Development of Dynamic Web Portal for Centre for Research in Smart Manufacturing

Software Requirements Specification

Version 1.0

26/01/2018

Group G

GVS Akhil : 160001021

Saptarshi Ghosh : 160002052

Souvik Mandal : 160002056

Aravind Ravikumar : 160001006

Rishika Patel : 160001050

Prepared for

CS 258 Software Engineering

Spring 2018

Revision History

Date	Description	Author	Comments
26/01/18	Version 1.0	Group G	First Revision

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date

Table of Contents

1. Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the "IIT Indore Industry 4.0 Cyber Twin System" (ISW) software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

1.2 Scope

This Software is a Cyber Physical System that provides Manual Interface for Machines such as 3D Printers, CNC Milling etc. with ability to adapt new Generic Machines. It attempts to modernize the industry by using Machine Learning to predict life of a certain Machine thus increasing productivity and reducing wastage. It allows each Machine to make decentralized independent decisions to make the production process more efficient.

It provides user-friendly interface to enter the current state of the Machine. Only Authenticated Machines as determined by the ISW Server are allowed to log-in.

The Machine States are stored in a local server for Analytic Purpose and for Predicting the Failure of certain Machine/Tool using Machine Learning Algorithms such as k-Nearest Neighbor / Neural Networks.

1.3 Definitions, acronyms, and abbreviations

Terms	Definitions
User	Someone who interacts with this Software
Industrialist	Any Industry Person who interacts with the Main ISW Server
Seller	Any Person who wishes to sell specific Machine Parts

States	The Different States of a Machine in Manual Interface
CPS	Cyber Physical Systems
Industry 4.0	Current trend of automation and data exchange in manufacturing technologies incorporating CPS
Web Portal	A Web Application that provides the Manual Interface for a Machine
Stakeholder	Any Person who has interaction with the system who is not a developer
Operator	Any industry operator who works on a Machine
ML	Machine Learning
DEP	Dependency
MUI	Manual User Interface
TAG	A unique, persistent identifier contained in a PLanguage statement [2]
GIST	A short, simple description of the concept contained in a PLanguage statement [2]
MUST	The minimum level required to avoid failure contained in a PLanguage statement [2]
PLAN	The level at which good success can be claimed contained in a PLanguage statement [2]
WISH	A desirable level of achievement that may not be attainable through available means contained in a PLanguage statement [2]
DEFINED	The official definition of a term contained in a PLanguage statement [2]

1.4 References

[1] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

[2] Competitive Engineering, Tom Gilb

1.5 Overview

The remainder of this document includes two chapters. The second one provides an overview of the system functionality and system interaction with other systems. This chapter also introduces different types of stakeholders and their interaction with the system. Further, the chapter also mentions the system constraints and assumptions about the product.

The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences.

2. General Description

This section will give an overview of the whole cyber physical system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe the different type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

2.1 Product Perspective

The Cyber Physical System is a cluster of various Machines that operate under a centralized main server (“ISW”). Each Machine is associated with its own local Server that logs the different states of the Machine namely the Process its undergoing, data collected by various sensors, components which have failed and status of the operator.

Each Machine provides its own user-friendly Manual Interface on its local server where the operator can enter the current state of the Machine. It is a Web-based Interface accessible both by the dedicated Machine and Smartphones. The MUI must be professional, scalable and as well as reusable.

Each Local Server incorporates an Analytic Engine that is capable of taking decentralized decisions for the Machine and perform their tasks as autonomously as possible. The Local

Server specific to a machine for e.g. 3D Printers, CNC Milling etc. also maintains its own database which logs the previous states of the Machine.

The Analytic Engine performs various Machine Learning (ML) Algorithms to predict the failure of a Machine by taking in various sensor provided Inputs of Temperature, Vibrations, Stress/Strain etc. along with the states of the Machines taken directly from the user for which input cannot be taken from sensors.

The Prediction of the Analytic Engine is then sent to the Main ISW Server which provides the platform for different tasks such as putting up a notice for different parts that are required so that interested sellers might contact.

2.2 Product Functions

This Cyber Physical System provides a Manual User Interface which provides various options to select the current Machine State. The Operator needs to manually enter the Data here as no sensors are available to detect the states defined in this Interface. Other Data are collected via sensors and fed to the Analytic Engine.

The Selected States are stored in a database of the Computer dedicated to a specific Machine. The Manual Interface must be accessible via Desktop as well as Mobile Browsers to facilitate the ease of operation.

The Analytic Engine performs various ML Algorithms on the data taken as input from the MUI and sensors to predict the life of a Machine/its Parts. This predicted data is accessible from the ISW Main Server on request of an Industry Person.

2.3 User Characteristics

There are mainly three types of user that interact with the system: Operator, the Industry Person and the Buyer/Seller.

1. **Operator**
   This category consists of users that have access to the Local system dedicated to a Machine. Operator can access the Manual Interface by logging into the system and hence can change the state of the Machine. Access is restricted to only a single Machine.

2. **Industry Person**
   This category consists of users who hold important positions in the industry. Industry Person can look up information of each machine and can view the current status of different machines as well as their individual servicing needs. They can access the Main ISW Server.

3. **Buyer/Seller**

These Users are not part of Local Server interaction. Instead, they directly interact with the main ISW Server. They can see basic information of machines such as those which have failed and need replacement. Buyer/Seller can contact the industry for purchasing/selling different parts etc.

None of the users is expected to have any technical know-how of the working of the software.

2.4 General Constraints

The General Constraints on this software include

1. **Fast Response Time**

   The Web Portal on the Local Machine is also constrained by the Capacity of the Main Server. Requests from various Machines may be forced to queue which increases the fetch / write time affecting Performance. Software related delays will further add to this problem. Hence, the response time of the system should be as minimal as possible.

2. **User Friendly Interface**
   The GUI of the Manual Interface must be user-friendly for the convenience of the users.
3. **Security**
   Since it is a web-based platform, the confidential device specific information of each Machine must be secured and not prone to breaches which can cause loss of money for the industry.

2.5 Assumptions and Dependencies

❖ Hardware Dependencies

One Assumptions about the product is that it will always be used on Server Grade Hardware that has ability to process parallel data fast.

The Internet Connection / Availability of LAN is a major constraint for this Software. Since Data is fetched from the Main Server and sent back to the same, it is crucial that there is Connectivity between the Servers.

❖ Software Dependencies

The Manual Interface will make use of latest technologies which require a modern compatible browser associated with it. It is assumed that such condition is always met and the Computers have minimum hardware specification to run the latest browsers.

Since it is a web-based portal it is intended to be Operating System independent. It is also assumed that a OS is being used which supports any of the modern browsers. The Manual Interface will be accessible both from Mobile Phones and a Dedicated Desktop associated with each Machine.

Apart from that Servers must be running/having support for

- Node JS
- Python 3.6
- Mongo DB

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interface

Manual User Interface

- Login Page

    1. The MUI shall have a login portal which would be used to authenticate and identify each smart machine on the network.
    2. The login portal shall request the username and password of the specific smart machine that it is assigned to.
    3. The login portal shall prevent users from accessing the states page if the user is not authenticated.

- State of Machine Page

    1. The MUI shall provide a clean, scalable and re-useable GUI to efficiently record the states of the machine.
    2. The MUI shall display the active working state of the machine in an easy-to-read manner.
    3. The MUI shall be dynamic and shall automatically adjust its size with respect to the resolution of the rendering machine such as mobile phone/tablets/computer etc

4. The MUI shall be re-useable such that it can be used for other types of machine.
5. The MUI shall be flexible and allow administrators to add/modify/delete states of the machine.

3.1.2 Hardware Interface

Each smart machine on the network shall have a dedicated server and database. The server and database shall run on a computer assigned to each smart machine.

The server would collect information from various sensors located in the machine.

The ISW portal shall have dedicated servers to host the central network.

Since ISW portal and each smart machine are connected over the internet, all the hardware shall require to connect to the internet. E.g. Hardware: Modem, WAN – LAN, Ethernet Cross-Cable

3.1.3 Software Interfaces

- The MUI shall communicate with the ISW portal to authenticate the machine.
- The ISW portal shall communicate with the Machine server to request and retrieve data.
- The MUI shall communicate with the Machine server to send current status data.
- The Machine server shall communicate with the local MongoDB to log/retrieve data as well as to perform analytics on the data.

3.1.4 Communication Interfaces

- The main ISW portal shall communicate with the dedicated servers of the smart machines over the internet through the use of HTTP protocol.
- The sensors in the machine shall communicate with the dedicated machine server through Arduino / Raspberry Pi by emitting packets on the local network which shall be received by the local node.js server.
- The ‘Manual User Interface’ page shall be rendered on a web browser and the button clicks shall be recorded through POST requests to the server and shall be sent to the central ISW server through packets over the internet by using the socket-I/O node.js library.
- The data logged by the MUI and the sensors shall be stored in a NoSQL database known as MongoDB by using the ‘mongo-DB’ node.js library.
- The local machine server can run the analytical engine by executing python scripts with the help of ‘python-shell’ node.js library.

3.2 Functional Requirements

3.2.1 Manual Interface

3.2.1.1 Log-In Page

The Local Machine allows only Authenticated Users to login. Each User or Machine has its own Profile Created in the Main ISW Server. In order to use the Manual Interface, the following must be entered in the Login-Form.

- User-Id
- Password

Scenario: On Successful Login

Given the user is logged in,

The Manual Interface is presented

Scenario: Login Failed Due to Incorrect User-Id and/or Password

A warning is displayed stating incorrect Credentials and asks for login-info again.

3.2.1.2 Manual Interface

Each Machine has a dedicated Server that collects information about the Machine and stores the data locally as well as sends it to the main ISW Server. The Manual Interface allows to choose states of the Machine that cannot be taken via sensors such as Availability of Operator and Failure of a Certain Components.

The States are pre-defined (before deployment) but can be extended to every Machine. The States are majorly categorized as

- Active
- Failure
- Idle

Only a Single state can be selected at a time. The Active Modes / Failure Modes are specific to each Machine, the data of which is collected from the Client before-hand. The Category “Idle” also contains several sub categories such as ‘Operator Busy’, ‘Operator Unavailable’ as well as Machine specific states such as ‘No Raw Materials’ etc.

It Provides a *Generic Template* which can be customized for different Machines.

Scenario: On Selecting a State

Given the User is logged in,

And a Valid State is Selected,

The Currently selected state is stored on the Local Machine's Database.

3.2.1.3 Analytic Engine

The Analytic Engine is local to each Machine's dedicated server. It Performs k-Nearest Neighbour Pattern Recognition Algorithm to Analyse the Stored **States** and predict the failure of a certain Machine Part based on the recent Machine **States**.

ML Algorithms are executed via Python scripts at specific time as requested by ISW Server.

The final output of the ML Algorithms is sent back to the ISW Server.

DEP: Manual Interface

3.3 Non-Functional Requirements

3.3.1 Performance

Since the system is web based, the servers should be capable of handling large number of simultaneous requests. Thus, the internet bandwidth should be as high enough (2mbps recommended) to handle the large number of requests. It is recommended that the server hardware have at least 2GHz processing speed and 4GB RAM.

3.3.2 Reliability

The system should be able to manage machine status, lifetime even after some years without errors even though new machines may be added etc.

3.3.3 Availability

The system should have a downtime not greater than a few hours during which the buyer may not be able to view machine status, lifespan etc.

3.3.4 Security

Each machine personal data should store on machine own server and no other machine can view or change that data. Only a user verified by log in through ISW server can see machine data. To change or add machine data administration privilege required.

3.3.5 Maintainability

The database should be accessible to the administrators so as to carry out maintenance. The database should also be periodically backed up to prevent information loss due to system crashes.

3.3.6 Portability

The platforms on which the system runs should be generic enough to allow substantial amount of portability.

3.4 Design Constraints

The Web Portal on the Local Machine is constrained by the Capacity of the Main Server. Requests from various Machines may be forced to queue which increases the fetch / write time affecting Performance. Software related delays will further cause delays. This software must attempt to minimize such performance issues.

The Data Generated by the MUI/Sensors is in order of Gigabytes per sec. The Machine Learning Algorithm done by the Analytic Engine must be optimized enough to keep up with the huge data generated.

The web Ports used by the Software for sending Data must be in compliance with the companies' policy, since it is mandated that the ports must be different from regular web ports such as '80' or '443' so as to prevent throttling of network.

Also, the server framework recommended by the client is 'Node.JS'. This will ensure future re-usability of the code. Also 'MongoDB' is suggested as the Database framework by the same for above mentioned reasons.

3.5 Logical Database Requirements

The primary database management system that shall be used is MongoDB. The local machines store the data obtained through various sensors and Manual User Interface. The result after ML algorithm is run is also stored in the local database. This data would be sent to the main server on request.

The main server has a dedicated database. This main database contains separate databases for all the different machines in the network. These are the databases which would contain all the details of the machines in the network, including information such as user id and password for authentication, history of all states the machine has been in and the current state the machine is on.