

**SIMULATION OF SCHEDULER**

**Software Requirement Specification Document**

**Project Timeline: 28/11/2022 to 9/12/2022**

Team Members

M. Kavyasri

Mamatha

V.Varshitha

N.Ramya

Sk. Neha Sulthana

**Table of Contents**

**1. Introduction**

1.1 Purpose

1.2 Intended Audience and Reading Suggestions

1.3 Scope of Application

**2. Overall Description**

2.1 Application Perspective

2.2 Application Features

2.3 Design and Implementation Constraints

2.4 Assumptions and Dependencies

**3. System Features**

3.1 Functional Requirements

**4. Technical Requirements**

**5. Non Functional Requirements**

1. **Introduction**

**1.1 Purpose**

The purpose of this Requirements Elicitation document is to provide a clear understanding as to what actually is Simulation of a Scheduler application and to identify the critical requirements essential for the application successful completion. This document provides an abstract overview of the Simulation of Scheduler and provides a general overview of the entire application.

**1.2 Intended Audience**

This document explains our team architecture, our team’s initial understanding of the user needs. It will assist our team in understanding the system specifications and analyze the critical aspects of our project. This document will briefly discuss the stakeholders involved in the development, documents will show how our team was divided to handle the multiple stakeholders, the sources of the requirements, provide an informal preliminary requirements description, and address any issues encountered while transforming the requirements.

**1.3 Scope of Application**

The scope of the Simulation of Scheduler is to show the process scheduling for different types of algorithms and displaying the running process and processes in queue to user. We are having user interface to interface the user data to server and from server to scheduler.

**2. Overall Description**

**2.1 Application Perspective**

Simulation of Scheduler gives the following information.

* Algorithm:

It means it runs based on the algorithm selected and process and attributes for each algorithm varies and type of algorithm depends on the user.

* Running and waiting process:

It includes the current running process queue and wait queues to store process which re in wait state.

* Display:

Displays the current running process of algorithm and the processes that are in queue.

**2.2 Application Features**

The major application of this feature is its client-server relationship and internal process selection and running processes based on the type of algorithm. The process of compilation is different for different types of algorithms chosen and it is dependent on one other.

**2.3 Design and Implementation Constraints**

1. The fragmentation schema and maintaining server-client relation.

2. Using Data structures and System calls for above application.

3. Generating response at user Interface for every phase in server.

4. Implementing process separately for each type of algorithm.

**2.4 Assumption and Dependencies**

Assuming if two types of algorithms to run simultaneously,

It needs server to decide which algorithm to run first and also efficiently for two processes. Implementation of this assumption will be depending on the accuracy to switch between algorithms.

**3. System Features**

**3.1 Functional Requirements**

**3.1.1** The application should display a main menu to select options such as algorithm choice menu, help etc.

**3.1.2** The application should display further sub menu options based on selected menu options.

**3.1.3** The application should validate the menu options at each level. Ifany incorrect option or entry by the user should display an error.

**3.1.4** For FCFS algorithm selection, the given process order is treated as the order of execution.

**3.1.5** For SJF/LJF algorithm selection, the user should provide total execution time in seconds for each process along with process path in format given below.

**3.1.6** For priority algorithm selection, the user should provide the priority for each process along with the process path in the format given below.

**3.1.7** For RR algorithm selection, the user should provide time quantum duration in seconds for each process along with process path.

**3.1.8** Start the scheduler if the option is chosen.

**3.1.9** The application should maintain a data structure to store each algorithm related data which will be taken from the user.

**4. Technical Requirements**

* Process synchronization
* Shared Memory in Linux
* Socket Programming in TCP
* Support for statistics
* I/O Multiplexing (in Linux using CPP)
* Logging and debugging framework

**5. Non Functional Requirements**

* Multi-file multi-directory solution is expected.
* Makefile to build application.
* Use valgrind tool on application executable to detect memory leak. Final valgrind report to be submitted in “reports” directory.
* Level 0 DFD (context diagram), Level 1 DFD, Flow diagram and 2 flowcharts showing core functions logic.
* Use CPP Unit to automate unit testing.
* Any other UML diagrams developed while designing the system.
* HLD, LLD of the system.
* RTM, Plan, Presentation
* Unit test cases and Integration test cases in UT\_IT document. Both types of test cases i.e. sunny and rainy should be present in this document