Software Integration to achieve Poka-Yoke in pre-packing tightness automation

Sahana C[1], Nayanashree S[2], Monish H S[3], Poornima V, Dr. Jitendranath mungara

Dr. Jitendranath Mungara, Professor, Dept. of ISE, New Horizon College of Engineering, Bangalore, Karnataka.

[1234]B.E students, Dept. of ISE, New Horizon College of Engineering, Bangalore, Karnataka.

ABSTRACT: PF pump is a diesel fuel injection pump. This pumps are assembled by operators; after the assembly it is verified in the pre-packing tightness automation station. In this pre-packing tightness automation, the defects in the pf pumps are detected. This defects have to be eliminated otherwise it causes the customer complains.

The common defects that occur in pf pumps are:

- Air leakage
- Pressure check
- Leakage in pump joints

If the pf pump has any one of these defects, it may lead to the serious damage in the engine management system. This defects are detected with the help of pre-packing tightness application. This application is developed by using the concept of interlocking and traceability which is used to test the pump defects and to achieve the mechanism of poka-yoke i.e. zero defect pumps [1].

The aim of our project is to achieve zero defect pumps, reducing customer complains, enhancing productivity by minimizing time and cost.

Keywords: poka-yoke, pf pumps, traceability, interlocking, quality, tightness test.

I. INTRODUCTION

The word Poka means unintentional mistakes and yoke means to avoid, thus Poka-yoke means mistake proofing. It is a simple technique that helps to eliminate the troubles associated with defects and mistakes in operation. This concept was initiated by Dr. Shield Shinto and aims at achieving zero error. Poka-yoke is part of the zero quality control [2]. It is used to identify the defects at the source which in turn helps in lowering the cost and consumes time. Its purpose is to eliminate product defects by preventing, correcting or drawing attention to human errors as they occur. Poka-yoke is a method for building quality into a process by detecting errors that occur in the production to prevent the bad parts from passing down the stream.

Nowadays the main aim of the companies is to achieve high productivity, value of the product in the market and to satisfy the customer. This can be achieved by using the techniques of Poka-yoke [3].

Prevention is the basic principle of effective Poka-yoke. It makes the whole system error proof, i.e., it does not allow anyone to make mistakes either intentionally or unintentionally. Manufacturing industries make use of these techniques for high production. This technique is applied even in the software field [4]. This concept helps us detect the mistakes at early stages so that it can be corrected immediately and eliminate those defects at the initial stages itself.

In this paper, we explain how the concept of Poka-yoke is used in the pre-packing tightness automation to achieve zero defect products. In order to avoid defected pumps early, detection in terms of calibration and data interlocking is used. Clear propagation of tightness automation is provided in various stages and effective analysis is made. It is a fully automated testing station for testing, tightness of the pumps to avoid errors. In pre-packing tightness automation, the defects in the PF pumps are detected [5].

II. EXISTING SYSTEM

Earlier, the pumps were tested manually through visual detection, in which single pump was inserted in tightness tank containing the oil. Checking was done by dipping the pumps in oil, and if the bubbles occurred from the pump, such pumps were considered defected. Thus, there was no signs traceability and interlocking techniques.

III. PROPOSED SYSTEM

In the proposed system, an application is developed to detect the errors in the pump with the help of traceability and interlocking techniques. A Software database is created to store the employee data. The operator logins with his username/employee ID and password. Another database is maintained to store the details of the good pumps that were scanned and selected by the employees who have access to the application. This application also enables traceability.

IV. SYSTEM ANALYSIS AND REQUIREMENTS

SOFTWARE USED:

- C#:

  C# is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed byMicrosoft within its .NET initiative and later approved as a standard by ECMA (ECMA-334) and ISO (ISO/IEC 23270:2006). C# is one of the programming languages designed for the Common Language Infrastructure.

- SQL:

  SQL (Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data where there are relations between different entities/variables of the data. The scope of SQL includes data query, data manipulation (insert, update and delete), data definition (schema creation and modification), and data access control

- Windows Forms:

  A Windows Forms application is an event-driven application supported by Microsoft's .NET Framework. Unlike a batch program, it spends most of its time simply waiting for the user to do something, such as fill in a text box or click a button.

Project Execution Related Requirements:

Development Environment:

★ **Operating System:**

Windows 7 or Windows 10

★ **Tools& Languages:**

Microsoft Visual Studio - Visual C#

★ **Database:**

Oracle database 10g

V. SYSTEM ARCHITECTURE

System architecture is a conceptual model that defines the structure, behavior, and more views of a system. It comprises of system components that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages.

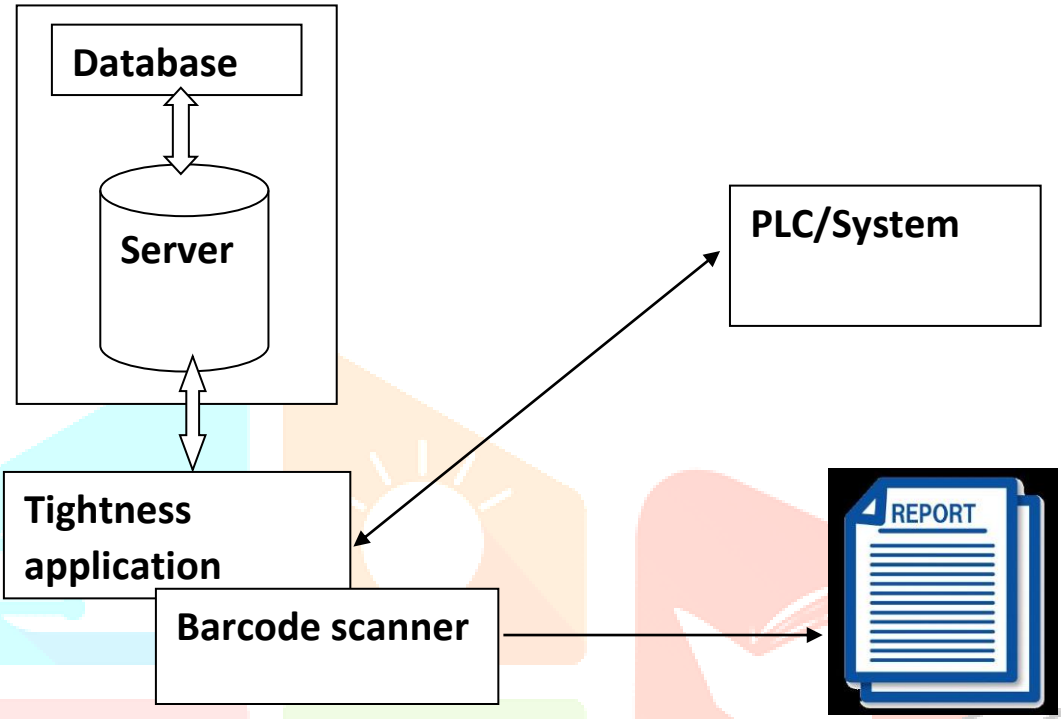

Fig 1 System architecture

The system architecture consists of following components:

1. Server

It is used to process the database queries. Server helps us to store and retrieve the data in the form of files.

2. Database

It is software that runs on the server.

Database is an organized storage collection of employee details (employee id and employee password) and pump details (pump type, MFD and pump number).

3. Tightness Application

It is used store the information about the pumps that are in good condition, and link the particular pump back to employee who is involved in the process of tightness testing.

4. Scanner

This component is used to scan the good pumps and store the information of pump such as pump number, manufacture date and serial number.

5. System/PLC logic

Programmable Logic Controller tracks all the operations involved in tightness testing of PF pumps.

VI. MODULES USED

Apart from application, these are several other modules that are used to achieve poka yoke. They are:

Barcode:

There is a unique bar code (Data Matrix code) on each and every pump. This data matrix code is two dimensional barcode consisting of black and white "cells" or modules. This cells are arranged in either square pattern or rectangular pattern which is known as matrix. In this the information can be text or numeric data. In this code the cell which are light represents 0 and the cell which are dark represents 1 or vice versa. When the barcode is scanned the required data is extracted from the pattern that is present in both horizontal and vertical component of the image. The purpose of the barcode on the pump is to track the pump. The barcode is scanned only for the pumps that are in good condition.

Barcode scanner/Barcode reader:

A barcode reader or barcode scanner is an electronic device that can read and output printed barcode to a computer. There is a light source, a lens and a light sensor converting optical impulses into electrical ones. Additionally, nearly all decoder circuitry analyzing the barcode's image data provided by the sensor and sending the barcode's content to the scanner's output port. There is a unique barcode on each and every pump. This barcode is scanned by the scanner only for the pumps that are in good condition [6].

Database:

A database is an organized collection of data. A relational database more restrictively is a collection of schemas, tables, queries, reports, views and others elements. It is software that runs on the server. Database is an organized storage collection of employee details (employee id and employee password) and pump details (pump type, MFD and pump number). Once the good pumps are scanned by the scanner then details about those pumps get stored in database. The details about the employee will be stored in the database by using schemas [7].

VII. TESTING AND RESULTS

Various tests were performed using the following test cases and the results were recorded:

Test id	User name	password	Expected output	Actual output	status
1.	Correct	correct	Login successful	Login successful	pass
2.	correct	incorrect	Password incorrect	Password incorrect	fail
3.	incorrect	correct	Username incorrect	Username incorrect	fail
4.	incorrect	incorrect	Username and password incorrect	Username and password incorrect	fail

Fig 2: Test cases

The following snapshots shows the output obtained after the execution of the application:

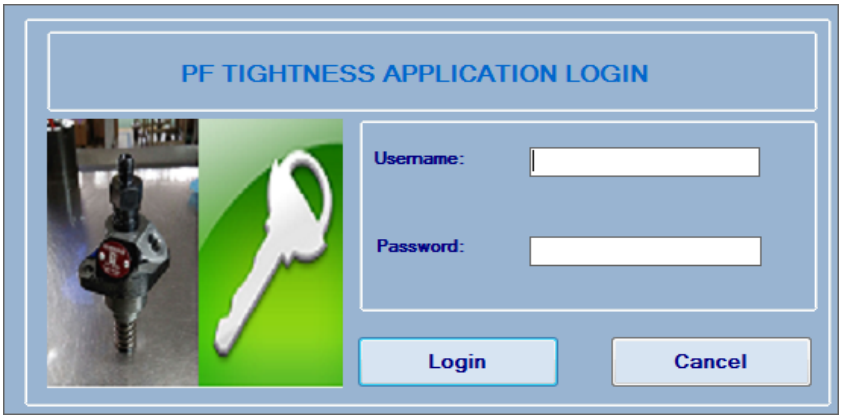

Fig.3: Snap shot of login page

Fig. 4: snap shot of application screen

Fig.5: snap shot of the pump that has not gone through all the stages

Fig.6: snap shot of good pumps that are scanned

Fig.7: snap shot for scanning the next set of pumps

VIII. FUTURE WORK

- The system currently focuses only on checking the leakages in the pf-pump.
- Only the pump that are in good condition are manually scanned using scanner and the details are stored in the database.
- It can be further enhanced to include sensor test, the amount of leakage and to indicate its selection/rejection with green/red light depending on the extent of bubbles the pump releases which is dipped in tank containing oil.
- In future the system can be made to display the number of selected and the number of rejected pumps on LCD and LED monitor for quick reference to track of fine pumps.

IX. CONCLUSION

Pf pumps are assembled in BOSCH company. This pump is assembled in mass production, during assembly there may be some parts which may pass by operators or there could be some defects in the pump. Hence this pump is checked in pre-packing tightness automation station. This station is mainly used for detecting the defects in the pump at earlier stages. This station is developed based on the concept of poka-yoke. The fundamental principle behind poka-yoke is to find errors at their source and to prevent their conversion to defects.

For this pre-packing tightness automation station, we developed a tightness application. When the employee tests the pumps, the pumps that are in good condition will be scanned and this data will be stored in database and the bad pumps will be sent back to previous stations. The information of employee is stored in database for processing the pumps which helps in traceability.

REFRENCES

[1] PERFORMANCE ANALYSIS ON BOSCH PF DIESEL FUEL INJECTION PUMP international Journal of Mechanical And Production Engineering, ISSN: 2320-2092, Volume- 2, Issue-9, Sept.-2014 1RATHI VIVEK M.,

[2] Dr. Arakerimath. R. Rachayya. “Parametric analysis and optimization of fuel delivered and torque by FIP of Diesel engine using genetic algorithm”, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 3, May-Jun 2012, pp. 018-022.

[3] ‘POKAYOKE’ OR QUALITY BY MISTAKE PROOFING DESIGN AND CONSTRUCTION SYSTEMS ABSTRACT Iris D. Tommelein1

[4] POKA-YOKE TECHNIQUE, METHODOLOGY & DESIGN Rajan kumar1Indian Journal of Engineering

[5] Grout, J. (2007). Mistake Proofing the Design of Health Care Processes. AHRQ, available online at http://www.ahrq.gov/qual/mistakeproof/mistakeproofing.pdf

[6] https://en.wikipedia.org/wiki/Barcode_reader

[7] https://en.wikipedia.org/wiki/Database