Software Requirement Specifications

Log-Based Testing Through Machine Learning For Hospital Management Systems

Version: 1.0

Project Code	F23-301D
Supervisor	Dr. Atif Tahir
Co Supervisor	Dr. Nouman Durrani
Project Team	Hafsa Baig Samia Azeem Maryam Raheem
Submission Date	2023-12-10

Document History

Version	Name of Person	Date	Description of change
1.0	Hafsa Baig	2023-11-19	*Document Created*
1.1	Dr. Nouman Durrani	2023-12-04	*Added Non-functional requirements*
1.2	Samia Azeem	2023-12-05	*Updated UC002*
1.3	Hafsa Baig	2023-12-07	*Added UC003*
1.4	Maryam Raheem	2023-12-07	*Updated Functional Requirements*
1.5	Dr. Atif Tahir	2023-12-09	*Reviewed Document*
1.6	Hafsa Baig	2023-12-10	*Final Version for submission*

Distribution List

Name	Role
Dr. Atif Tahir	*Supervisor*
Dr. Nouman Durrani	*Co- Supervisor*

Document Sign-Off

Version	Sign-off Authority	Sign-off Date
1.2	Dr. Nouman Durrani	2023-12-04
1.5	Dr. Atif Tahir	

Table of Contents

1. INTRODUCTION *7*

1.1. Purpose of Document 7
1.2. Intended Audience 7
1.3 Abbreviations 7
1.4. Document Convention 7

2. OVERALL SYSTEM DESCRIPTION *8*

2.1. Project Background 8
2.2. Project Scope 8
2.3. Not In Scope 8
2.4. Project Objectives 8
2.5. Stakeholders 8
2.6. Operating Environment 8
2.7. System Constraints 8
2.8. Assumptions & Dependencies 8

3. EXTERNAL INTERFACE REQUIREMENTS *9*

3.1. Hardware Interfaces 9
3.2. Software Interfaces 9
3.3. Communications Interfaces 9

4. FUNCTIONAL REQUIREMENTS *10*

4.1. *FUNCTIONAL HIERARCHY* *10*

4.2. Use Cases 10
4.2.1. [Title of use case] 10

5. NON-FUNCTIONAL REQUIREMENTS *11*

5.1. Performance Requirements 11
5.2. Safety Requirements 11
5.3. Security Requirements 11
5.4. User Documentation 11

6. REFERENCES *12*

7. APPENDICES *13*

1. Introduction

1.1. Purpose of Document

The purpose of this document is to outline the functional and nonfunctional requirements for the development of the Hospital Data Anomaly Detection System. It serves as a foundation for system design, implementation, and testing.

1.2. Intended Audience

The document is intended for:

- *Software Developers*
- *System Architects*
- *Quality Assurance Teams*
- *Project Manager*
- *Project Stakeholders*

It provides a comprehensive understanding of the system requirements for all involved parties.

1.3 Abbreviations

ML: Machine Learning

IT: Information Technology

UC: Use Case

API: Application Programming Interface

1.4 Document Convention

This document follows a standard convention:

- *Font: Arial*
- *Font Size: 12 for the main text*
- *Headings: Bold for easy navigation and readability.*

2. Overall System Description

2.1. Project Background

In the context of the healthcare industry, hospitals are dealing with vast amounts of patient data, electronic health records, operational information and event logs.[1] The efficient management of this data is crucial for providing quality patient care, optimizing resource allocation, and ensuring compliance with healthcare. This hospital data anomaly detection through Machine learning system is positioned to address specific challenges related to data anomalies, irregularities, and potential security breaches within the hospital information ecosystem.

The existing hospital systems face challenges related to data/event anomalies.[1] These challenges have a direct impact on patient care and operational efficiency. Hospitals are testing their system manually or automated but still require resources and a lot of time so our project aims to detect anomalies using machine learning algorithms to be more effective and fast.

2.2. Project Scope

The project includes:

- *User authentication and authorization*
- *Real-time anomaly detection*
- *Data Visualization/Analysis*
- *Notification System for critical anomalies*

2.3. Not In Scope

The system will not involve any external devices.

2.4. Project Objectives

The objectives include:

- *Enhancing security.*
- *Improving anomaly detection accuracy by using ML algorithm.*
- *Providing real-time data visualization.*

2.5. Stakeholders

Stakeholders involved will be:

- *Medical Professionals*
- *Administrators*
- *IT professionals*
- *Software developers and testers*

2.6. Operating Environment

Hardware platform:

Our system will operate on standard computing hardware commonly found in hospital IT environments.

Operating System:

The system will be designed to operate on Windows operating systems commonly used in healthcare IT environments.

Network Environment:

The network environment includes Internet connectivity.

Software components and Applications:

The system will interact with various software applications:

Web Browsers: Compatibility with standard web browsers for user interface access.

Our system will be dependent on the following components:

Machine learning libraries, Web frameworks and APIs.

2.7. System Constraints

The System includes following constraints:

- *Software constraints*

  The use of specific machine learning libraries for anomaly detection imposes constraints on the compatibility and versioning of these libraries.

- *Hardware constraints*

  The user interface must be designed to accommodate variations in end-user devices, including desktop computers, laptops, and tablets.

- *Cultural constraints*

  The user interface and any accompanying documentation must consider language diversity within the hospital environment.

- *Environmental constraints*

  In hospital environments with potential noise pollution, the system's user interface and alerting mechanisms should account for a noise-sensitive context.

- *User constraints*

  The system must cater to diverse user profiles within the hospital setting, including medical professionals, administrators, and IT personnel. The user interface should be intuitive and customizable to accommodate varying levels of technical expertise.

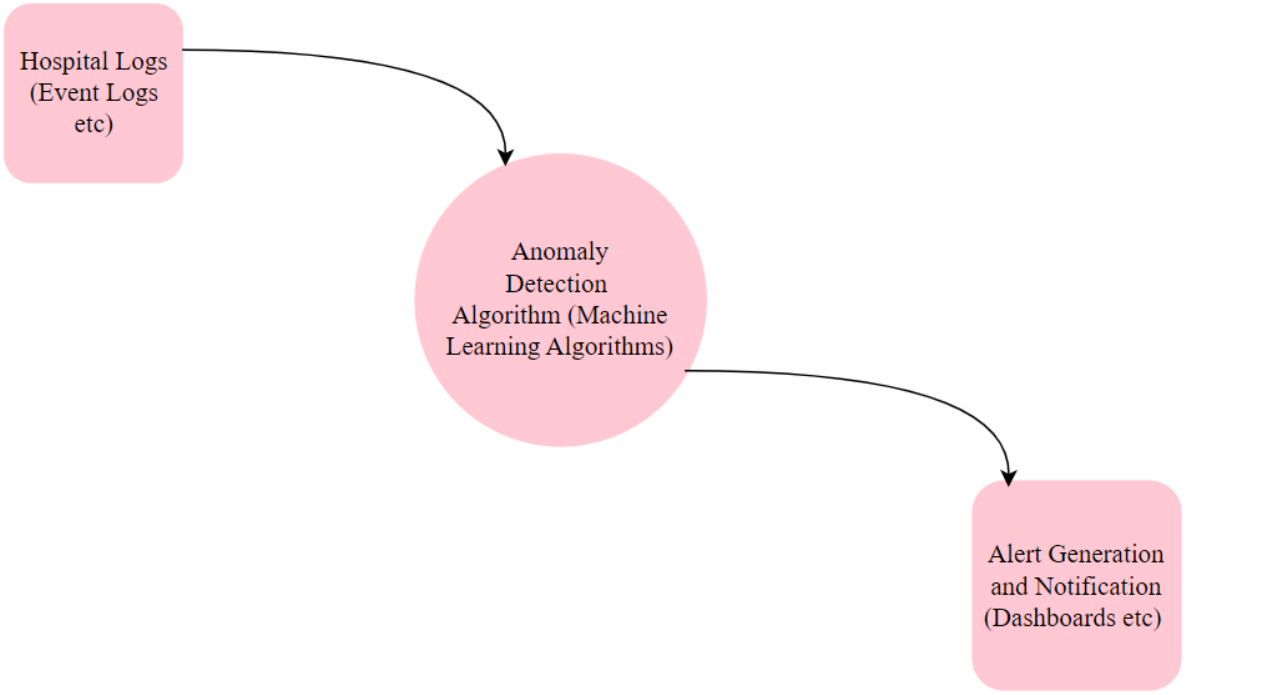
- *Off the shelf components such as web frameworks or security software, may impose constraints related to compatibility, licensing and versioning.*

2.8. Assumptions & Dependencies

- *It is assumed that the users using the system will adhere to data privacy regulations.*
- *The effectiveness of the system depends on the availability of labeled data for training machine learning models.*
- *The system is dependent on a reliable and representative dataset to ensure accurate anomaly detection.*

3. External Interface Requirements

Context Diagram:



3.1. Hardware Interfaces

The system will interface with the following hardware components:

- *The system will be deployed on a cloud server with sufficient storage capacity.*
- *Compatibility with desktop computers which are very commonly used by organizations, especially hospitals.*
- *Compatibility with laptops as the system will have the responsive design.*

3.2. Software Interfaces

- *The system will interface with a Database Management System for efficient data storage and retrieval.*

  Type: MongoDB Compass

  Version: 1.39.1

- *The system will relies on machine learning for implementing anomaly detection algorithms:*

  Scikit-learn

  Classification Algorithms(DT, LR, SVM etc)

- *We will use Following technologies:*

  Frontend: React.js

  Backend: Node.js

3.3. Communications Interfaces

To ensure secure communication, the system will implement the following:

- *All communication between the user and backend components will be encrypted using HTTPS.*
- *Patient data transmitted and stored by the system will be encrypted to ensure confidentiality and integrity.*

4. Functional Requirements

4.1. Functional Hierarchy

4.1.1 Data Ingestion and preprocessing

Sub-Function 1: Retrieve and preprocess hospital logs from various sources.

Sub-Function 2: Validate and clean log data for consistency.

4.1.2 Anomaly Detection Module

Sub-Function 1: Apply machine learning algorithms to identify anomalies.

Sub-Function 2: Evaluate patterns and deviations in the log data.

4.1.3 Alert Generation

Sub-Function 1: Generate alerts for detected anomalies.

Sub-Function 2: Prioritize alerts based on severity levels.

4.1.4 Notification System

Sub-Function : Notification alert.

4.1.5 Dashboard

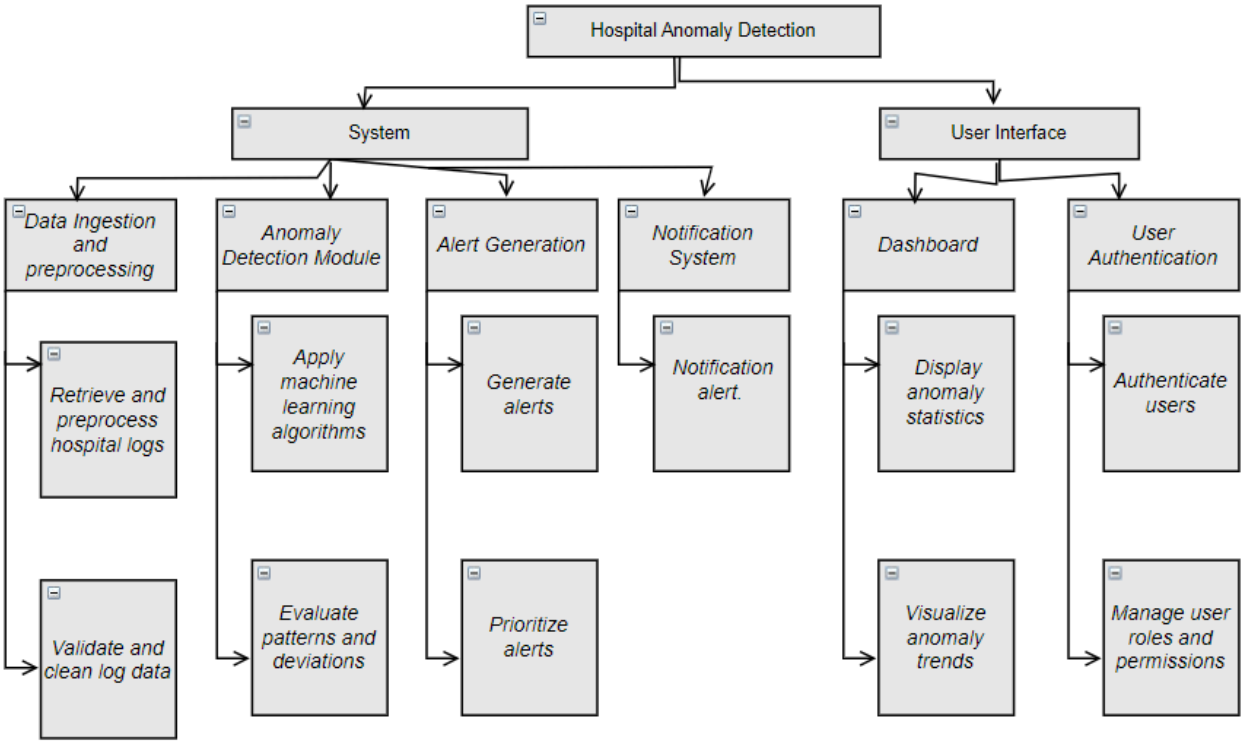
Sub-Function 1: Display anomaly statistics.

Sub-Function 2: Visualize anomaly trends.

4.1.6 User Authentication

Sub-Function 1: Authenticate users securely.

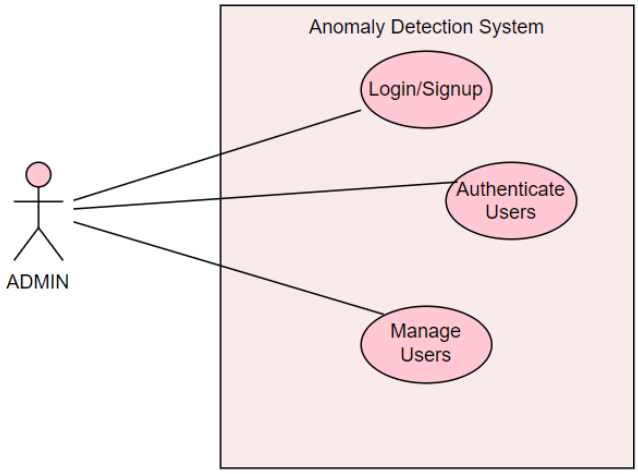
Sub-Function 2: Manage user roles and permissions.



4.2. Use Cases

4.2.1. Login and Authentication

Use Case Diagram:



Use Case Description:

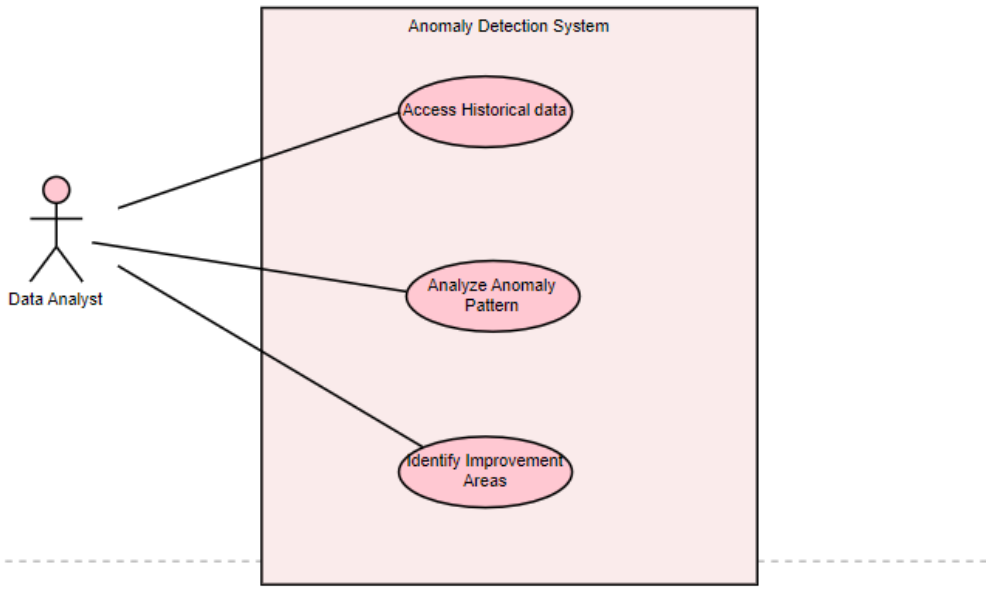
***Actor**:Admin*

***Description**: The admin will login into the system and also manage authentication settings of other users so that no invalid user will be accessing the system.*

UC001: Login and Authentication		
Use case Id:	UC001	
Actors: Admin		
Feature: login and authentication		
Pre-condition:	TheAdmin has access rights to configure authentication settings	
Scenarios ***1. The admin authenticates users' access.*** ***2. User roles and permissions are managed.***		
Step#	***Action***	***Software Reaction***
1.	The admin will login to the system.	The system authenticates admin details and moves to home page.
2.	The admin will manage users.	The system will allow admin to manage users.
Alternate Scenarios: Following are some alternatives		
1a: If a user enters invalid credentials, the system will only give 3 chances to enter valid credentials.		
Post Conditions		
Step#	***Description***	
1.	Login and authentication settings successfully configured by the administrator.	
Use Case Cross referenced	Authenticate User	

4.2.2. Reviewing Anomaly Trends

Use Case Diagram:



Use Case Description:

Actor*: Data Analyst*

Description*: The data analyst interacts with the system to review historical anomaly trends. This use case includes analyzing anomaly patterns, and identifying what improvements can be done to overcome these.*

UC002: Reviewing Anomaly Trends		
Use case Id:	UC002	
Actors: Data Analyst		
Feature: Anomaly Analysis		
Pre-condition:	The system has historical data available for analysis.	
Scenarios ***3. The Data analyst access data through system dashboard*** ***4. Anomaly patterns are analyzed.*** ***5. Identify areas of improvement.***		
Step#	***Action***	***Software Reaction***
1.	The data analyst navigates to the historical data section.	The system loads historical data on the dashboard.

2.	The data analyst identifies patterns.	The system may suggest areas for improvement..

Alternate Scenarios: N/A

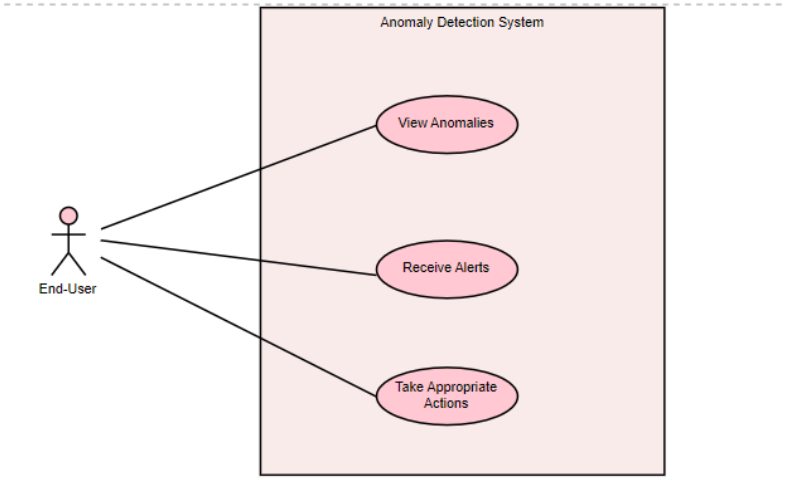
Post Conditions

Step#	***Description***
1.	The data analyst has gained insights into historical anomaly trends and found areas for improvement.

Use Case Cross referenced	Authenticate User, logged in

4.2.3. Analyzing Anomalies

Use Case Diagram:



Use Case Description:

Actor*: End-User*

Description*: The system will allow end users to view real-time anomalies, receive alerts, and take appropriate actions in response to detected anomalies.*

UC003: Analyzing Anomalies		
Use case Id:	UC003	
Actors: End-User, Analyst		
Feature: Anomaly Analysis		
Pre-condition:	The system is operational and has access to real-time log data	
Scenarios ***6. The hospital staff views anomalies on the system dashboard*** ***7. End-User takes appropriate actions in response to alerts*** ***8. The system generates alerts for detected anomalies.***		
Step#	***Action***	***Software Reaction***
1.	The end user navigates to the system dashboard	The system displays real-time anomalies.
2.	The end user reviews the list of anomalies	The system visualizes real-time anomalies data on the dashboard.
3.	Anomaly detection module identifies a critical anomaly.	The system generates an alert.
4.	Hospital staff receives the alert notification	The system prioritizes the alert based on severity..
Alternate Scenarios: Following are some alternatives		
1a: If no anomalies, the system will display 'no anomalies' ***2a: The system prioritizes alerts.***		
Post Conditions		
Step#	***Description***	
1.	The end user has successfully analyzed real-time anomalies and taken appropriate actions..	
Use Case Cross referenced	Authenticate User, logged in	

5. Non-functional Requirements

5.1. Performance Requirements

5.1.1 Speed

The system must achieve real-time anomaly detection with a response time not exceeding 2.5 seconds. The speed of anomaly detection is crucial for timely decision making and intervention.

5.1.2 Precision

The system is required to achieve a minimum accuracy rate of 95% in detecting anomalies.

5.1.3 Reliability

For continuous monitoring and timely anomaly detection the system is expected to maintain an uptime of at least 99%.

5.2. Safety Requirements

The system must implement robust measures to ensure the confidentiality of hospital data. Access to sensitive information must be restricted to authorized personnel only. To ensure continuous operation, the system should have redundancy and failover mechanisms in place.

5.3. Security Requirements

5.3.1 User authentication and authorization

Access to system functionalities must be role-based, with different user roles having specific permissions. This ensures that users only have access to the functionalities necessary for their roles.

5.3.2 Data security

All the hospital data transmitted and stored by the system must be encrypted.

5.4. User Documentation

Following is the list of the user documentation components that will be delivered along with the software:

- *User manuals*
- *Online help*
- *Tutorials*

6. References

[1] He, S., Zhu, J., He, P., & Lyu, M. R. (2020). Experience Report: System Log Analysis for Anomaly Detection. 2020 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE). doi:10.1109/issre.2016.21

[2] He, S., Zhu, J., He, P., & Lyu, M. R. (2020). Experience Report: System Log Analysis for Anomaly Detection. 2020 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE). doi:10.1109/issre.2016.21

[3] Zhu, J., He, S., Liu, J., He, P., Xie, Q., Zheng, Z., & Lyu, M. R. (2019). Tools and Benchmarks for Automated Log Parsing. In 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)

7. Appendices

Glossary:

Admin*: A user with access to manage and configure a system.*

Dashboard*: A visual representation of anomalies.*

Anomaly*: Deviation from normal behavior in the hospital logs.*