

Factory Simulation Software System

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

**MD AZAM
222CS3112**

M.Tech Software Engineering

CONTENTS

- 1. Introduction**
 - 1.1. Purpose**
 - 1.2. Scope of Project**
 - 1.3. Description**
 - 1.4. Definitions**
 - 1.5. Environmental Characteristics**
 - 1.5.1. Hardware**
 - 1.5.2. People**
- 2. General Description**
 - 2.1. Product Perspective**
 - 2.2. Product function Overview**
 - 2.3. User Characteristics**
 - 2.4. Operating Environment**
 - 2.5. General Constraint**
- 3. Functional Requirement**
- 4. External Interface Requirement**
 - 4.1 User Interface**
 - 4.2 Hardware Interface**
 - 4.3 Software Interface**
- 5. Design and Implementation Constraint**
- 6. Assumptions**
- 7. Limitations of Project**

1.0 Introduction

The Software Requirements Specification is designed to document and describe the agreement between the customer and the developer regarding the specification of the software product requested. Its primary purpose is to provide a clear and descriptive “statement of user requirements” that can be used as a reference in further development of the software system. This document is broken into a number of sections used to logically separate the software requirements into easily referenced parts. This Software Requirements Specification aims to describe the Functionality, External Interfaces, Attributes and Design Constraints imposed on Implementation of the software system described throughout the rest of the document. Throughout the description of the software system, the language and terminology used is unambiguous and consistent throughout the document.

1.1 Purpose:

The software system being produced is called Factory Service Simulation Software or FSSS. It is being produced for factories interested in automating and utilisation of maximum power of machines as well as workers or adjusters (which repair the machines) available in the factory. This system is designed to “provide the efficient way” to use and manage different machines and the adjusters which repair those machines.

1.2 Scope Of Project:

This software system will be a Factory Service Simulation Software System. This system will be designed to maximize the ease that how the factory service manager should assign the adjuster to repair the machine to achieve maximum utilization of both machines and adjusters. The automation property of the system will make the working system very simple, fast and free of Humanerrors. By maximizing the user’s work efficiency the system will meet the user’s needs while remaining easy to understand and use.

1.3 Description:

This section includes details about what is and is not expected of the Factory Simulation Software system in addition to which cases are intentionally unsupported and assumptions that will be used in the creation of the FSS system.

The Factory Simulation Software System will allow a Factory Service Manager to assign adjusters to repair particular machines with maximum utilisation of both machines and adjusters. The service manager will have the option to login into the system and assign an adjuster to repair a machine. Service manager will also have the option to check the working status of adjusters and machines and submit the same working statistics of machines and adjusters to the factory head.

Factory Head will have the option to analyse the statistics and check machine utilisation and adjuster utilisation.

1.4 Definitions:

Head:

Factory Head manages the whole factory. Head gets statistics from the service manager and Head analyse those statistics.

Service manager:

A one who assigns work to adjusters and maintains the working statistics of adjusters and machines and submit that corresponding data or statistics to factory head. Service manager maintains separately Machine Queue and Adjusters Queue.

Adjuster:

A one who repairs machine in the factory by taking order from the service manager.

Machine Queue:

Here Machine queue refers to the queue of inoperative machines

Adjuster Queue:

Here Adjuster queue refers to the queue of free adjusters.

FSSS:

Factory Simulation Software System.

1.5 Environmental Characteristics:

1.5.1 Hardware:

The software requires a pc either running on windows or Linux and should have java IDE installed. There are no hardware or software requirements beyond these including, but not limited to, memory or specific software packages that need to be utilized nor software packages that need not be utilized.

1.5.2 People:

This software can be used by any user having basic skills in operating a computer since the user interface is very simple.

2. General Description

2.1 Product Prospective:

Earlier we have a traditional Software where everything like maintenance of machine etc., is done manually but now we are try to develop a software in a such a way that everything can be automated. Only little manual intervention is needed.

2.2 Product Functional Overview:

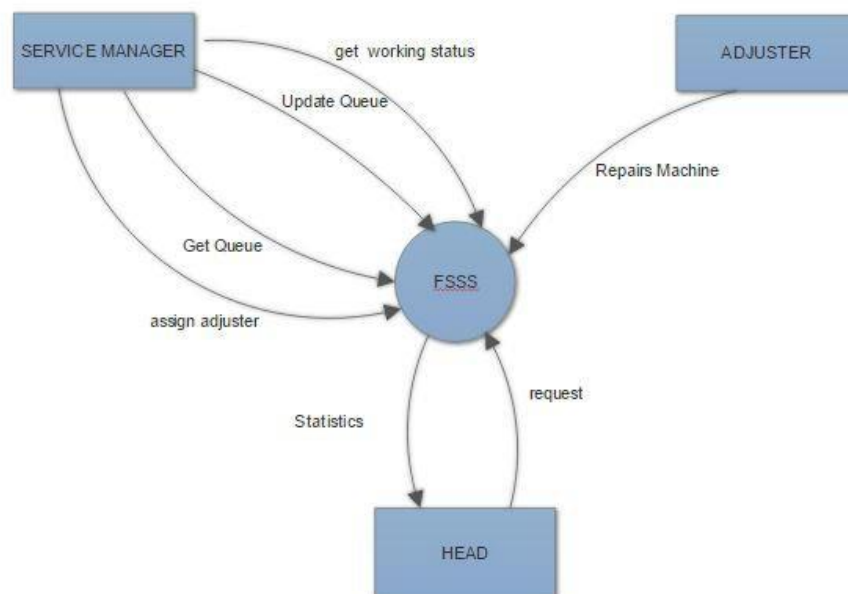
Context Diagram

The FSSS is shown as a process with the Adjuster, Service Manager and Head as external entities.

Inputs: Add Machine , Add Adjuster, Manage Queue.

Process: Factory Service Simulation Software (FSSS)

Outputs: Get Statistics



Data Flow Diagram

2.3 User characteristics:

User can be manager or an employee who is good at basic computer skills. He should be given a prior training to operate the computer and GUI interface of the software.

2.4 Operating environment:

It can be operate in both Windows XP and above as well as Linux environment.

2.5 General Constraint:

We need to install a proper power supply so system would run without any issue. As we as initial setup would also involve some intangible cost like Air Condition etc. To avoid heating of the system.

3. Functional Requirements:

Functional Requirements are those that refer to the functionality of the system, i.e. what services it will provide to the user. Nonfunctional requirements pertain to the other information needed to produce the correct system and are detailed separately.

The set of functionalities that are supported by the system are documented below

Assign Adjuster:

Description: Whenever there exist an inoperative machine and an adjuster to repair that machine then that adjuster gets assigned to repair that inoperative machine by the service manager.

At first service, the manager will access the queue of the inoperative machine and free adjusters. Then service manager will check for the adjusters who can repair those inoperative machines. After assigning adjusters to repair the machines service manager will update the queue of machines and adjusters and corresponding adjusters will set their own status as working.

Get machines queue:

Input: Get machine queue command from the service manager.

Output: service manager receives machines queue which are inoperative.

Get adjusted queue:

Input: Get the adjuster queue command from the service manager.

Output: service manager receives adjusters queue which are free.

Update machines queue:

Input: Update machine queue command from service manager with information to be updated. Processing: Machines queue gets updated.

Update Adjusters queue:

Input: Update adjusters queue command from service manager with information to be updated. Processing: Adjusters queue gets updated.

Repair machine:

Input: repair machine command from the service manager to adjuster .

Processing: adjuster will start repairing the machine and will update his status as working.

Update working status:

Input: No need of input. It will directly get executed when the repair machine will get executed.

Processing: Working status gets updated as working automatically.

Update machine status:

Input: Confirmation that machine got repaired completely.

Processing: machine status gets updated as working automatically.

Add inoperative machine:

Description: Service manager is regularly supposed to check if there is any inoperative machine in the factory which is not in the queue. Such a machine gets added to the queue of an inoperative machines and the corresponding queue gets updated.

Check any inoperative machine:

Input: check any inoperative machine command from the service manager.

Processing: it checks if there is any inoperative machine which is not in the queue.

Output: Inoperative machines not in the queue.

Add inoperative machine:

Input: Information about the inoperative machine not in the queue.

Processing: Adds inoperative machine to the queue.

Update machines queue:

Input: Update machine queue command from service manager with information to be updated. Processing: Machines queue gets updated.

Update Statistics:

Description: Service manager is regularly supposed to update statistics of machines and adjusters working to the head of the factory.

Regularly service manager maintains the working statistics of machines and adjusters. Service manager get these statistics and submit these to the head.

Get Working status:

Input: Get a working status command from the service manager.

Output: Service manager receives working statistics of machines and adjusters.

Update statistics:

Input: update statistics command from the service manager.

Processing: updating statistics to head.

Output: success message.

Analyse Statistics:

Description: Head receives statistics from the service manager. Head analyses statistics of machine utilisation and adjuster utilisation and responds to service manager accordingly.

Get Statistics:

Input: Get the statistics commands from the head.

Output: Head receives working statistics of machines and adjusters.

Analyse Statistics:

Input: Analyse statistics command from the head.

Processing: Calculating different parameters to analyse statistics

Output: Head receives results from the analysis of data.

Response:

Input: response String which gets sent to the service manager as a response to statistics.

Processing: Responding to service manager.

4 External Interface Requirements:

4.1 User Interfaces:

Only a mouse and keyboard are required for interacting with the system.

4.2 Hardware Interfaces:

Hardware: Personal Computer

Operation System: WindowsXP or more, linux, Mac

Internet Connection: Both Wired as well as Wireless(Wifi)

4.3 Software Interfaces:

The software will be coded in JAVA IDE using Eclipse.No other software interface required.

5. Design and Implementation Constraints

Security is not a concern for this system. The database may store passwords in “serializable” file and there doesn't need to be a password recovery feature nor lockout after numerous invalid login attempts. As such, the system may not work correctly in cases when security is a concern. We are not forcing users to have “strong password”. A strong password is a password that meets a number of conditions that are set in place so that user's passwords cannot be easily guessed by an attacker. Generally, these rules include ensuring that the password contains a sufficient number of characters and contains not only lowercase letters but also capitals, numbers, and in some cases, symbols

6. Assumptions

- Every machine will have a different id number.
- An adjuster may or may not repair all machines.
- Adjusters which can repair a particular machine will repair that machine in equal time.
- Adjusters can't deny the work that has been assigned to him/her.
- The service manager will assign only that adjuster to a particular machine that can repair that machine

7. Limitations of the software

As stated, security is not a concern of this project. As such, it is beyond the scope of this system to encrypt personal user data and information, prevent unauthorized login attempts, or any other concern of this nature. Additionally, the system is not responsible for incorrect information about the machine provided by the user.