

Factory Simulation Software System

SOFTWARE ANALYSIS AND STRUCTURED
DESIGN(SA/SD)

GROUP-6

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

CONTENTS

1. Introduction

- 1.1. Scope of Project
- 1.2. Purpose
- 1.3. Glossary
- 1.4. References
- 1.5. Document Overview

2. Feasibility Study

- 2.1. Understanding the Problem
- 2.2. Scope of the Problem
- 2.3. Analyzing Stakeholders(DFD Analysis)
- 2.4 Report

3. Requirement Analysis

- 3.1. Functional Requirements
- 3.2. Non Functional Requirements
- 3.3. Report

4. Structural Chart

1. INTRODUCTION

1.1 Scope of Project:

This(FSSS) 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.

1.2 Purpose:

This SA/SD Document provides a complete description of all the functions and specifications of the Factory Service Simulation Software (FSSS) developed for determining the machine utilization and adjuster utilization of a Factory. And, also an optimum number of adjusters they should employ. The expected users of this software are the Service Manager of a Factory.

1.3 Glossary:

FSSS Factory Service Simulation Software

DFD Data Flow Diagram

1.4 References:

IEEE_Std_830_1998___Recommended_Practice_for_SA/SD

1.5 Overview of the Document:

The below chapters and their contents are:

Section 2 is the Feasibility study which helps us understand the problem by analyzing the stakeholders and their functions and also their alternatives.

Section 3 is Requirements analysis where the various functional and nonfunctional requirements are elucidated.

Section 4 is the last section where the Global System Architecture is defined and the platform including hardware, software, and networking.

2. FEASIBILITY STUDY

2.1 Understanding the Problem

FSSS is intended to determine the machine utilization and adjuster utilization of a Factory. It is used to help various stakeholders in the process realize their responsibility properly and promptly. Since everything is automated, there is very little scope for inconsistency and ambiguities.

2.2 Scope of the Problem

- Assignment of the inoperative machine to the available adjuster by the Service Manager.
- Assignment of the adjuster to the inoperative machine by the Service Manager.
- Repairing the assigned machine by the adjuster.
- Getting the machine utilization and adjuster utilization by Head.

2.3 Analyzing the Stakeholders

There are a set of registered people in the system.

- Adjusters
- Service Manager
- Head of the Factory

Adjuster: First, the Adjuster logs in the system using his username and password. Adjusters repair the Inoperative machine assigned by the Service Manager.

Service Manager: The System responds when the Service Manager updates the Queues. First, the Service Manager logs in to the system using his username and password. The dialog box which opens has three options: Get Queue, Update Queue and Assign Adjuster to repair a machine. Each option does the thing as its name suggests.

Head: First, the Head logs in the system using his username and password. Then, the Head analyzes statistics which includes Machine Utilization and Adjuster Utilization.

Level 1 DFD: Level 1 DFD is the Data Flow diagram which is a context diagram which is made to explain the relations between many entities in the software. Below is the DFD for the FSS Software.

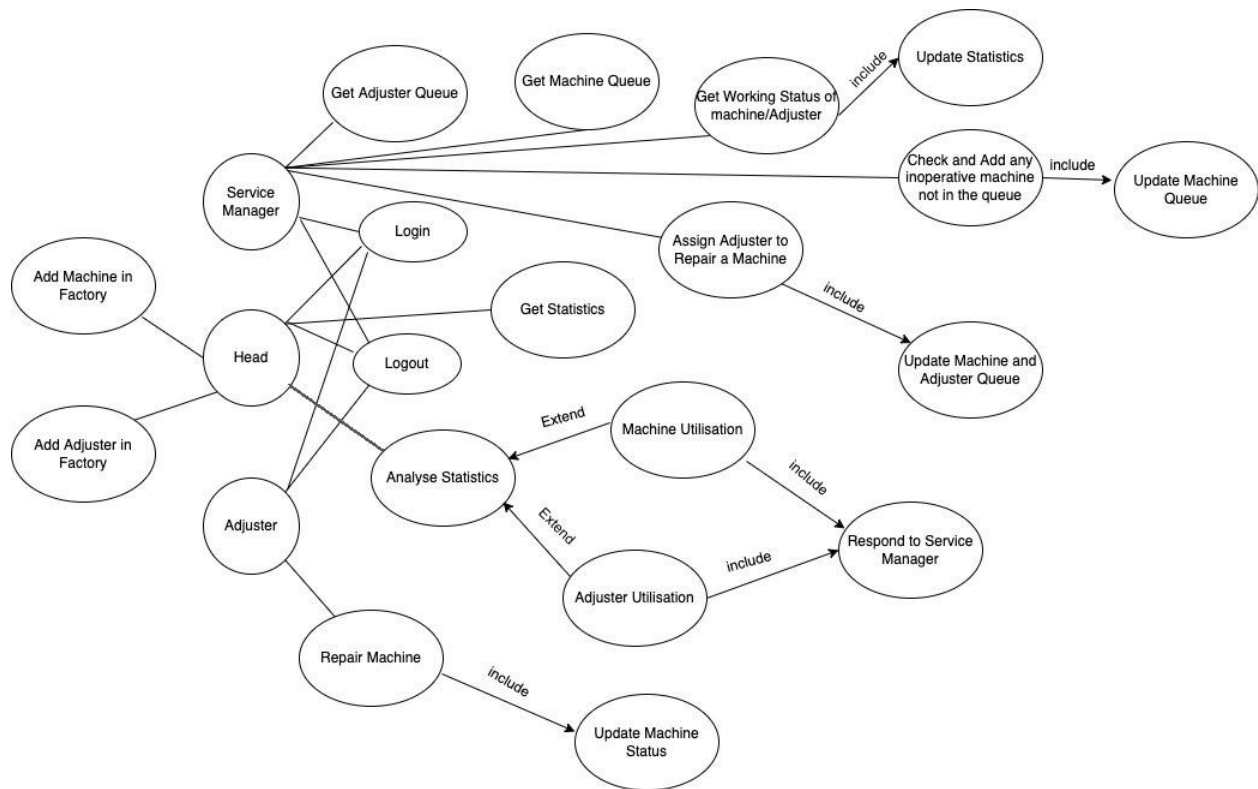


Fig: LEVEL 1 DFD (Use Case Diagram of Adjuster, Service Manager, and Head of the Factory)

2.4 Report

In the present section, we went through the various details of the problem statement. The objectives and scope have been presented in detail. Then, the roles of different stakeholders were analyzed. The alternatives were developed while noting the cost and lifetime of the parts the alternative brings. The alternatives include software, hardware, and security. The primary criteria of evaluation of the alternatives were discussed.

3. REQUIREMENT ANALYSIS

3.1 Functional Requirements

Data Flow

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

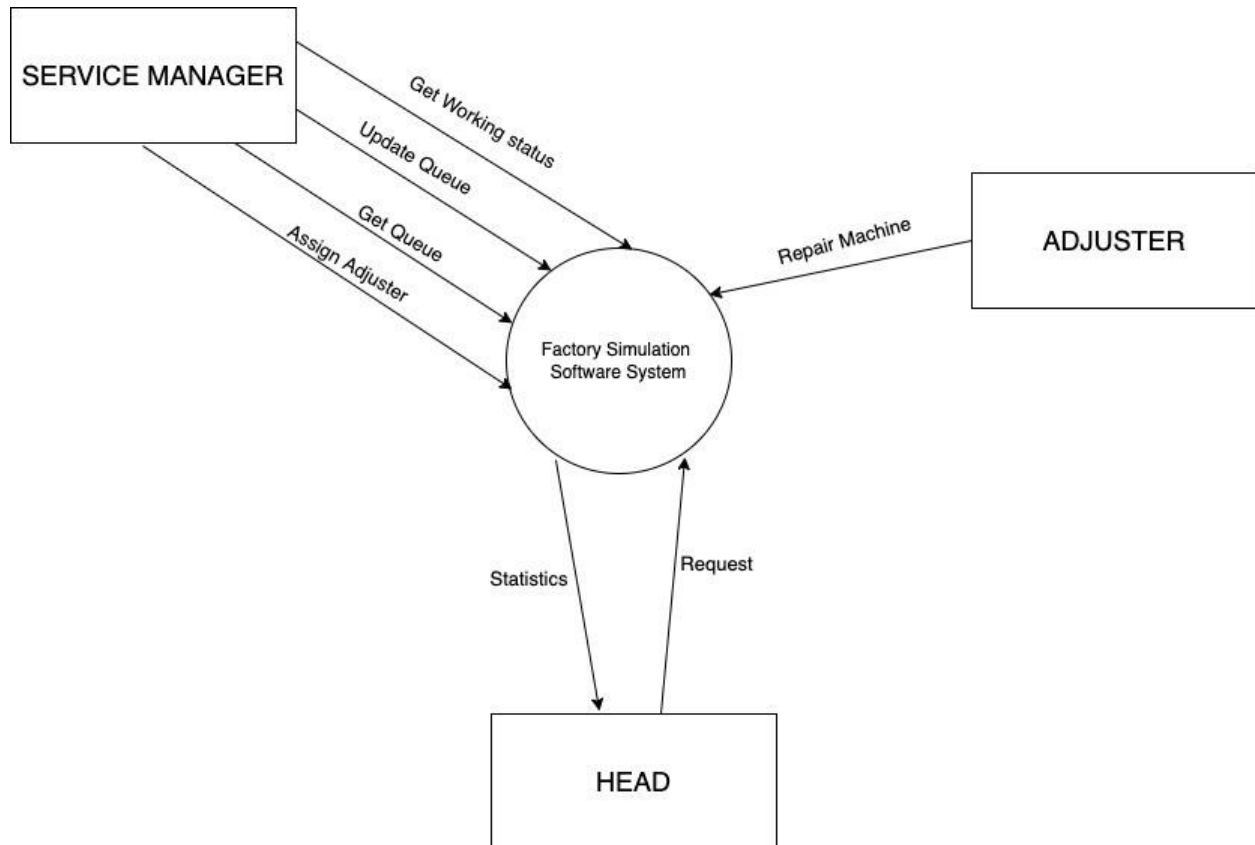


Fig: Level-0 DFD

3.2 NON-FUNCTIONAL REQUIREMENTS:

Database Requirements: If the no. of machines or adjusters are more, then the size of the database will increase proportionally. So, there must be space on the hard disk to store the total amount of data which might be more than what was specified in the hardware specifications section.

Legal Requirements: FSSS cannot be distributed freely by anyone as it has a software license agreement.

Availability of FSSS: FSSS will be available as long as the office of the corporation is open and any of the stakeholders is present.

Data Dictionary:

- Machine-name: string
- MTTF: int
- Quantity: int
- Repair-time: int
- Machine_id: String
- Adjuster_id: String
- Adjuster_Name: String
- Adjuster_Email: String
- Machine_type: String
- Service_Managaer_Name: String
- Manager_id: String
- Manager_email: String
- Head_name: String
- Head_id: String
- Head_email: String

3.3 Report:

In this section, the functional requirements of FSSS are explained in detail using data flow diagrams and a Structure chart. The DFD graphically represents the “flow” of data through the system. The DFD along with the Structure Chart help in the visualization of data processing of FSSS.

The Nonfunctional requirements are also discussed which ensure an operable and manageable system that functions uninterrupted and in a reliable fashion.

4. STRUCTURAL CHART

The structure chart shows the breakdown of our simulation software to its lowest manageable levels in the form of a tree. The different states of the software appear as labeled rectangles, and the arrows denote relationships and data flow between the various states.'

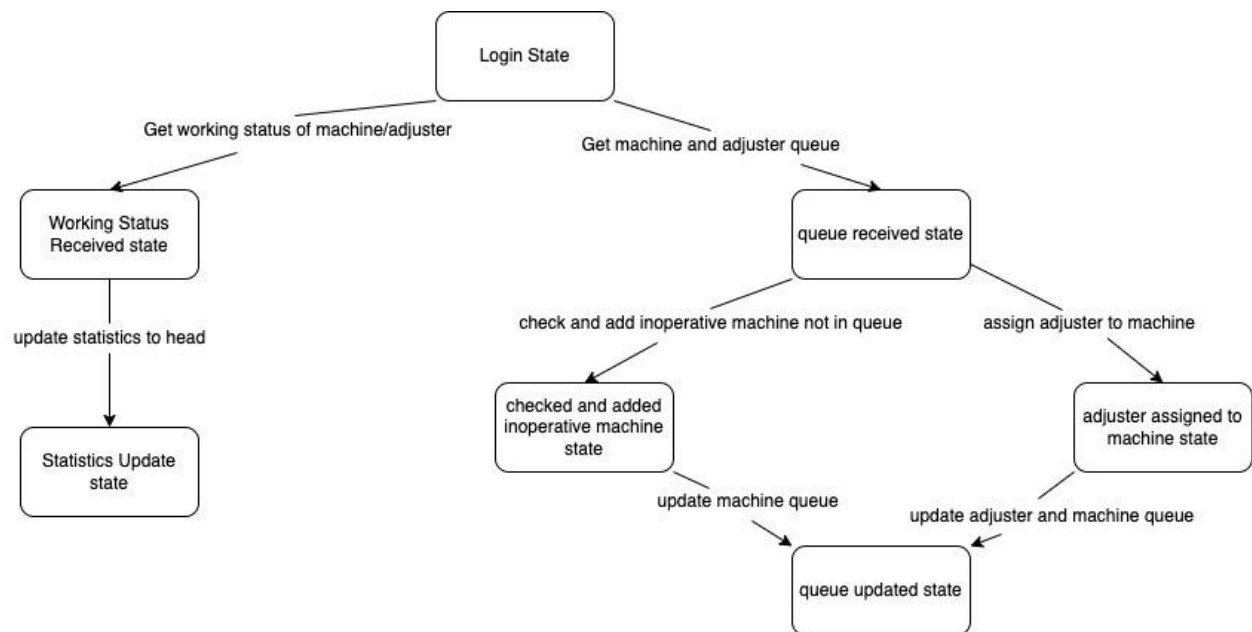


Fig: Structural Chart

Points:

- Module at the top is called Root. Here, Login is the root module.
- Control relationships refer to the functions that connect different modules and the flow of control occurs according to these arrows only.