
Software Requirements Specification

Project: Industrial quality area

Document file

Date	Revision	Author	Verified quality	dep.
01/20/2024	1.0	Martinez Lopez Santi Erubey Barragan Hernandez Antonio Lemus Tarango naomi Denisse		

Document validated by the parties on date:

By the client	By the supplying company
Fdo. D./ Dña	Fdo. D./Dña

Document file

1. Introduction

- 1.1. Purpose
- 1.2. Product Scope
- 1.3. Staff Involved
- 1.4. Definitions and Acronyms
- 1.5. References
- 1.6. Overview

2. General Description

- 2.1. Product Perspective
- 2.2. Product Functionality
- 2.3. User Characteristics
- 2.4. Constraints
- 2.5. Assumptions and Dependencies
- 2.6. Apportioning of Requirements

3. Specific Requirements

- 3.2. Functional Requirement
- 3.3. Non-functional requirements
- 3.4. Performance Requirements
- 3.5. Non-functional requirements
 - 3.5.1. Optimization
 - 3.5.2. Security
 - 3.5.3. Availability
 - 3.5.4. Intuitive
- 3.6. System Attributes
- 3.7. External Interface

Appendix

1. Introduction

This document describes the functional and non-functional requirements for the development of a quality management system. The system will enable an organization to efficiently oversee and optimize processes related to quality assurance, certification, and compliance, ensuring adherence to established quality standards across key areas such as process control, product evaluation, operational safety, regulatory compliance, and continuous improvement initiatives..

1.1. Purpose

The purpose of the project is to have a control of the products that are handled in an industrial facility, showing the functional and non-functional requirements will make known what is the purpose of this.

1.2. Product Scope

The system will allow sector workers to check the status of products passing through IoT device scanning and the app's camera. By leveraging IoT devices, the system will automatically collect data on product quality, such as weight, dimensions, temperature, or any other relevant metrics, depending on the type of product. This information will be analyzed and stored in real time.

Through the app's camera, workers will also have the capability to perform manual quality inspections by scanning product labels, QR codes, or barcodes.

1.3. Staff Involved

Name	Martinez Lopez Santi Erubey
Professional Category	TSU-Software Development
Responsibilities	Software programming
Contact Information	032305914@ut-tijuana.edu.mx

Name	Lemus Tarango Naomi Denisse
Professional Category	TSU-Software Development
Responsibilities	documentation and database
Contact Information	0323105950@ut-tijuana.edu.mx

Name	Barragan Hernandez Antonio
Professional Category	TSU-Software Development
Responsibilities	IOT process
Contact Information	3023105853@ut-tijuana.edu.mx

1.4. Definitions and Acronyms

Name	Description
User	Person who uses the system to manage the inventory
SRS	Software Requirements Specification
FR	Functional requirement
NFR	Non-Functional requirement

1.5. References

Reference	Title
Standard IEEE 830	IEEE
Fontela, C. (2012). ISBN: 978-987-1609-22-2	UML Software Model for Professionals.

1.6. Overview

In the first part we will show a brief introduction of what the document is about, together with its purpose. In the second part we will show the general description of the project and at the end the requirements.

2. General Description

2.1. Product Perspective

The system will be a mobile application developed in JAVASCRIPT that will manage activities related to the Quality Management System. Sector workers will be able to realize different actions related to the quality sector.

2.2. Product Functionality

The software project consists of an application focused on the quality process of medical products, specifically single-use devices (disposable products) which are the following:

- Needles
- Syringes
- Catheters
- Medical gloves
- Masks
- Bandages

The program consists of having a quality control of these products, these would be in boxes which will have a QR code and will pass through a conveyor belt that will have a weight sensor, in case it does not have its code or does not comply with the indicated weight, a notice will be given by the application in which the person in charge will have to review the product and make a report of the incident and its QR will be scanned to give a confirmation notice and follow its process.

We will use a database already established by another department, more products cannot be added, only queries can be made.

The following requirements must be met:

1. The QR code must have information about the type of product (product name, product ID, production date, name of the company to which it will be addressed), date, weight.
2. The report will be used to record the error of the product problem and will have a record of the error giving the following data: date, time, name of the person in charge, product, product number, description. To then record your QR code, this being your confirmation that the product was confirmed.
3. The tables are already previously made, the products already registered in the database to be able to make queries about their quality packaging. Once everything is done, another department will be in charge of its distribution.

2.3. User Characteristics

The users that will be used will be only one, and it will be the person in charge of the quality area.

User type	admin
Training	Review and monitor the presenting problem.
Activities	monitor issues and resolve them through the app

2.4. Constraints

- The program will be executed in the form of an app.
- Language and technologies of use: HTML, CSS, JavaScript and MySQL.

2.5. Assumptions and Dependencies

- It must be a stable system for the user.
- Only trained people can use it.
- The materials needed must meet the requirements.

2.6. Apportioning of Requirements

- Performance optimization.
- Advanced security.
- Implementation of new areas.
- Fix bugs and problems that arise in the system

3. Specific Requirements

- User Interface: The application must have an intuitive and user-friendly interface.
- Equipment Verification: Users must be able to verify the quality of the product
- Reporting issues: Users should be able to submit a report about problems with a product when passing through the quality verification device.
- Reports: Users must submit a report when the product is verified showing the time, date, and who verified the product..

3.2. Functional Requirement

Requirement number	RF01
Requirement name	read QR codes
Description of the requirement	Sector employees must be able to manually perform product quality inspections using the mobile app's camera for scanning product labels, QR codes, or barcodes.
NON-functional requirement	•
Priority of the requirement	High

Requirement number	RF02
Requirement name	Automatic system that monitors
Description of the requirement	The system must automatically collect and analyze data from IoT devices to monitor product quality metrics in real time.
NON-functional requirement	•
Functional requirement	
Priority of the requirement	High

Software Requirements Specification

Requirement number	RF03
Requirement name	identify faults
Description of the requirement	The system must generate automatic alerts to sector workers if a product fails the quality verification process.
NON-functional requirement	•
Priority of the requirement	High

Requirement number	RF04
Requirement name	make reports
Description of the requirement	The system must allow employees to submit reports documenting product inspection results, including the time, date, and identity of the inspector.
NON-functional requirement	•
Priority of the requirement	Hlgh

Requirement number	RF05
Requirement name	inspect for faults
Description of the requirement	The system must provide detailed reports of product inspections, including metrics such as weight, dimensions, and other key quality factors.
NON-functional requirement	•
Priority of the requirement	Hlgh

Requirement number	RF06
Requirement name	inspect for faults
Description of the requirement	The system must provide detailed reports of product inspections, including metrics such as weight, dimensions, and other key quality factors.
NON-functional requirement	•
Priority of the requirement	Hlgh

Requirement number	RF07
Requirement name	Real-time dashboard
Description of the requirement	The system must provide a real-time dashboard for supervisors, displaying quality metrics, alerts, and inspection status summaries.
NON-functional requirement	•
Priority of the requirement	Hlgh

Software Requirements Specification

Requirement number	RF08
Requirement name	Customizable inspection templates
Description of the requirement	The app must allow employees to attach multimedia files, such as photos or videos, to inspection reports for better documentation of defects or issues.
NON-functional requirement	•
Priority of the requirement	Hlgh

Requirement number	RF09
Requirement name	Support for multimedia attachments
Description of the requirement	The system must provide detailed reports of product inspections, including metrics such as weight, dimensions, and other key quality factors.
NON-functional requirement	•
Priority of the requirement	Hlgh

Requirement number	RF10
Requirement name	Analytics and insights
Description of the requirement	The system must generate periodic analytical reports, providing insights on quality trends, recurring issues, and process efficiency.
NON-functional requirement	•
Priority of the requirement	Hlgh

3.3. Non-functional requirements

Requirement number	RNF01
Requirement name	Security
Description of the requirement	security is not necessary as the system will be able to function without it.
Priority of the requirement	High

Requirement name	Optimization
Description of the requirement	even if it is not optimized quickly, the system can still be used.
Priority of the requirement	High

Requirement number	RNF02
Requirement name	User interface
Description of the requirement	The app must be simple but understandable for the user.
Priority of the requirement	High

3.4. Performance Requirements

- **RP1: Response Time** The system must be able to process any request (such as product inspections, generating reports, or updating data) within a maximum time of 2 seconds under normal load conditions to ensure real-time operation and efficiency.
- **RP2: Scalability** The system must be able to handle a minimum of 200 concurrent users while maintaining the same level of performance. The system must be capable of managing up to 3,000 active quality inspection records and IoT-connected devices without affecting performance, ensuring smooth operation even with high data volumes.
- **RP3: Resource Consumption** CPU usage on the server must not exceed 75% during normal operations, and memory usage must not exceed 60% under medium load scenarios. The system must be optimized to reduce bandwidth usage, particularly when transmitting large quality inspection reports or data from IoT devices, ensuring that network resources are efficiently utilized.
- **RP4: Availability** The system must have an availability rate of 99%, ensuring it is available to users at all times for product inspections and quality control tasks. The system must implement backup and automatic recovery mechanisms in case of system failures, allowing for quick restoration of services.
- **RP5: Handling Large Volumes of Data** The system must be capable of storing and retrieving historical inspection records and quality data without significantly affecting performance. Queries about the product inspection history or data from IoT devices must execute within a maximum time of 3 seconds, ensuring that users can access critical information quickly.
- **RP6: Analytics and Insights**
The system must be capable of generating analytical reports and insights from large volumes of historical quality data within a maximum time of 5 seconds. Reports must include key trends, recurring issues, and process efficiency metrics, ensuring supervisors and management can make timely decisions.
- **RP7: Real-time Dashboard**
The system must update the real-time dashboard with quality metrics, alerts, and inspection summaries within 2 seconds of data being received or processed. The dashboard must remain responsive and provide seamless updates even under high data loads.
- **RP8: Customizable Inspection Templates**
The system must allow administrators to create, edit, and save customized inspection templates within 3 seconds. Retrieval of pre-existing templates must occur in less than 2 seconds, ensuring an efficient customization workflow.
- **RP9: Multimedia Attachments**
The system must support uploading multimedia attachments, such as photos and videos, with a maximum file size of 50 MB per file. Uploads must complete within 5 seconds under normal network conditions, and multimedia files must be retrievable within 3 seconds from the inspection records.

3.5. Non-functional requirements

3.5.1. Optimization

The system must be optimized to perform processes immediately without noticeable delays.

3.5.2. Security

Users should only be able to perform actions corresponding to their role within the company.

3.5.3. Availability

The program must be available at any time it is required during working hours.

3.5.4. Intuitive

The web page must have a user-friendly and intuitive interface, using different types of menus and modules that are easy to understand for any user at any given time.

3.6. System Attributes

1. Security

- Confidential data.
- All system activities must be logged and monitored.

2. Usability

- The system must be intuitive and easy to use for people with varying levels of technological skill.
- The training time required for users to operate the system efficiently should be minimized.

3. Portability

- The software must be compatible with multiple operating systems to ensure flexibility in server choice

3.7. External Interface

User Interface

- Description: The system must provide an intuitive and user-friendly graphical interface for employees.
- Technology: Languages such as JavaScript and other materials are being used for its implementation.
- Employees: They must be able to carry out the entire quality management process from the IoT device to verification in the app.

Appendix