Factory Simulation Software System

SOFTWARE REQUIREMENTS SPECIFICATIONS(SRS)

GROUP-6

Anurag Anand 19je0168
Arpit Agarwal 19je0180
Arvapalli Tejaswi 19je0185
Aryan Karn 19je0189

CONTENTS

1.0 INTRODUCTION

The primary purpose of SRS 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.It is designed to document and describe the agreement between the customer and the developer regarding the specification of the software product requested. 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 utilization 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 of 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 utilization 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 analyze the statistics and check machine utilization and adjuster utilization.

1.4 ENVIRONMENTAL CHARACTERISTICS:

1.4.1 Hardware and Software:

The software requires a pc either running on windows or MAC and should have a code editor. There are no hardware or software requirements beyond these including, but not limited to, memory or specific software packages that need to be utilized.

1.4.2 People:

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

1.5 DEFINITIONS:

FSSS:

Factory Simulation Software System.

Head:

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

Adjuster:

A person who repairs machines in the factory by taking orders from the service manager.

Service manager:

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

Machine Queue:

A queue is a linear data structure which follows the first in first out method .Here Machine queue refers to the queue of inoperative machines.

Adjuster Queue:

Here Adjuster queue refers to the queue of free adjusters.

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. Non-functional 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.

Following are the different use cases.

Use Case: Login

Actors: Head, Adjuster, Service Manager
Type : Primary and essential

Description: This use case is initiated when the user tries to log in to the system. The user is then Prompted to enter in their username and password in order to Proceed and access their account and these credentials are unique for every user. Two users can not have the same username. Every user has to go through this step to do any work in the factory.For login the user must first signup with our web software.

Includes: None
UseCases: None

Use Case: Get Adjuster Queue

Actors:Service Manager
Type:Primary

Description : This use case gives the information to the logged in users about the adjusters which are currently free.This use case is required to get an adjuster which can be assigned to inoperative machines.
Includes: None
Use Cases :The Login use case must be completed.

Use Case: Get Machine Queue

Actors:Service Manager
Type:Primary

Description : This use case gives the information to the logged in users about the machines which are inoperative.This use case is required to get a machine which requires an adjuster.

Includes: None
Use Cases :The Login use case must be completed.

Use Case: Assign Adjuster

Actors:Service Manager
Type:Primary

Description : This use case is required to assign free adjusters to inoperative machines depending on the status of machine and adjuster queues.

Includes : Update Machine and Adjuster Queue use cases.
Use Cases : 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 : This use case is called when there is modification in the queue as when some adjuster is assigned to an inoperative machine, the queue should be modified.

Includes: None.
Use Cases :Login, Assign Adjuster, Get Adjuster queue and Get Machine queue 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 a machine that is inoperative. If the machine is not in the queue and is inoperative then it is added to the queue. This will give mean time to failure (MTTF) for each type of machine (lathe machines, turning machines, drilling machines, soldering machines etc.)

Includes : Update machine and adjuster use case.
Use Cases :Login, Get Machine queue use cases must be completed.

Use Case: Check and add free adjusters

Actors: Service Manager
Type: Primary

Description : User is supposed to check regularly if there is an adjuster that is currently free.If the adjuster is not in the adjuster queue and is currently free then it is added to the queue

Includes : Assign Adjuster and Get queue use case.

Use Cases :Login , Get queue use cases must be called.

Use Case: Get Working Status of Machines and Adjusters

Actors: Service Manager
Type: Primary

Description : User executes this step to collect working statistics of machines and adjusters.

Includes :Update Statistics use case.
Use Cases : Login use cases must be completed.

Use Case: Update Statistics

Actors:Service Manager
Type:Primary

Description : After completing the work, when the user needs to submit the working statistics to the head, this use case is initiated.

Includes:None
Use Cases :Login , Get Working Status of machines and adjusters use cases must be completed.

Use Case: Get Statistics

Actors:Head
Type:Primary

Description : This use case gives the working statistics to the head ie. statistics of machines and adjusters.

Includes: None
Use Cases :The Login use case must be completed.

Use Case: Analyse Statistics

Actors:Head
Type:Primary

Description :Using this use case, the user gets the analysis of ongoing work in the factory and also gets to know about machine and adjuster utilization in the factory.

Includes: Response to service manager use case.
Extend :Machine utilization and adjuster utilization.

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

Use Case: Response to service manager

Actors:Head
Type:Primary

Description :This use case is used to respond to the analysis of adjuster utilization and machine utilization.

Includes:None
Use Cases : Login , Get Statistic and Analyze adjuster and machine Utilization use cases must be completed.

Use Case: Repair machine

Actors:Adjuster
Type :Primary and essential

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

Includes : Update machine status.
Use Cases : Login use case must be completed.
Use Case: Update machine Status

Actors:Adjuster
Type :Primary and essential

Description :After repairing the machine it's the user's responsibility to execute this use case and to update machine status as working as well as declare itself as a free adjuster.

Includes:None
Use Cases :Login use case 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 users should remember to execute.

Includes:None
Use Cases :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. REQUIREMENT SPECIFICATIONS

3.1 EXTERNAL INTERFACE REQUIREMENTS:

3.1.1 User Interfaces:

The user interface is basically divided into three main sections: Head, Adjuster and Service manager.

3.1.2 Hardware Interfaces:

Hardware: Personal Computer(Laptop or desktop)
Operation System: WindowsXP, linux, Mac etc.
Internet Connection: Either LAN connection or WiFi

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 to achieve different goals are documented below:

Assign Adjuster:

Description: Whenever there exists 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 the service manager will access the queue of inoperative machines and free adjusters. Then the service manager will check for the adjusters who can

repair those inoperative machines. After assigning adjusters to repair the machines, the service manager will update the queue of machines and adjusters and corresponding adjusters will set their own status as working.After the process is finished the machine changes its status to operative and the adjuster again becomes free.

Get Machine queue:

Input: Get machine queue command from service manager.
Output: service manager receives machines queue which are inoperative.

Get adjuster queue:

Input: Get adjuster queue command from service manager.
Output: service manager receives adjusters queue which are currently 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 the machine and will update its status as working.

Update working status:

Input: No need for input. It will directly get executed when the repair machine is 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 machine queue and the corresponding queue gets updated. A machine is inoperative/operative is reflected in its status.

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 inoperative machines 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.
The service manager maintains the working statistics of machines and adjusters. Service manager gets these statistics and submits them to the 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.

Analyze Statistics:

Description : Head receives statistics from service manager .Head analyzes statistics of machine utilization and adjuster utilization and responds to service manager accordingly.

Get Statistics:

Input: Get statistics command from the head.
Output: Head receives working statistics of machines and adjusters.

Analyze Statistics:

Input: Analyze statistics command from the head.
Processing:Calculating different parameters to analyze statistics
Output: Head receives results from the analysis of data.

Response:

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

3.3 DETAILED NON-FUNCTIONAL REQUIREMENTS:

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

Data must be saved properly in the database.
Status of machine(operative/inoperative) and adjuster(free/busy) must be maintained properly as it is required for analysis purposes.
Queues should be maintained and updated at the correct time.
Correct data must be given to different internal functions.
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.
The user should be provided with tutorials and manuals on how to use the software.
Username should be unique for security reasons

4. 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.
Adjuster can’t deny the work that has been assigned to him/her.Service manager will assign only that adjuster to a particular machine who can repair that machine.

---

5. CONSTRAINTS

Security is not a concern for this system. The database may store passwords in a “ 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 a “strong password”. A strong password is a password that meets a number of conditions that are set in place so that the 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. 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.