   - Develop ML algorithms to predict equipment failures.
   - Use classification, and anomaly detection.

4. Realtime Monitoring:
   - Continuously monitor sensor data.
   - Provide alerts for potential failures.

5. Maintenance Scheduling:
   - Schedule maintenance based on predictions.
   - Optimize schedules to minimize downtime.

6. User Interface Design:
   - Create an intuitive web interface.

- Include interactive visualizations.

7. Security & Data Privacy:
    - Ensure data security and user privacy.
    - Implement authentication mechanisms.
8. Scalability & Performance:
    - Build for handling large data volumes.
    - Optimize for timely responses.

9. Documentation & Training:
    - Provide comprehensive documentation.
    - Offer user training resources.

10. Integration with CNC Machines:
    - Develop interfaces for CNC integration.
    - Facilitate data exchange for proactive maintenance.

2.3 User Classes and Characteristics

The system is designed to accommodate the following user classes:

Primary actors

1. **Administrator:** The Administrator oversees the deployment, configuration, and maintenance of the Predictive Maintenance System for CNC machines. They are responsible for managing the system's infrastructure, ensuring its availability, and maintaining data security.

2. **Quality Assurance Personnel:** Quality assurance personnel are responsible for ensuring that CNC machining operations meet quality standards and specifications. They utilize the Predictive Maintenance System to monitor machine performance, detect anomalies, and optimize production processes to maintain product quality.

3. **Data Analyst:** Data analysts play a critical role in analyzing CNC machine data, validating predictive models, and generating insights to improve maintenance strategies and system accuracy. They leverage their expertise in data analysis and statistical techniques to refine the model and view specifications.

Secondary Actors:

1. **Sensors:** Sensors installed on CNC machines collect data related to temperature, speed, torque, tool wear, and other relevant parameters. This data is transmitted to the Predictive Maintenance System for analysis and monitoring.

2. **External data resources:** External data sources, such as weather data or supply chain information, may impact CNC machine operations. For example, ambient temperature fluctuations could affect machine performance, or variations in raw material quality could influence tool wear rates. Integrating such external data sources can enhance the system's predictive capabilities by providing additional context for analyzing machine behavior.

2.4 Operating Environment

The system is intended to operate in a standard web environment, compatible with modern web browsers. The recommended operating environment includes:

- **Python Environment:** Python 3.x is used as the primary programming language.
- **Google Colab:** Development and execution environment for running Python code. Provides free access to GPU/TPU resources, Jupyter Notebook integration, and cloudbased storage.
- **Python Libraries:** Libraries like Pandas, NumPy, Scikitlearn, TensorFlow, Keras, Matplotlib, Seaborn, Streamlit.
- **Machine Learning Algorithms:** Utilized for predictive modeling and anomaly detection based on sensor data. Examples are Random Forest, Support Vector Machine (SVM), Gradient Boosting

2.5 Design and Implementation Constraints

Design and Implementation Constraints for SRS:

1. Scalability: Ensure the system can handle large data volumes efficiently.
2. Realtime Processing: Support realtime analysis for timely predictions.
3. CNC Compatibility: Integrate with CNC machines for seamless data exchange.
4. ML Algorithms: Specify algorithms and consider resource constraints.
5. Interoperability: Ensure compatibility with existing systems.
6. UI Design: Prioritize usability and compatibility.
7. Regulatory Compliance: Adhere to industry standards.
8. Data Quality: Implement processes for data accuracy.
9. Maintenance: Define support and update constraints.
10. Cost: Consider budgetary limitations.
11. Training: Provide user training and documentation.

2.6 User Documentation

Comprehensive user documentation for the Predictive Maintenance System for CNC Machines would encompass a range of resources tailored to meet the needs of different user roles and proficiency levels. It would include detailed user manuals providing stepbystep instructions on system usage, feature descriptions, and troubleshooting guidelines. Additionally, interactive tutorials and walkthroughs would be available to help users navigate the system interface and perform common tasks efficiently.

2.7 Assumptions and Dependencies

Assumptions:

1. Data Availability: Historical data on machinery performance, sensor readings, and operational parameters are accessible and sufficient for model training.
2. Data Quality: Assumption of accurate, consistent, and reliable historical data for unbiased model training.
3. Parameter Relevance: The parameters listed in the sample dataset are assumed to be relevant to machinery health.
4. Sensor Readings: Dependence on consistent sensor readings reflecting the true machinery state.
5. Operational Stability: Stability assumption in the operational environment during predictive maintenance processes.

Dependencies:

1. Sensor Data Access: Dependence on realtime sensor data integration from machinery.
2. Data Preprocessing: Essential preprocessing steps for cleaning, transforming, and normalizing raw sensor data.
3. Machine Learning Algorithms: Selection and implementation of appropriate algorithms for predictive maintenance.
4. Model Training and Evaluation: Dependence on historical data for model training and evaluation.
5. Website Integration: Integration of predictive maintenance functionality into the website interface.
6. Maintenance Scheduling: Integration with maintenance scheduling systems for proactive planning based on predictions.

3. External Interface Requirements

3.1 User Interfaces

- The system provides a userfriendly webbased interface built with Streamlit, accessible within the Google Colab environment. This interface allows users to interact with the system's features and functionalities directly within the Colab notebook interface.
- Users can access different modules, visualize machine performance metrics, schedule maintenance tasks, and view predictive analytics insights seamlessly.

3.2 Hardware Interfaces

- The system interfaces with sensors installed on CNC machines to collect realtime data on machine parameters. While the user interacts with the system through the Streamlit web application interface, sensor data is collected from CNC machines and transmitted to the Colab environment for processing and analysis.

3.3 Software Interfaces

- Python Environment: Python 3.x serves as the primary programming language, offering flexibility and robustness in development.

- Google Colab: Utilized as the development and execution environment, Google Colab facilitates the execution of Python code. It grants free access to GPU/TPU resources, integrates seamlessly with Jupyter Notebooks, and provides cloudbased storage for enhanced collaboration and accessibility.

- Python Libraries: The system leverages essential Python libraries such as Pandas, NumPy, Scikitlearn, TensorFlow, Keras, Matplotlib, Seaborn, and Streamlit. These libraries offer a wide range of functionalities for data manipulation, machine learning, visualization, and web application development.

- Machine Learning Algorithms: The system employs various machine learning algorithms for predictive modeling and anomaly detection based on sensor data. Examples include Random Forest, Support Vector Machine (SVM), and Gradient Boosting, which enable accurate predictions and early anomaly detection to optimize maintenance procedures.

3.4 Communications Interfaces

- Users interact with the Predictive Maintenance System for CNC Machines through Google Colab notebooks, leveraging the Streamlit web application hosted within this environment. Accessible via a standard web browser
- Facilitated through HTTP or HTTPS protocols, communication between the user's web browser and the Google Colab server ensures secure and efficient data exchange, enabling users to monitor CNC machine performance, schedule maintenance tasks, and access predictive analytics insights with ease.

4. System Features

1. **Real-time Monitoring:**

- **Description:** Continuously monitor CNC machine parameters such as temperature, speed, torque, and tool wear in real-time.
- **User Roles:** Administrator
- **Dependencies:** Integration with sensors installed on CNC machines to collect real-time data.

2. **Predictive Analytics:**

- **Description:** Analyze historical data and sensor readings to predict potential equipment failures and maintenance needs.
- **User Roles:** Data Analysts
- **Dependencies:** Machine learning algorithms trained on historical data and sensor readings.

3. **Data Analysis Tools:**
    - **Description:** Utilize data analysis tools and statistical techniques to analyze CNC machine data, identify trends, and detect anomalies.
    - **User Roles:** Quality Assurance Personnel.
    - **Dependencies:** Availability of historical data and sensor readings for analysis.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Real-time Processing: Analyze data instantaneously to deliver immediate alerts on abnormal behaviors.
- System Maintenance: Regularly update and maintain software and hardware to ensure consistent performance.
- Scalability: Adapt to accommodate data from an increasing number of machines as operations expand.
- Data Integrity: Ensure high accuracy and integrity of data for reliable predictions and decision-making.

5.2 Safety Requirements

- Accurate Failure Prediction: Ensure reliable predictions of machine failures to enhance safety and prevent accidents.

5.3 Software Quality Attributes

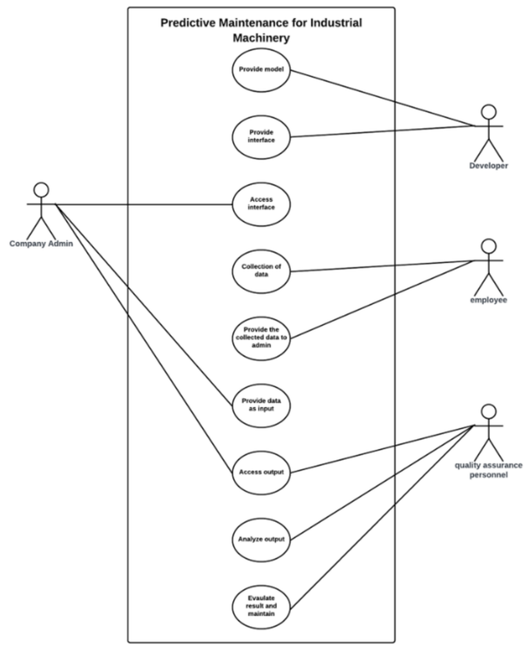
- Reliability: The system needs to function consistently and accurately over a designated timeframe without failure.
- Scalability: The software must adjust its capacity based on the growth or reduction in CNC machine count and data volume.

5.4 Business Rules

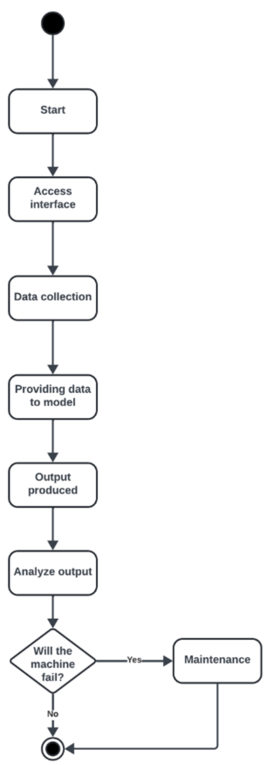
- Data Collection Frequency: Collect sensor data from CNC machines at every specified interval (X seconds/minutes) to maintain timely and accurate predictive analysis.
- Data Retention: Retain all machine data and maintenance logs for a minimum period (Y years) to meet compliance requirements and facilitate thorough analysis.
- Data Accuracy: Ensure data used for predictions maintains a minimum accuracy level (Z%), and report any deviations from this standard.

APPENDIX A: UML DIAGRAMS

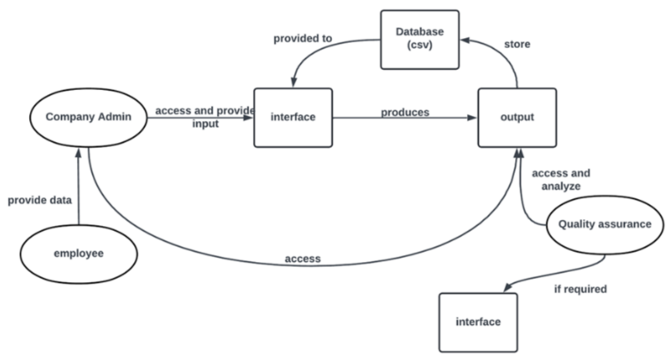
Use Cases



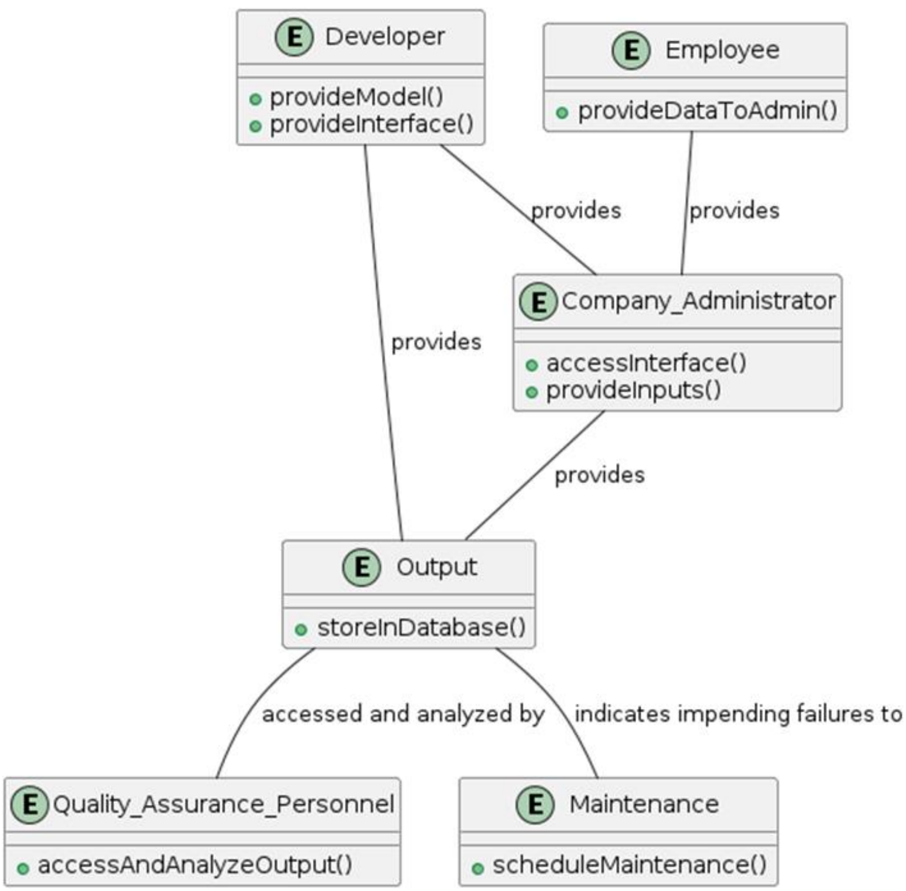
Activity Diagram



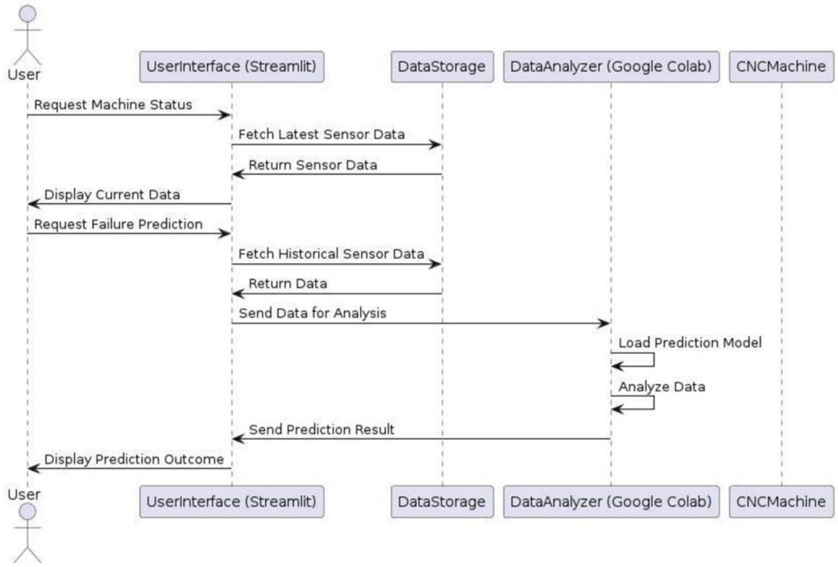
DFD



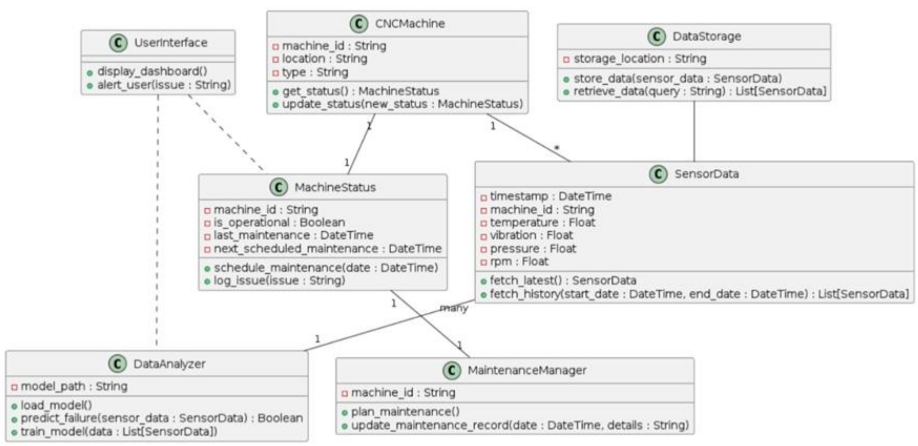
ER Diagram



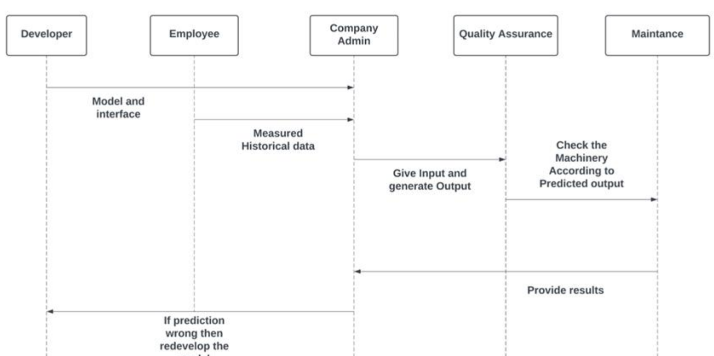
Interaction Diagram



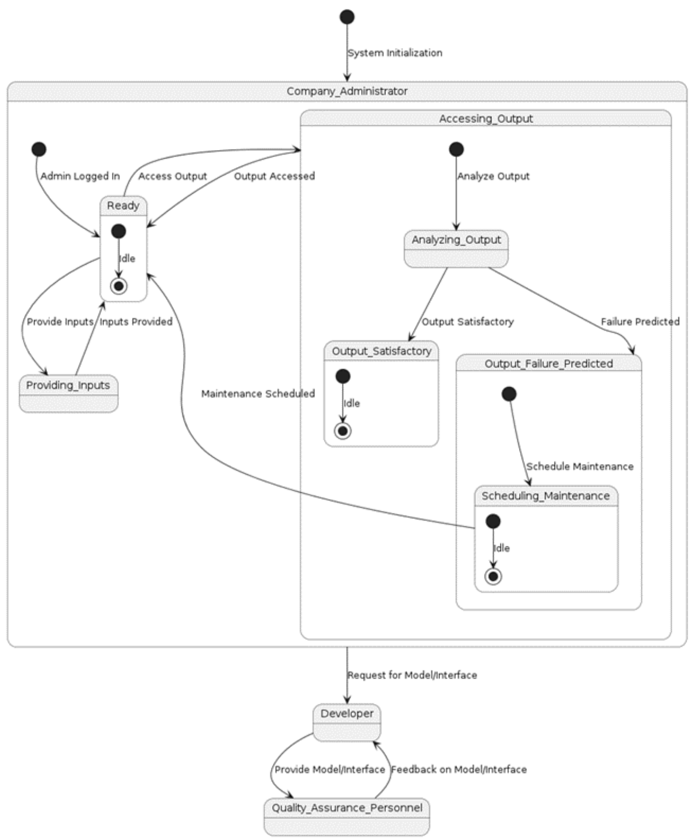
Class Diagram



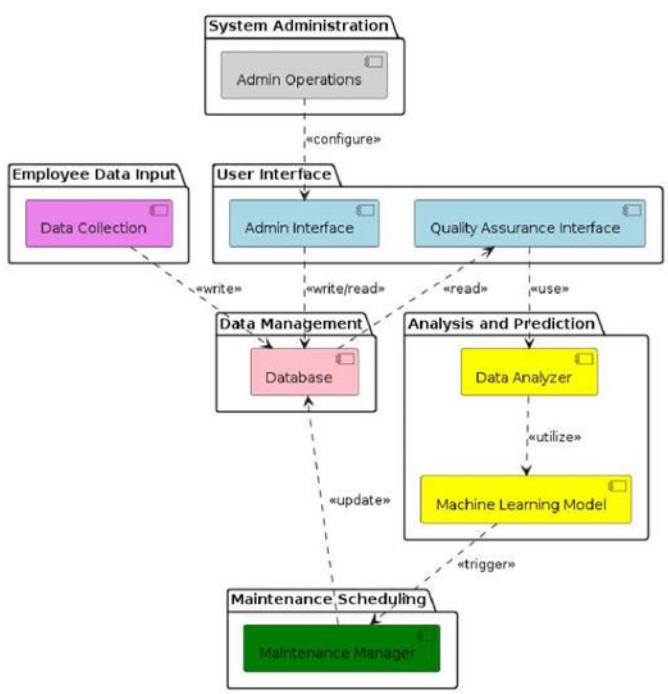
Sequential Diagram



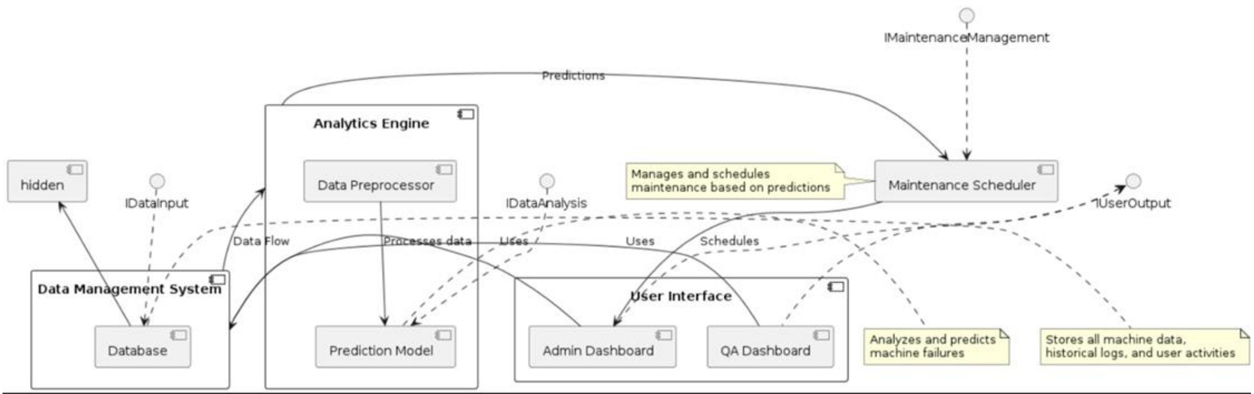
Case Diagram



Package Diagram



Component Diagram



APPENDIX B: ACRONYMS

Acronyms	Explanation
SRS	Software Requirements Specification
HTML	Hypertext Markup Language
HTTPS	Hypertext Transfer Protocol Secure
API	Application Programming Interface
DFD	Data Flow Diagram
ER	Entity-Relationship
UML	Unified Modeling Language
IEEE	Institute of Electrical and Electronics Engineers