

Factory Simulation Software System

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

Vedant Pate

14CS30021

Shubham Kumar

14CS30033

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. Overall Description**
 - 2.1. Functional Requirements**
 - 2.2. Use cases**
 - 2.3. Non Functional Requirements**
- 3. Requirements Specification**
 - 3.1. External Interface Requirements**
 - 3.1.1. User Interfaces**
 - 3.1.2. Hardware Interfaces**
 - 3.1.3. Software Interfaces**
 - 3.2. Functional Requirements**
 - 3.3. Detailed Non Functional Requirements**
- 4. Constraints**
- 5. Assumptions**
- 6. 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 Human-errors. 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 machine 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 option to check working status of adjusters and machines and submit the same working statistics of machines and adjusters to factory head.

Factory Head will have 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. Overall Description

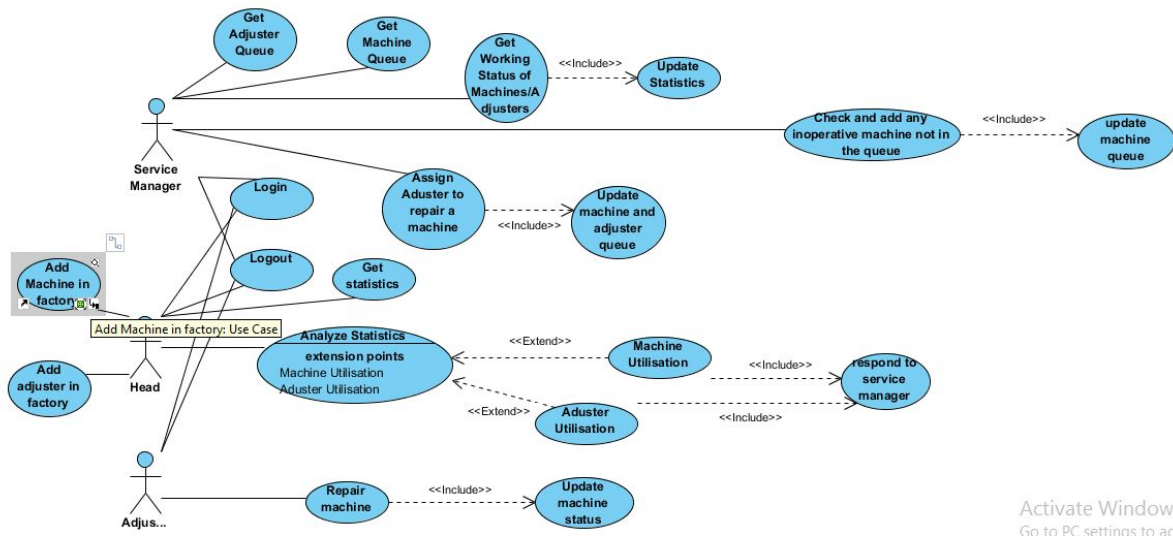
2.1 Functional Requirements:-

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.

2.2 Use Cases:-

In software and systems engineering, a **use case** is a list of steps, typically defining interactions between a role (known in Unified Modeling Language (UML) as an "actor") and a system, to achieve a goal. The actor can be a human or an external system.

In systems engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals.



Use Case: Login

Actors: Head, Service Manager, Adjuster

Type: Primary and essential

Description: Initiated when a user tries to access his account. The user is then Prompted to enter in their username and password in order to Proceed and access their account. Every user has to go through this step to do any work in the factory.

Includes: None

Use-Cases: None

Use Case: Get Adjuster Queue

Actors: Service Manager

Type: Primary

Description: After Login when user have to know information about adjusters which are currently free, this step is needed.

Generally this step is require for assign adjuster to machine use case.

Includes: None

Use-Cases: The Login use case must be completed.

Use Case: Get Machine Queue

Actors: Service Manager

Type: Primary

Description: After Login when user have to know information about machines which are currently nor working or inoperative, this step is needed. Generally this step is require for assign adjuster to machine use case.

Includes: None

Use-Cases: The Login use case must be completed.

Use Case: Assign Adjuster

Actors: Service Manager

Type: Primary

Description: When user has to assign particular adjuster to repair any inoperative machine, this step is required.

Includes: Update machine and adjuster queue use case

Use-Cases: The Login ,Get Adjuster Queue, Get machine Queue use cases must be completed.

Use Case: Update Machine And Adjuster Queue

Actors: Service Manager

Type: Primary

Description: When user make changes in the machine queue or in the adjuster queue this step needs to get executed.

Includes: None

Use-Cases: The Login ,Get Adjuster Queue, Get machine Queue use cases must be completed.

Use Case: Check and add inoperative Machine

Actors: Service Manager

Type: Primary

Description: User is supposed to check regularly if there is any inoperative machine not in the queue and then add that inoperative machine to the queue.

Includes: Update machine and adjuster queue use case

Use-Cases: The Login , Get machine Queue use cases must be completed.

Use Case: Get Working Status of Machines and Adjusters

Actors: Service Manager

Type: Primary

Description: User executes this step to collect working statistics of machine and adjuster.

Includes: Update Statistics use case

Use-Cases: The Login use case must be completed.

Use Case: Update Statistics

Actors: Service Manager

Type: Primary

Description: After completing the work when user needs to submit the working statistics to head this step gets initiated.

Includes: None

Use-Cases: The Login ,Get Working Status of machines and adjusters use cases must be completed.

Use Case: Get Statistics

Actors: Head

Type: Primary

Description: By this step User receives the working statistics of machines and users.

Includes: None

Use-Cases: The Login use case must be completed.

Use Case: Analyse Statistics

Actors: Head

Type: Primary

Description: Using this step user get the analysis of ongoing work in the factory and also gets to know about machine and adjuster utilisation in the factory.

Includes: Response to service manager

Extend: Machine Utilisation and adjuster utilisation

Use-Cases: The Login, Get Statistics use cases must be completed.

Use Case: Response to service manager

Actors: Head

Type: Primary

Description: This needs to respond on the analysis of adjuster utilisation and machine utilisation.

Includes: None

Use-Cases: The Login ,Get Statistics and Analyse adjuster and machine Utilisation use cases must be completed.

Use Case: Repair machine

Actors: Adjuster

Type: Primary and essential

Description: This step gets initiated by user to get to know that which machine user has to repair.

Includes: Update machine status

Use-Cases: The Login use case must be completed.

Use Case: Update machine Status

Actors: Adjuster

Type: Primary and essential

Description: After repairing the machine its user's responsibility to execute this use case and to update machine status as working.

Includes: None

Use-Cases: The Login ,repair machine use cases must be completed.

Use Case: Logout

Actors: Head, Service Manager, Adjuster

Type: Primary and essential

Description: Every user needs to logout from the system after completing the work. This is the one of the most important things user should remember to execute.

Includes: None

Use-Cases: The Login use case must be completed.

2.3 Non-Functional Requirements:-

There are requirements that are nonfunctional in nature. Specifically, these are the constraints the system must work within.

3. Requirements Specification

3.1 External Interface Requirements:-

3.1.1 User Interfaces:-

Only mouse and keyboard are required for interacting with system.

3.1.2 Hardware Interfaces:-

Hardware:	Personal Computer
Operation System:	WindowsXP or more , linux
Internet Connection:	Either LAN connection or Wi-Fi

3.1.3 Software Interfaces:-

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

3.2 Functional Requirements:-

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 manager will access the queue of 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 service manager.

Output: service manager receives machines queue which are inoperative.

Get adjustes queue:

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

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

Update working status:

Input: No need of input. It will directly get executed when 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 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 is there any inoperative machine which is not in the queue.

Output:- Inoperative machines not in the queue.

Add inoperative machine:

Input:- Information about 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 head.

Get Working status:

Input: Get 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 statistics command 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 send to service manager as response to statistics.

Processing: Responding to service manager.

3.3 Detailed Non-functional Requirements:-

The set of non-functional requirements can be stated as follows:

Queues should be maintained and updated at correct time.

Data must be saved properly

Correct data must be given

The software should be protected from customers and non-employees of the factory.

The latest version of java IDE is installed in the computer in which it is going to be run.

Username and passwords should be remembered and entered correctly.

4. 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.

5. Assumptions

Every machine will have 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.

Adjuster can't deny the work that has assigned to him/her.

Service manager will assign only that adjuster to a particular machine who can repair that machine.

6. 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 the incorrect information about the machine provided by the user.