



MACHINE JOB SCHEDULING

Software Requirement Specification (SRS) Document

Sprint 1 Implementation

Project Timeline: 07.09.2022 to 14.09.2022

INDEX

1. Introduction		
1.1 Purpose	-----	3
1.2 Intended audience	-----	3
1.3 Intended use	-----	3
1.4 Scope	-----	3
2. Overall description	-----	4
2.1 Assumptions and dependency	-----	4
3. System feature and requirements	-----	4
3.1 Functionality	-----	4
3.1.1 MJ_01 createList	-----	4
3.1.2 MJ_02 addToList	-----	4
3.1.3 MJ_03 stringValid	-----	4
3.1.4 MJ_04 invalidFile	-----	4
3.1.5 MJ_05 displayList	-----	4
3.1.6 MJ_06 validAdd	-----	4
3.1.7 MJ_07 assignJob	-----	4
3.2 System requirement	-----	5
3.2.1 Tools to be used	-----	5
3.3 System Features	-----	5
4. DataFlow Diagram	-----	6
4.1 DFD level 0	-----	6
4.2 DFD level 1	-----	6

1. Introduction: -

The introduction of the software requirement specification provides an overview of the entire software. The entire SRS with overview description purpose, scope, tools used and basic description. The aim of this document is to gather, analyze and give an in-depth insight into the complete Machine Job Scheduling application by defining the problem statement in detail. The detailed requirements of the Machine Job Scheduling application is provided in this document.

1.1 Purpose: - The purpose of this document is to show the requirements for the “Machine Job Scheduling Application”, in which we will process jobs data and assign requested machines to the jobs then schedule for each machine is generated.

1.2 Intended Audience: - This document is intended to be read by, Client.

1.3 Intended Use: -

- Development Team
- Maintenance Team
- Clients

Since this a general-Purpose Software any one can access it.

1.4 Scope: -

In a manufacturing company, there are three machines to do manufacturing jobs in which the jobs are received from clients in separate text files. The machine job scheduling application is to be developed, which will be processing the jobs data and then assigning requested machines to the jobs. A schedule for each machine is generated in text files.

2. Overall Description: -

This project aims to create the development of a Machine Job Scheduling Application. Which takes the Job description files as input which contains information like: Job No, Description, Machine No, Duration(in minutes), Client name (optional). Three machines are resources to be shared among jobs. All machines start at time T1. Whenever a machine is allocated to a job its time will advanced by “Duration” and an entry will be written in “Schedule”. Schedule entries contain - Job No, Start time, End time.

2.1 Assumptions and Dependency: -

- System should have Ubuntu Linux installed.
- System should have either 4GB or more RAM.
- The service is used preferably on a desktop or laptop.

3. System Features and Requirements: -

3.1 Functionality: -

3.1.1 MJ_01 : createList : This function is mainly used to create the linkedlist.

3.1.2 MJ_02 : addToList : This function is used to separate each word of the valid line and then it stores that words into structured format.

3.1.3 MJ_03 : stringValid : This function is used whether the string present in the file given by the client is valid or not. Valid in the sense the string present in the file is in the format needed by the machine.

3.1.4 MJ_04 : invalidFile : This function is used to store the strings which are not valid into the invalidRecord.txt.

3.1.5 MJ_05 : displayList : This function is used to display the contents of the valid strings present in the linkedlist.

3.1.6 MJ_06 : validAdd : This function stores the valid jobs that are needed to be performed by the machine into three text files for three different machines. It stores the information like job no.start time, and end time.

3.1.7 MJ_07 : assignJob : This function is used to fetch the file line by line and it checks if the line is valid or not, suppose if the line is valid then it appends

the entry into the respective schedule files and if the line is invalid then it calls the invalid file function.

3.2 System Requirements: -

3.2.1. Tools to be used:

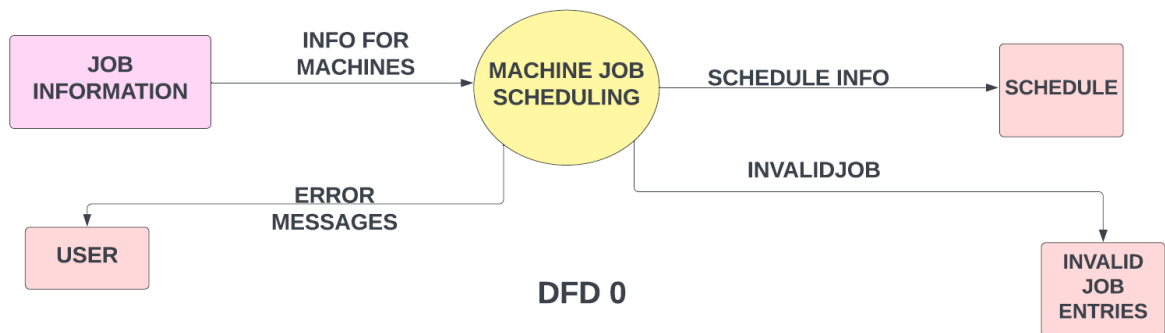
- Pthread Library
- C File Handling
- C Language
- System Programming
- Gprof
- Gcov
- Cunit
- Valgrind
- Splint
- GDB
- Makefile

3.3 System Features: -

- Supportability: The system is easy to use.
- Design Constraints: The system is built using only C language.
- Usability: The automatic bowling scorer application can be used to replace the old means. Recording score by displaying various information to the player such as number of pins knocked down by each ball, the frame number which is currently going on, the cumulative score gained throughout the the end of each frame and the total number of points scored by the player after the completion of a total of ten frames.
- Reliability & Availability: The system is available 24/7 that is whenever the user would like to use the system, they can use it up to its functionalities.
- Performance: The system will work on the user's terminal.

4. DataFlow Diagram:

4.1 DFD Level 0 –



4.1 DFD Level 1 –

