

**CS29006 Software Engineering**

# **Software Requirements Specifications**

**For**

# **Factory Service Simulation Software**

**(FSSS)**

- **Group 12**
  - **Ayush Pattnayak(19CS10014)**
  - **Kamalesh Garnayak(19CS10074)**
  - **Sayantan Saha(19CS30041)**

## Table of Contents

<i>Table of contents</i>	
<i>Revision History</i>	<i>pg no</i>
1. <i>Introduction</i>	
1.1. <i>Purpose</i>	3
1.2. <i>Document Conventions</i>	3
1.3. <i>Intended Audience and Reading suggestions</i>	3
1.4. <i>Product Scope</i>	3
1.5. <i>References</i>	3
2. <i>Overall Description</i>	
2.1. <i>Product Perspective</i>	4
2.2. <i>Product Functions</i>	4
2.2.1. <i>Machine</i>	
2.2.2. <i>Adjuster</i>	
2.2.3. <i>Service Manager</i>	
2.3. <i>User Classes and characteristics</i>	5
2.3.1. <i>Machine type</i>	
2.3.2. <i>Adjuster type</i>	
2.3.3. <i>Machine</i>	
2.3.4. <i>Adjuster</i>	
2.3.5. <i>Service Manager</i>	
2.4. <i>Operating Environment</i>	6
2.5. <i>Design and Implementation Constraints</i>	6
2.6. <i>User Documentation</i>	7
2.7. <i>Assumption and Dependencies</i>	7
3. <i>External Interface Requirements</i>	
3.1. <i>User interface</i>	7
3.2. <i>Hardware Interface</i>	8
3.3. <i>Software interface</i>	9
3.4. <i>Communication Interfaces</i>	9
4. <i>System Features</i>	
4.1. <i>Use Case: Login</i>	9
4.2. <i>UML Use Case diagram</i>	10
4.3. <i>Use Case: Get Adjuster Queue</i>	10
4.4. <i>Use Case: Get Machine Queue</i>	10
4.5. <i>Use Case: Assign Adjuster</i>	11

4.6.	<i>Use Case: Update Machine and Adjuster Queue</i>	11
4.7.	<i>Use Case: Check and Add inoperative machine</i>	11
4.8.	<i>Use Case: Get Working Status of Machines and Adjusters</i>	11
4.9.	<i>Use Case: Update Statistics</i>	12
4.10.	<i>Use Case: Get Statistics</i>	12
4.11.	<i>Use Case: Repair machine</i>	12
4.12.	<i>Use Case: Update Machine Status</i>	12
4.13.	<i>Use Case: Logout</i>	12
5.	<i>Other Non-Functional Requirements</i>	
5.1.	<i>Performance Requirements</i>	13
5.2.	<i>Safety Requirements</i>	13
5.3.	<i>Security Requirements</i>	13
5.4.	<i>Software Quality Attributes</i>	13
5.5.	<i>Business Rules</i>	13
6.	<i>Other Requirements</i>	
	Appendix A: Glossary	14
	Appendix B: Analysis Models	15
	Appendix C: To Be Determined List	15

## Revision History

Name(s)	Date	Reason for change(s)	Version
Sayantana Saha,Ayush Pattnayak,Kamalesh Garnayak	19-03-2021	Initial draft	1.0

# 1.Introduction

## 1.1 Purpose

It is proposed to develop a software which can be used by factories to determine the optimum number of adjusters that they should employ. The existing manual based system for simulation has a lot of disadvantages and is time-consuming. The software developed will help the factories interested in automating and obtaining the best possible utilisation of machines and adjusters in an efficient way over the manual system like accurate results,time-efficient, secure etc.

## 1.2 Document Conventions

The entire document is written in 'Arial' font. Every requirement statement has to have its own priority.

## 1.3 Intended Audience and Reading Suggestions

This document is intended for project developers and users who want to view project requirements and specifications. It describes various functional and non-functional requirements of the software.

## 1.4 Product Scope

This software will maximise the ease with which the factory service manager must assign the adjuster to machines in order to achieve maximum utilisation of machines and adjusters. It is designed in a user-friendly way that is simple to use. It gives accurate results eliminating human errors in a time-efficient manner. It is also a secured system, giving control only to the factory service manager. Hence we maximise the work of efficiency of the system.

## 1.5 References

- [Java: The Complete Reference 8th Edition by Herbert Schildt](#)
- [TutorialsPoint/Java](#)
- [Google](#)
- [Stack Overflow](#)

- [GeeksForGeeks](#)

## 2.Overall Description

### 2.1 Product Perspective

This software is built to determine the optimum number of adjusters needed to repair the faulty machines of the factory. It can add new machines as per availability in the factory. Similarly, it can add new adjusters as per the availability. The service manager has to sign up first, and then he/she can login using his/her login credentials. After that, the manager can maintain the faulty machines and available adjusters queues and allot them properly. Along with that, the manager can edit the information of the machines and the adjusters. Finally, he/she can check the machine and adjuster utilisations.

### 2.2 Product Functions

There are three users(actors) for this software.

- 1) Machine
- 2) Adjuster
- 3) Service Manager

The set of functions supported by this software are as follows:

#### 2.2.1 Machine

1.Checking Working Status: Every machine checks whether the machine is working or it has some faults and needs repairing.

2.Checking Repairing Status: Every faulty machine checks whether the machine is in the queue waiting to be repaired or it is being repaired by an appropriate adjuster.

3.Counting No. of working days: Every machine keeps a record of no. of days it worked out of the total simulation period.

### **2.2.2 Adjuster**

1.Checking Working Status: Every adjuster checks whether it is repairing any appropriate faulty machine or it is waiting in the queue.

2.Counting No. of working days: Every adjuster keeps a record of no. of days it worked out of the total simulation period.

### **2.2.3 Service Manager**

1.Adding New Machines and New Adjusters: The Service Manager can add new machines and adjusters with respective information in the simulation.

2.Editing Details: The service manager can edit the machine and adjuster details before simulation.

3.Checking Details: He/she can check the working details of machines and adjusters.

4.Simulating the system: He/she can simulate the system to check the machine and adjuster utilizations. It can be simulated for a small period(like 1-2 years) or a large period(like 15-20 years).

5.Finding optimum adjusters: He/she can edit the adjuster details and find when the machine and adjuster utilisations are optimum.

## **2.3 User Classes And Characteristics**

### **2.3.1 Machine Type**

This class has the upper-level machine types. It is an abstract class of Machine class. It contains information like name, MTTF, repairing type, quantity etc. For example, a turning machine is a machine type.

### **2.3.2 Adjuster Type**

This class has the upper level adjuster types. It is an abstract class of Adjuster class. It contains information like id, the machine Types that adjuster repairs, quantity etc.

### **2.3.3 Machine**

This class is inherited from abstract class Machine Type. It contains information like name, MTTF, repairing type, quantity, working status, repairing status , no. of days it worked etc.

#### ***2.3.4 Adjuster***

This class is inherited from abstract class Adjuster Type. It contains information like id, the machine types that adjuster repairs, working status, waiting status , no. of days it worked etc.

#### ***2.3.5 Service Manager***

This is a new class. It contains user id, password etc. Using this login credentials it can have connections among machines and adjusters and can get details of them. It has an unique simulating function.

### ***2.4 Operating Environment***

This software is developed in JAVA, running on Windows 10 x64 Architecture. It should also be compatible with 64-bit Operating Systems have JAVA installed and connected to the internet.

### ***2.5 Design and Implementation Constraints***

Security is not a concern for this system. The database may store passwords in “serialisable” 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 passwords”. 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.

As stated, security is not a concern for this project. As such, it is beyond the scope of this system to encrypt personal user data and information, prevent unauthorised 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.

## **2.6 User Documentation**

Every machine will have a different id number.

Every machine's repairing time is an integer no. of days.

An adjuster may or may not repair all machines.

Adjusters who 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. The service manager will assign a particular machine to only that adjuster who can repair that machine.

These manuals should be followed by the user.

## **2.7 Assumptions and Dependencies**

This software has been targeted at Windows and Linux

Operating System. It depends on the online database and JAVA. This software requires an internet connection to use and store data in an online database. Since it is developed using JAVA, it is platform-independent.

# **3.External Interface Requirements**

## **3.1 User Interface:**

- Home Page
  - Create Account interface
  - Manager login
    - Username
    - Password
- Create Account Interface
  - Enter details:
    - Name
    - Date of birth
    - Username



- Password
  - Create
  - Exit
- After manager login:
  - Add a new machine
    - Enter details:
      - Machine type
      - MTTF
      - Repair time
      - Machine Count
    - Add machine
  - Add a new adjuster
    - Enter details:
      - Adjuster ID
      - Add compatible machines
      - Adjuster count
    - Add adjuster
  - Display current machines
    - Shows all machines
    - Delete machine
      - Choose
      - Deletes machines and corresponding adjusters
  - Display current adjusters
    - Shows all adjusters
    - Delete adjuster
      - Choose
      - Deletes the chosen adjuster
  - Enter simulation time
    - Enter time in days/months/years
  - Simulate
- Simulation Results
  - Show machine utilisation
    - Shows utilisation in % for each machine
    - Shows average utilisation
  - Show adjuster utilisation
    - Shows utilisation in % for each adjuster
    - Shows average utilisation

### **3.2 Hardware Interface:**

- A general PC/laptop with a mouse(touchpad) and keyboard are the only basic requirements.
- Recommended system specifications include RAM(2GB +), processor(1.5Ghz +) , free disk space(1GB +) and memory(1GB +).

### **3.3 Software interfaces:**

- Operating system: WindowsXP(+), Linux or Mac
- Java needs to be installed through the JRE kit available freely online.
- Java IDE (recommended: Eclipse editor)

### **3.4 Communication Interfaces:**

- The system can work offline; however, a good stable internet connection is preferable.

## **4.System Features**

### **4.1 Use Case: Login**

Actors: Service Manager

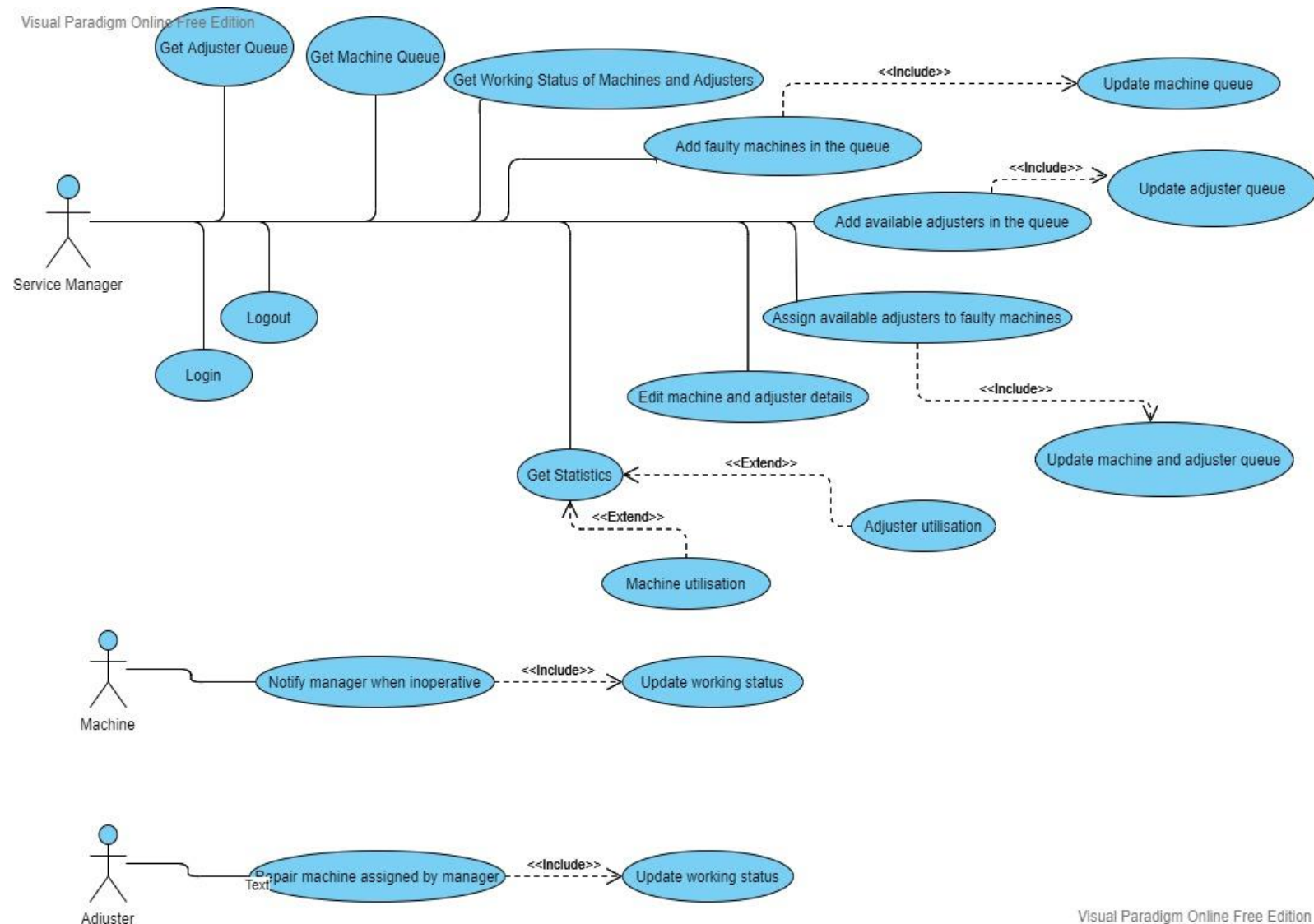
Type: Primary and essential

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

UseCases: None

## 4.2 UML Use Case Diagram



### 4.3 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 required for assigning adjusters to machine use cases.

**Includes:** None

**UseCases:** The Login use case must be completed.

#### **4.4 Use Case: Get Machine Queue**

Actors: Service Manager

Type: Primary

Description: After Login, when users have to know information about machines that are currently not working or inoperative, this step is needed. Generally, this step is required for assigning adjusters to machine use cases.

Includes: None

UseCases: The Login use case must be completed.

#### **4.5 Use Case: Assign Adjuster**

Actors: Service Manager

Type: Primary

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

Includes: Update machine and adjuster queue use case

UseCases: The Login, Get Adjuster Queue and Get Machine Queue use cases must be completed.

#### **4.6 Use Case: Update Machine And Adjuster Queue**

Actors: Service Manager

Type: Primary

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

Includes: None

UseCases: The Login, Get Adjuster Queue and Get Machine Queue use cases must be completed.

#### **4.7 Use Case: Check and add inoperative Machine**

Actors: Service Manager

Type: Primary

Description: The 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

UseCases: The Login and Get Machine Queue use cases must be completed.

#### **4.8 Use Case: Get Working Status of Machines and Adjusters**

Actors: Service Manager

Type: Primary

Description: The user executes this step to collect working statistics of the machines and adjusters.

Includes: Update Statistics use case

UseCases: The Login use case must be completed.

#### **4.9 Use Case: Update Statistics**

Actors: Service Manager

Type: Primary

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

Includes: None

UseCases: The Login, Get Working Status of machines and Adjusters use cases must be completed.

#### **4.10 Use Case: Get Statistics**

Actors: Service Manager

Type: Primary

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

Includes: None

UseCases: The Login use case must be completed.

#### **4.11 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

UseCases: The Login use case must be completed.

#### **4.12 Use Case: Update machine Status**

Actors: Adjuster

Type: Primary and essential

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

Includes: None

UseCases: The Login and Repair Machine use cases must be completed.

#### **4.13 Use Case: Logout**

Actors: Service Manager

Type: Primary and essential

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

Includes: None

UseCases: The Login use case must be completed

## **5.Other Non Functional Requirements**

### **5.1 Performance Requirements**

The software works in the most time-efficient way presently.

The database management must be cost-effective. The requirement is that the queue is updated from time to time, and that the correct data is provided.

Whenever there is an update in machine versions, the simulation results have to be reverified.

The user should also install the latest Java IDE for the best results.

### **5.2 Safety Requirements**

The software should be protected from customers and employees of the factory as they may tamper with the database storing the details regarding machines and adjusters. passwords and login must be remembered and kept undisclosed.

### **5.3 Security Requirements**

The software should be secure with proper login and password so that only the factory manager can provide data or change data regarding the machines and adjusters in the database.

### **5.4 Software Quality Attributes**

The software is user friendly and has an appealing interface.

It is expected to be fast, secured, accurate and platform-independent.

### **5.5 Business Rules**

The software is designed such that the factory service manager can only access the software and give the inputs of machines and adjusters.

## 6. Other Requirements

The developers should be contacted in case of

- (i) Any existential issues with the project, such as technical snags and incorrect results
- (ii) Any updated requirements at a later point of time in order to improve the system.

## Appendix A: Glossary

Service Manager: One who maintains machine and adjuster queue and assigns machines to the adjusters. He/she can get statistics of machine and adjuster utilisation.

Adjuster: One who can repair inoperative machines.

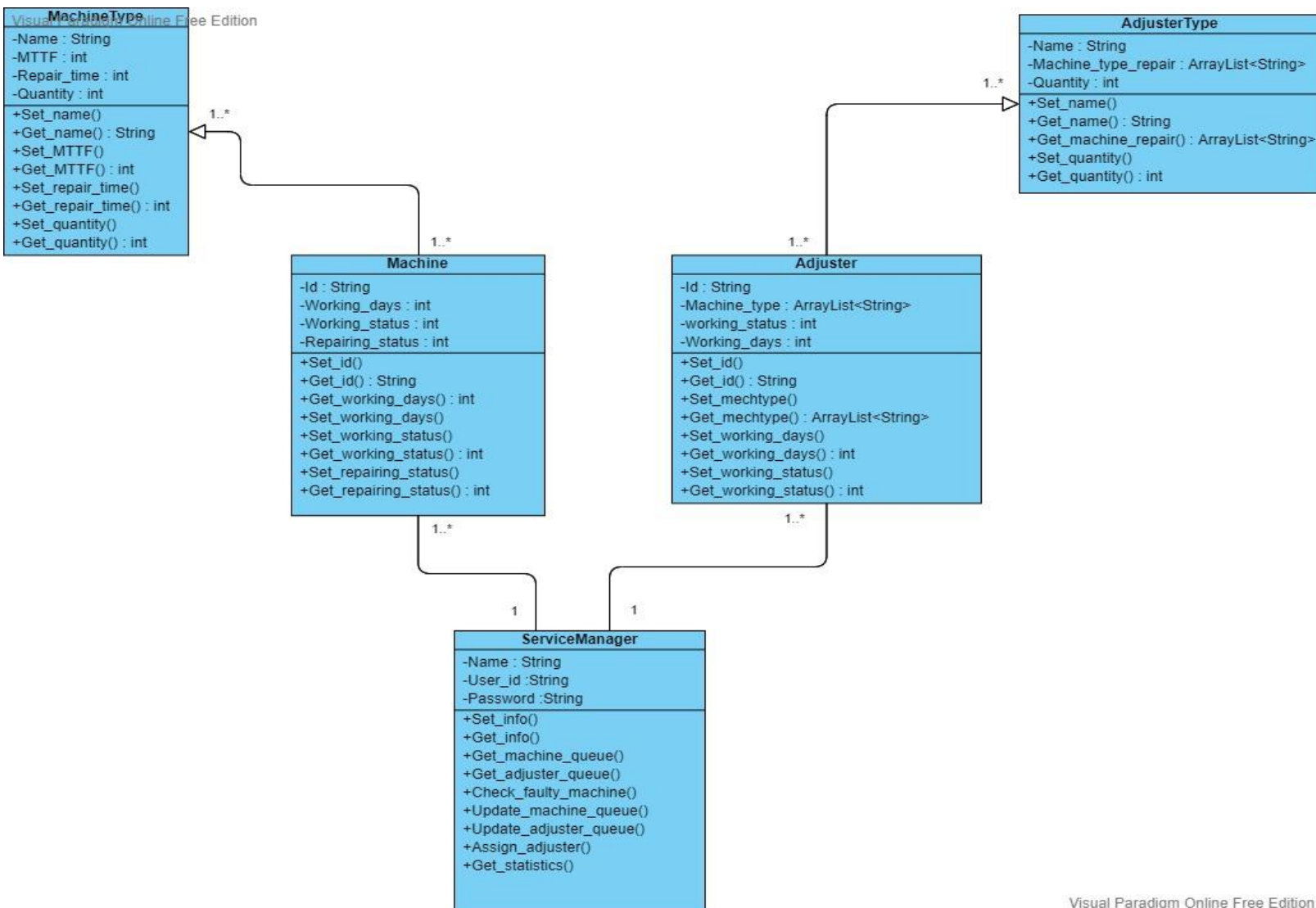
Machine Queue: One queue which contains all the inoperative machines in order.

Adjuster Queue: One queue which contains all the available free adjusters ready to repair faulty machines.

FSSS: Factory Service Simulation Software

## Appendix B: Analysis Models

### Class Diagram



## Appendix C: To Be Determined List

It is to be determined how much the service manager can edit and delete existing machines and adjusters and add some new adjuster or machine.