Software Requirements Specification

Factory Simulation Software

Team 9

Gosangi Gunadeep	**118CS0142**
Arijeet De	**118CS0163**
Anwesha Nayak	**118CS0186**
Ankit Kumar Sahoo	**118CS0205**
Aravind Arul	**118CS0583**
Aushish Roy Pentapaty	**118CS0686**

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Anwesha	08/03/2021	Draft 1	1.0

1. Introduction

1.1 Purpose

The SRS document is laid down for "Factory Simulation Software". The purpose of this software is to provide the factory management with a system to strike the perfect balance between number of workers and its machineries alongside utilizing all the data available about the machineries to its fullest for an optimal service routine.

1.2 Definitions and Abbreviations

MTTF: Mean Time To Failure is machine type dependent.
OID: A unique Object Identifier is available for every entry in the database.
Adjuster: The worker responsible for repairing the machines.
Service Manager: The worker responsible for service routine.
Factory Manager: The highest governing body of factory to whom analysis data are shown.

1.3 Intended Audience and Reading Suggestions

This SRS document is intended for the software developers, service routine managers, factory managers and adjusters.

1.4 Product Scope

The scope of this software is limited to the efficient management of adjusters in the maintenance of machineries and providing the factory heads with the simulation data of these details. The software is to be managed by a service routine manager and shouldn't be expected to be automated on the field of updating the records.

2. Overall Description

2.1 Product Perspective

Factory simulation software is made for automation of maintenance routine and drawing inferences from the MTTF and adjuster routine that will be beneficial to the factory.

2.2 Product Functions

The software updates the entries:

- *In the maintenance routine queue made by Service Manager*
- *In machine database and status made either by Service Manager or Adjuster*
- *In adjuster database made either by Service Manager or Adjuster*
- *In simulation window when asked by the Factory Manager*

2.3 User Classes and Characteristics

2.3.1 **Adjusters** should be able to

- Update own details.
- Change own password.
- Change status of the currently assigned machine.
- View his assigned task.

2.3.2 **Service Manager** should be able to

- Update own details.
- Change own password.
- View the Adjuster and Machine Database.
- Add a new Machine to the database.
- Update (Includes deletion of) entries from Adjuster Database.
- Update (Includes deletion of) entries from Machine Database.
- Assign an adjuster a new task.

2.3.3 **Factory Manager** should be able to

- Update own details.
- Change own password.
- View the Adjuster and Service Manager Database.
- Simulate the MTTF by machine type.
- Simulate the adjuster to machine relationship.

2.4 Operating Environment

Operating System: Windows
Database: SQL
Web Platform: Internet Explorer 10, Firefox 85, Chrome 86, Edge 89, Opera 72.
System Requirement: 1GB RAM, 512MB Disk Space

2.5 Design and Implementation Constraints

No lockout is implemented for numerous invalid login attempts. The users need not have strong passwords. However, the factory manager is required to provide a common secret key to avoid undesirable database entries from people outside of the organization.

2.6 Assumptions and Dependencies

It's assumed of all the machineries and adjusters to have unique IDs. The Service Manager is assumed to have a decent knowledge in data entry and the factory manager is only interested in analyzing the Adjuster to Machine efficiency. It's also assumed that the adjuster assigned a task by the service manager is available and efficient enough to carry it out and updates the machine status after his job.

3. External Interface Requirements

3.1 User Interfaces

The system will make use of icons and toolbars to provide a uniform look across users.

3.2 Hardware Interfaces

Minimum System Requirements:
Input Devices: Mouse, Keyboard
Output Devices: Monitor
Network Devices: Server PC, Client PC, Router
Hard Disk space: 512Mb
RAM: 1GB
Processor: Pentium III

3.3 Software Interfaces

The FSS is a stand-alone Desktop application developed in Python. However for smooth functionality it's advised to have Python 3.0 or higher versions installed in both client and Server PCs.

3.4 Communications Interfaces

The communication architecture must follow the client-server model. Communication between the client and server should utilize a REST-compliant web service and must be served over HTTP Secure (HTTPS).

4. System Features

The software has following features:

4.1 User Login

4.1.1 Description and Priority

This is the welcome window of the software and is of high priority. The user needs to get through this window in order to be able to use the features

4.1.2 Stimulus/Response Sequences

If provided with correct credentials the users will be directed to their corresponding home page however if wrong details are provided then error message will be displayed.

4.1.3 Functional Requirements

Username: The User should be able to enter Username.

Password: The User should be able to enter Password.
Login: Takes User to the Home Page when provided with correct credentials.
Sign Up: Directs to the new registration window.
Forgot Password: Directs to the Forgot Password Window

4.2 New Registration

4.1.1 Description and Priority

This window allows the new employees to register themselves. It's of medium priority.

4.1.2 Stimulus/Response Sequences

If provided with correct Secret Key it'll register the New Entry into User Database.

4.1.3 Functional Requirements

Username: The User should be able to enter Username.
Password: The User should be able to enter Password.
Secret Key: The User should be able to enter Secret Key.
Submit: Registers the User if correct Secret Key is provided.

4.3 Forgot Password

4.1.1 Description and Priority

This high priority window allows the new employees to recover their passwords.

4.1.2 Stimulus/Response Sequences

Sends an Email to the employee mail with the current password.

4.1.3 Functional Requirements

User mail: The User should be able to enter Username.
Send: Sends the current password to the provided mail if mail ID is found in database.

4.4 Machine Window

4.1.1 Description and Priority

This high priority window is responsible for maintaining the machinery database.

4.1.2 Stimulus/Response Sequences

Updates the database as instructed.

4.1.3 Functional Requirements

Insert: Adds a new machine to the database.
Search: Searches for a machine with given ID and displays details such as machine type, MTTF and working status.
Update: Updates the Working Status of a machine.
Delete: Deletes the machine and all its related details from the database.

4.5 Adjuster Window

4.1.1 Description and Priority

This is the home window for Adjusters and is of high priority.

4.1.2 Stimulus/Response Sequences

Updates the Adjuster Database and fetches details of .

4.1.3 Functional Requirements

Change Details:	The User should be able to update their details.
Assigned Tasks:	The User should be able to view the assigned tasks.
Change Task Status:	The User should be able to update his task status which automatically changes the status of the machine the adjuster worked on.

4.6 Service Manager Window

4.1.1 Description and Priority

This is the home window for Service Manager and is of high priority.

4.1.2 Stimulus/Response Sequences

Updates the Adjuster and Machine Database and fetches routine details.

4.1.3 Functional Requirements

Change Details:	The User should be able to update their details.
Assign Tasks:	The User should be able to assign new tasks to adjusters.
Check Machine Status:	The User should be able to check machine details.
Check Adjusters:	The User should be able to fetch Adjuster details.
Update Database:	The User should be able to add/delete machines/adjusters.

4.7 Factory Manager Window

4.1.1 Description and Priority

This is the home window for Factory Manager and is of high priority.

4.1.2 Stimulus/Response Sequences

Analyses the Adjuster and Machine relation and fetches employee details.

4.1.3 Functional Requirements

Change Details:	The User should be able to update their details.
Analyse:	The User should be able to analyse the MFFTs and maintenance routine relation.
Simulate:	The User should be able to simulate the adjuster-machine relation.
Check Employee:	The User should be able to fetch employee details.
Security Update:	The User should be able to update employee passwords and Secret Key.
Dismiss Employee:	The User should be able to fire an employee which in turn restricts all the access of the said employee to the database.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Response Time should be less than a second for all the operations except analysis and simulation results. The response time for analysis and simulation shouldn't exceed more than 5s.

5.2 Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database dating back up to one year that was backed up to archival storage (a portable hard disk of memory 1TB) and reconstructs a more current state by reapplying or redoing the operations of committed maintenance from the backed up log, up to the time of failure.

5.3 Security Requirements

Security systems need database storage just like many other applications. Thus, the security factory managers must choose their database partner carefully.

5.4 Software Quality Attributes

- ***AVAILABILITY:*** *The adjuster should be available on the specified date and specified time.*
- ***CORRECTNESS:*** *The software should display correct skillsets of the adjusters, correct MTTF of machines and the correct maintenance queue.*
- ***MAINTAINABILITY:*** *The service managers should maintain correct schedules of maintenance.*

5.5 Business Rules

The adjusters shouldn't have direct access to the maintenance queue. The factory manager is the only one in charge of Secret Key generation and also the only one having access to all the database. The service manager can't delete an OID indicating details filled by a Factory Manager.