

MACHINE JOB SCHEDULING

**Software Requirement Specification (SRS) Document Sprint 1 Implementation**

**Project Timeline: 06.12.2022 to 13.12.2022**

# INDEX

1. Introduction
   1. Purpose
   2. Intended audience
   3. Intended use
   4. Scope
   5. Key project objectives
2. Overall description
   1. Assumptions and dependency
3. System feature and requirements
   1. Functionality

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3.1.1 | MJ\_01 | fopen() |  |  |  |  |  |  |
| 3.1.2 | MJ\_02 | fclose() |  |  |  |  |  |  |
| 3.1.3 | MJ\_03 | fgets() |  |  |  |  |  |  |
| 3.1.4 | MJ\_04 | finddisplay&removeinvalidjobs() |  |  |  |  |  |  |
| 3.1.5 | MJ\_05 | schedulejob() |  |  |  |  |  |  |
| 3.1.6 | MJ\_06 | groupschedulelist() |  |  |  |  |  |  |
| 3.1.7  3.1.8 | MJ\_07  MJ\_08 | writeschedule()  readfile() |  |  |  |  |  |  |

* 1. System requirement
     1. Tools to be used
  2. System Features

1. Dataflow Diagram
   1. HLD
   2. LLD

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

### Intended Use: -

* + - Development Team
    - Maintenance Team
    - Clients

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

### Scope: -

In a manufacturing company, three machines are available to do manufacturing tasks, and the work of each machine is received from clients in different text files. It is necessary to create the machine job scheduling application, which will process the job data and assign the required machines to the jobs. Text files are created which include a schedule for each machine.

**1.5 Key Project Objectives:**

* Get a sorted way to assign jobs to machines.
* Full potential use of machines.
* Increases on time completion of jobs.
* Save time while assigning jobs to machines.

# Overall Description: -

The goal of this project is to construct a machine job scheduling application. It accepts as input Job Description files containing the following data: Job No, Description, Machine No, Duration(in minutes), Client name (optional). Three machines are resources that can be used by several jobs. At time T1, all machines operate. Every time a machine is assigned to a task, "Duration" is increased and an entry is made in "Schedule". Job No., Start time, and End time

are all listed in schedule entries.

### 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.

# System Features and Requirements: -

## Functionality: -

* + 1. **MJ\_01 : fopen :** it opens the file with the permissions of (w,r,wr)
    2. **MJ\_02 : fclose:** it closes the file
    3. **MJ\_03 : fgets():** it reads each char or string

**3.1.4 MJ\_04 : finddisplay&removevalidjobs() :** its displays invalid job entries

**3.1.5 MJ 05 : schedulejob() :** copies fields in job list into schedule list and also

includes start and end time

**3.1.6. MJ 06 : groupschedulelist() :** opensschedule files for each machine in

read and write mode to make entry of incoming.

**3.1.7 MJ 07 : writeschedule()** : writejob id and schedule of each machine into

respective schedule text file

**3.1.8 MJ 08 : readfile() :** reads input file and add to job list

## System Requirements: -

### to be used:

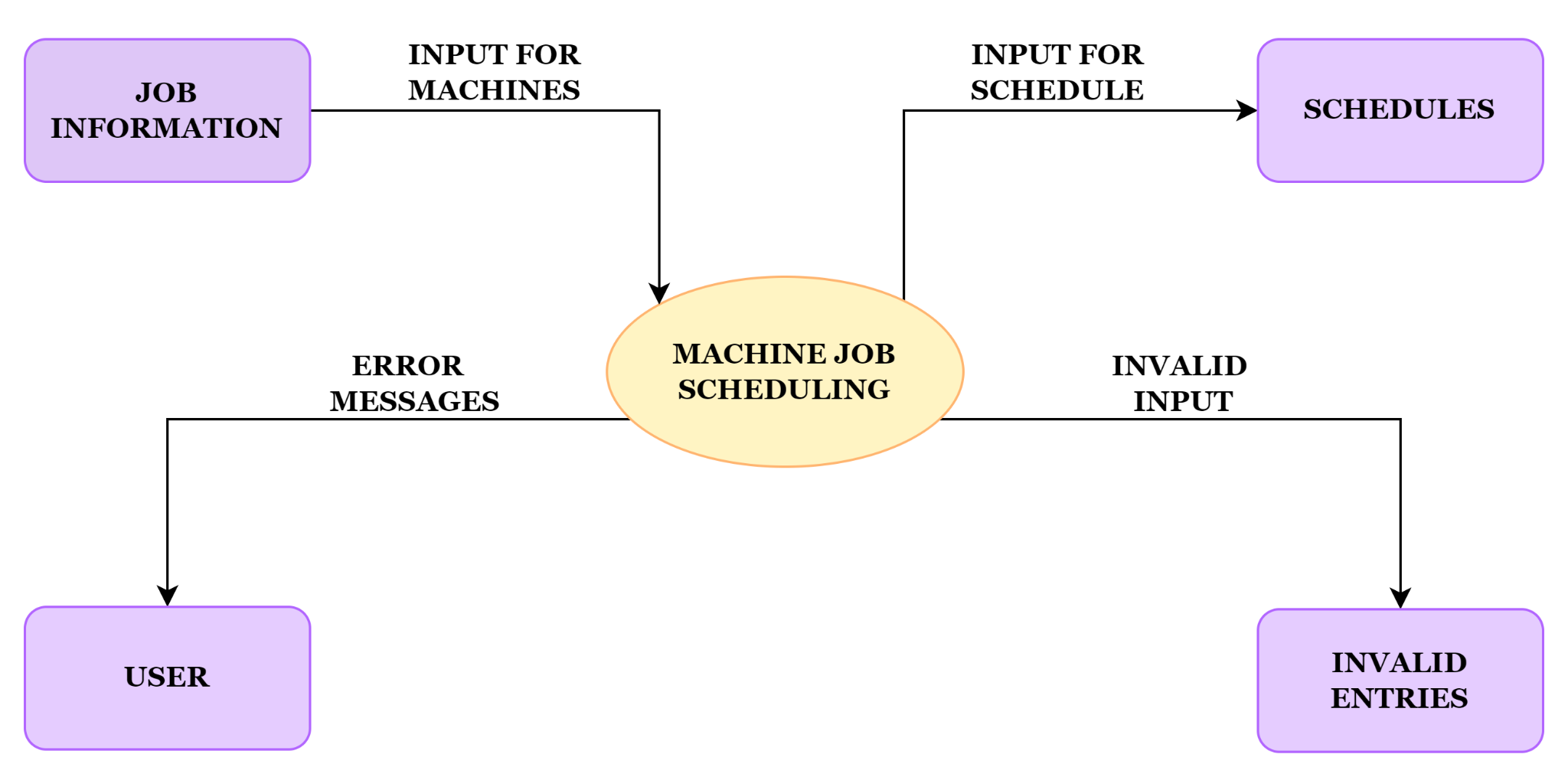
* + - * Valgrind
      * Make file
      * Splint

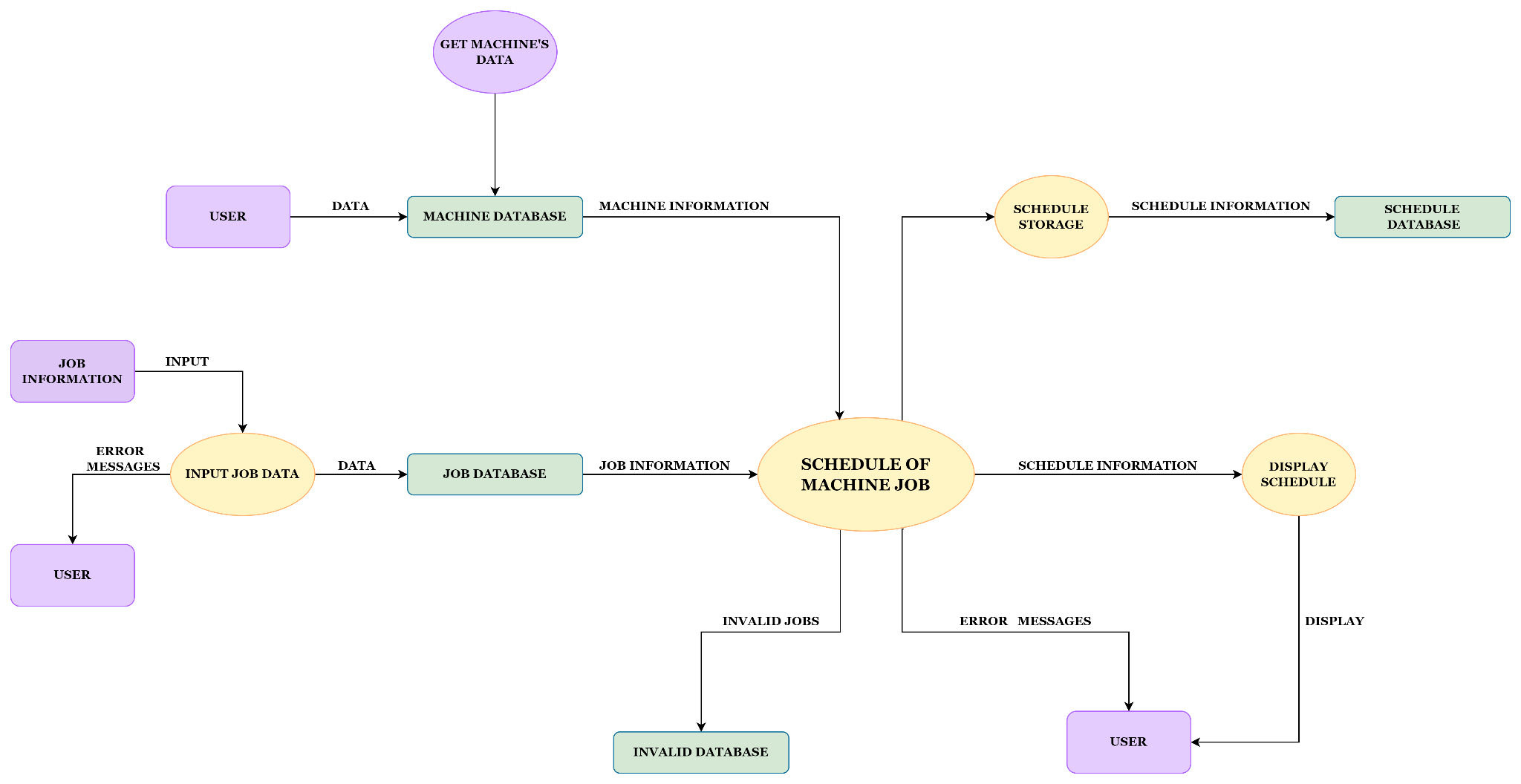
## System Features: -

* **Supportability:** The system is easy to use.
* **Design Constraints:** The system is built using only C language.
* **Usability:** The machines software can be used to process each and every file ,and allocate the particular machines for the updated files. The machine software can be used to process each file and assign specific machines to the updated files.
* **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.

# DataFlow Diagram:

### HLD –



4.2 LLD